CTWatch QUARTERLY

ISSN 1555-9874 AVAILABLE ON-LINE AT http://www.ctwatch.org/quarterly/ VOLUME 4 NUMBER 1 **MARCH 2008**

Urgent Computing: Exploring Supercomputing's New Role

GUEST EDITOR **PETE BECKMAN**

CyberInfrastructure Partnership

Cyberinfrastructure Technology Watch
http://www.ctwatch.org/

CTWatch QUARTERLY

ISSN 1555-9874 VOLUME 4 NUMBER 1 MARCH 2008

Urgent Computing: Exploring Supercomputing's New Role

GUEST EDITOR **PETE BECKMAN**

From the Editor-in-Chief

Jack Dongarra
University of Tennessee

Our all-too-frequent sense of surprise at how fast time has passed is usually at its most acute when we come to the end of a long, collective effort, especially one which has proved personally meaningful to those who have participated. Now that we have arrived at the last issue of the *Cyberinfrastructure Technology Watch Quarterly (CTWatch Quarterly)*, at least in its current incarnation, all of us on the CTWatch team are having this familiar experience, and in just that context. Typically this is also a good time to reflect on what has been accomplished and to think a little about where things should go next.

Back in 2004, when, in a moment of inspiration, Fran Berman saw that something like *CTWatch Quarterly* was needed as a complement to the Cyberinfrastructure Partnership (CIP) that the San Diego Supercomputer Center and the National Center for Supercomputing Applications were planning to form, the term "cyberinfrastructure" was still something of a buzzy neologism. The National Science Foundation, where the word was coined, had yet to fund the CIP, or to create the separate office that now bears this name. But some in the community recognized that "cyberinfrastructure" was not just marketing gloss, but that it represented a complex new reality which, in the digital age, was becoming more and more fundamental to scientific inquiry on every front. Thoroughly crosscutting, broadly interdisciplinary, and global in its reach and impact, the world of cyberinfrastructure already formed the nexus of a myriad of interconnected activities where the interests of government, private industry, and the academy all converged. Thus the spring of 2005 was a propitious time to launch a publication intended to provide a forum where the cyberinfrastructure community could discuss the opportunities, achievements, and challenges confronting it.

Looking back to review the three years and thirteen issues of *CTWatch Quarterly*, it is gratifying to see how many emerging developments and how much breadth of impact its pages have been able to reflect. Each installment contains illustrations of this point, but a few examples provide a happy reminder. Our issues on low power, high-performance computing (Fall 2005) and on the ramifications of the on-going revolution in multicore and heterogeneous processor architectures (Spring 2007) helped lead the way in raising awareness and educating the community on these watershed developments in the evolution of computing infrastructure. The two issues that focused on the global explosion of national and transnational cyberinfrastructure (Winter 2005, focusing on Europe, and Spring 2006, highlighting eight national projects spread across Asia, Africa, and South America) showed clearly that the era of e-Science has become ubiquitous, and that the drive to remain competitive is being reflected in significant infrastructure investments around the world. The two recent issues on cyberinfrastructure for the humanities, arts, and social sciences (Summer 2007) and on the digitally-driven transformation of scholarly communications (Fall 2007) show how the impact of cyberinfrastructure is already reaching every field and discipline across the entire curricula. Finally, in the current issue on urgent computing, guest editor Pete Beckman and his excellent group of author's show us that the practical and societal benefits of advanced cyberinfrastructure are on the verge of becoming more immediate, more universal, and more vitally important than ever before.

Given the current state of the field and the community, it is clear to me that the relevance of *CTWatch Quarterly*'s mission is far from exhausted. But the question of how to carry it forward into the post-CIP future remains open. Opportunities to continue with the quarterly in its current form are being explored, and if adequate funding can somehow be secured, publication may restart in the near future. Yet although that may be the easiest path to take, it may not be the most satisfying one. Perhaps the most remarkable feature of the cyberinfrastructure community is its propensity to innovate: not doing what you've done before with different tools, but reflecting on the untried opportunity space that emerging technology opens up for you and trying to envision your mission and your strategy for achieving it in a more original way. We are also searching for, and may yet find, a future for CTWatch more consonant with that spirit of innovation.

As we close up shop at the current stand, it is important to recognize the people who have been instrumental in helping us to achieve such success as we have had. First and foremost, we are profoundly grateful to the remarkable collection of guest editors and outstanding authors who worked with us over the last three years; they literally gave substance to our vision of a publication that could help keep the attention of the community focused on the leading edge of the cyberinfrastructure movement. Special thanks go to Fran Berman and Thom Dunning, and the CIP organizations they lead (SDSC and NCSA, respectively), for the constant and enthusiastic support they provided throughout. We also very much appreciate the contributions, feedback and suggestions of the members of our editorial board, who took time out of their busy schedules to help keep us on course and make us better.

Finally, I want to personally thank the members of the CTWatch team: Terry Moore (Managing Editor), Scott Wells (Production Editor), David Rogers (Graphic Designer), and Don Fike (Developer). Their expertise, diligence, and collective efforts, working through every phase of the process, made a high quality production like *CTWatch Quarterly* seem deceptively easy. Their presence makes the possibility of future community endeavors along the line of CTWatch a very happy prospect indeed.

Jack Dongarra,
Editor-in-Chief

INTRODUCTION

Urgent Computing: Exploring Supercomputing's New Role

Pete Beckman
Argonne National Laboratory

Large-scale parallel simulation and modeling have changed our world. Today, supercomputers are not just for research and development or scientific exploration; they have become an integral part of many industries. A brief look at the Top 500 list of the world's largest supercomputers shows some of the business sectors that now rely on supercomputers: finance, entertainment and digital media, transportation, pharmaceuticals, aerospace, petroleum, and biotechnology. While supercomputing may not yet be considered commonplace, the world has embraced high-performance computation (HPC). Demand for skilled computational scientists is high, and colleges and universities are struggling to meet the need for cross-disciplinary engineers who are skilled in both computation and an applied scientific domain. It is on this stage that a new breed of high-fidelity simulations is emerging – applications that need urgent access to supercomputing resources.

For some simulations, insights gained through supercomputer computation have immediate application. Consider, for example, an HPC application that could quickly calculate the exact location and magnitude of tsunamis immediately after an undersea earthquake. Since the evacuation of local residents is both costly and potentially dangerous, promptly beginning an orderly evacuation in only those areas directly threatened could save lives. Similarly, imagine a parallel wildfire simulation that coupled weather, terrain, and fuel models and could accurately predict the path of a wildfire days in advance. Firefighters could cut firebreaks exactly where they would be most effective. For these *urgent computations*, late results are useless results. As the HPC community builds increasingly realistic models, applications are emerging that need on-demand computation. Looking into the future, we might imagine event-driven and data-driven HPC applications running on-demand to predict everything from where to look for a lost boater after a storm to tracking a toxic plume after an industrial or transportation accident.

Of course, as we build confidence in these emerging computations, they will move from the scientist's workbench and into critical decision-making paths. Where will the supercomputer cycles come from? It is straightforward to imagine building a supercomputer specifically for these emerging urgent computations. Even if such a system led the Top 500 list, however, it would not be as powerful as the combined computational might of the world's five largest computers. Aggregating the country's largest resources to solve a critical, national-scale computational challenge could provide an order of magnitude more power than attempting to rely on a prebuilt system for on-demand computation.

Furthermore, costly public infrastructure, idle except during an emergency, is inefficient. A better approach, when practical, is to temporarily use public resources during times of crisis. For example, rather than build a nationwide set of radio towers and transmitters to disseminate emergency information, the government *requires* that large TV and radio stations participate in the Emergency Alert System. When public broadcasts are needed, most often in the form of localized severe weather, broadcasters are automatically interrupted, and critical information is shared with the public.

As high-fidelity computation becomes more capable in predicting the future and being used for immediate decision support, governments and local municipalities must build infrastructures that can link together the largest resources from the NSF, DOE, NASA, and the NIH and use them to run time-critical urgent computations. For embarrassingly parallel applications, we might look to the emerging market for "cloud computing." Many of the world's largest Internet companies have embraced a model for providing software as a service. Amazon's elastic computing cloud (EC2), for example, can provide thousands of virtual machine images rapidly and cost effectively. For applications with relatively small network communication needs, it might be most effective for urgent, on-demand computations simply to be injected into the nation's existing Internet infrastructure supported by Amazon, Yahoo, Google, and Microsoft.

In April 2007, an urgent computing conference at Argonne National Laboratory brought together an international group of scientists to discuss how on-demand computations for HPC might be supported and change the landscape of predictive modeling. The organizers of that workshop realized that *CTWatch Quarterly* would be the ideal venue for exploring this new field. This issue describes how applications, urgent-computing infrastructures, and computational resources can support this new role for computing.

LEAD Cyberinfrastructure to Track Real-Time Storms Using SPRUCE Urgent Computing

Suresh Marru
Dennis Gannon
Indiana University

Suman Nadella
The University of Chicago

Pete Beckman
Argonne National Laboratory

Daniel B. Weber
Tinker Air Force Base

Keith A. Brewster
Kelvin K. Droegemeier
University of Oklahoma

A tornado strikes the Kansas Plains near Clearwater. Photo by Keith Brewster.

The Linked Environments for Atmospheric Discovery (LEAD)[1 2] project is pioneering new approaches for integrating, modeling, and mining complex weather data and cyberinfrastructure systems to enable faster-than-real-time forecasts of mesoscale weather systems, including those than can produce tornadoes and other severe weather. Funded by the National Science Foundation Large Information Technology Research program, LEAD is a multidisciplinary effort involving nine institutions and more than 100 scientists, students, and technical staff.

Foundational to LEAD is the idea that today's static environments for observing, predicting, and understanding mesoscale weather are fundamentally inconsistent with the manner in which such weather actually occurs – namely, with often unpredictable rapid onset and evolution, heterogeneity, and spatial and temporal intermittency. To address this inconsistency, LEAD is creating an integrated, scalable framework in which meteorological analysis tools, forecast models, and data repositories can operate as *dynamically adaptive, on-demand, Grid-enabled systems*. Unlike static environments, these dynamic systems can change configuration rapidly and automatically in response to weather, react to decision-driven inputs from users, initiate other processes automatically, and steer remote observing technologies to optimize data collection for the problem at hand. Although mesoscale meteorology is the particular domain to which these innovative concepts are being applied, the methodologies and infrastructures are extensible to other domains, including medicine, ecology, hydrology, geology, oceanography, and biology.

The LEAD cyberinfrastructure is based on a service-oriented architecture (SOA) in which service components can be dynamically connected and reconfigured. A Grid portal in the top tier of this SOA acts as a client to the services exposed in the LEAD system. A number of stable community applications, such as the Weather Research and Forecasting model (WRF),[3] are preinstalled on both the LEAD infrastructure and TeraGrid[4] computing resources. Shell executable applications are wrapped into Web services by using the Generic Service Toolkit (GFac).[5] When these wrapped application services are invoked with a set of input parameters, the computation is initiated on the TeraGrid computing resources; execution is monitored through Grid computing middleware provided by the Globus Toolkit.[6] As shown in Figure 1, scientists construct workflows using preregistered, GFac wrapped application services to depict dataflow graphs, where the nodes of the graph represent computations and the edges represent data dependencies. GPEL,[7] a workflow enactment engine based on industry standard Business Process Execution Language,[8] sequences the execution of each computational task based on control and data dependencies.

[1] Droegemeier, K. K. et al. (20 other authors), "Linked Environments for Atmospheric Discovery (LEAD): A Cyberinfrastructure for mesoscale meteorology research and education," in *Proc. 20th Conf. Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Am. Meteorological Soc., 2004.

[2] LEAD - http://leadproject.org/

[3] Michalakes, J., Chen, S., Dudhia, J., Hart, L., Klemp, J., Middleco, J., Skamarock, W. "Development of a next-generation regional weather research and forecast model," in *Ninth ECMWF Workshop on the use of Parallel Processors in Meteorology*, Reading, U.K., November 2000. Argonne National Laboratory preprint ANL/MCS-P868-0101, 2001.

[4] TeraGrid - http://www.teragrid.org/

[5] Kandaswamy, G., Fang, L., Huang, Y., Shirasuna, S., Marru, S., Gannon, D. "Building Web services for scientific Grid applications," *IBM Journal of Research and Development*, 50(2/3):249-260, 2006.

[6] Foster, I., Kesselman, C. "Globus: A metacomputing infrastructure toolkit," *IJSA*, 11(2):115-128, 1997.

[7] Slominski, A. "Adapting BPEL to scientific workflows," Chapter 14 in *Workflows for e-Science*, I. J. Tayler, E. Deelman, D. Gannon, and M. Shields, eds. Springer, 2007.

[8] Andrews, T. et al. "Business Process Execution Language for Web Services version 1.1," online, 5 May 2003. ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf.

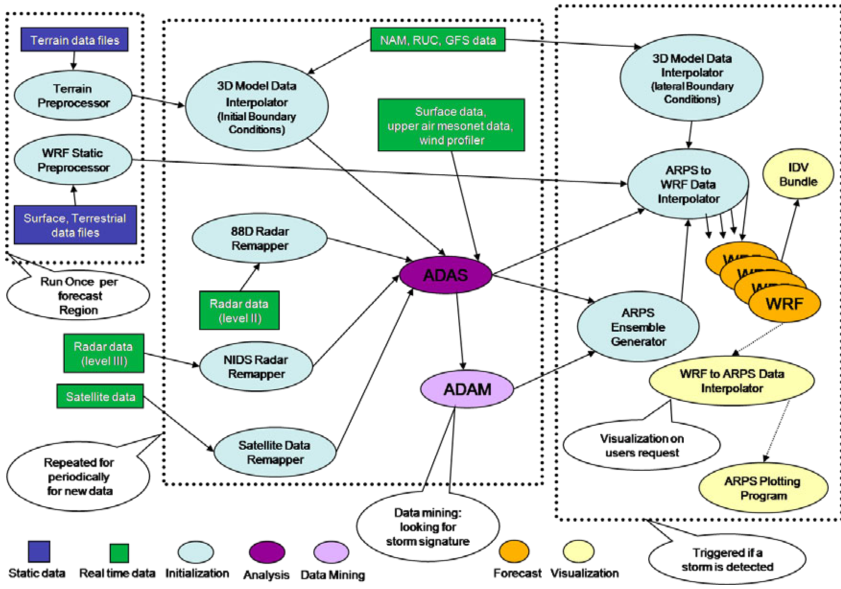


Figure 1. Dynamic workflow –WRF ensemble forecast initialized with assimilated data

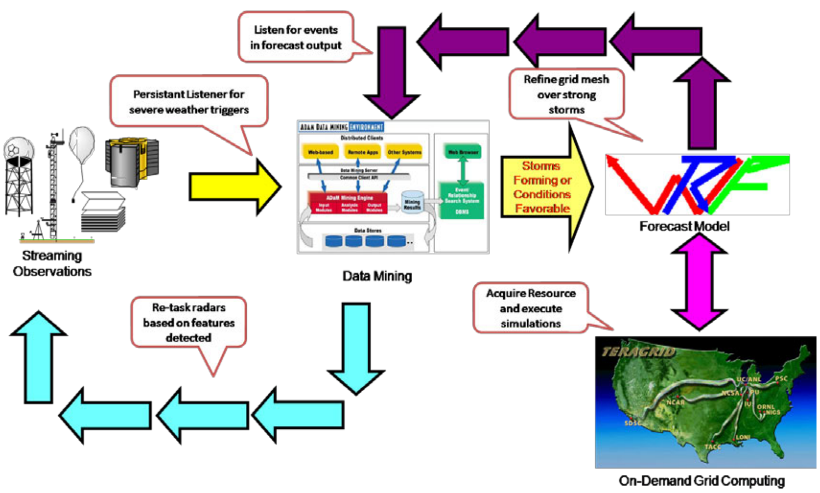


Figure 2. Dynamic adaptation in LEAD

Dynamic Adaptation to Weather

To dynamically interact and react to weather events (Figure 2), LEAD is working on adaptivity in four categories:

- Weather simulation and prediction
- Data collection
- Use of computational resources
- LEAD Cyberinfrastructure

In the following paragraphs, we briefly elaborate on these categories.

Adaptivity in Simulations: In the simulation phase of the prediction cycle, adaptivity in the spatial resolution is essential in order to improve the accuracy of the result. Specifically, finer computational meshes are introduced in areas where the weather looks more interesting. These may be run as secondary computations that are triggered by interesting activities detected in geographic subdomains of the original forecast simulation. Or they may be part of the same simulation process execution if it has been re-engineered to use automatic adaptive mesh refinement. In any case, the fine meshes must track the evolution of the predicted and actual weather in real time. The location and extent of a fine mesh should evolve and move across the simulated landscape in the same way the real weather is constantly moving.

Adaptivity in Data Collection: If we attempt to increase the resolution of a computational mesh in a local region, we will probably need more resolution in the data gathered in that region. Fortunately, the next generation of radars being developed by Center for Collaborative Adaptive Sensing of the Atmosphere (CASA)[9] [10] will be lightweight and remotely steerable. Hence, it will be possible to have a control service where a workflow can interact to retask the instruments to gain finer resolution in a specific area of interest. In other words, the simulation will have the ability to close the loop with the instruments that defined its driving data. If more resolution in an area of interest is needed, then more data can be automatically collected to make the fine mesh computationally meaningful. The relationship between LEAD and CASA is explained in detail in [11].

Adaptivity in Use of Computational Resources: Two features of storm prediction computations are critical. First, the prediction must occur before the storm happens. This faster-than-real-time constraint means that very large computational resources must be allocated as predicated by severe weather. If additional computation is needed to resolve potential areas of storm activity, then even more computational power must be allocated. Second, the predictions and assessment of uncertainty in the predictions can benefit from running ensembles of simulation runs that perform identical, or nearly identical, computations but start from slightly different initial conditions. As the simulations evolve, the computations that fail to track the evolving weather could be eliminated, freeing up computational resources. These resources in turn may be used by a simulation instance that needs more power. An evaluation thread must be examining the results from each computation and performing the ensemble analysis needed to gather a prediction. In all cases, the entire collection of available resources must be carefully brokered and adaptively managed to make the predictions work.

Adaptivity in LEAD Cyberinfrastructure: LEAD workflow infrastructure must respond to the dynamic behavior of the computational and grid resources in order to meet the requirement of "faster than real time" prediction. So a timely co-ordination of different components of the Cyberinfrastructure to meet soft, real-time guarantees is required. Co-ordination across the layers to allocate, monitor and adapt in real-time, while meeting strict performance and reliability guarantees and co-allocation of real-time data streams and computational resources, is required.

To summarize, LEAD has enormous demands: large data transfer, real-time data streams, and huge computational needs. But, arguably, most significant is the need to meet strict deadlines. On-demand computations cannot wait in a job queue for Grid resources to become available.

However, neither can the scientific community afford to keep multimillion dollar computational resources idle until required by an emergency. Instead, we must develop technologies that can support urgent computation. Scientists need mechanisms to find, evaluate, select, and launch elevated-priority applications on high-performance computing resources. Such applications might reorder, preempt, or terminate existing jobs in order to access the needed cycles in time.

[9] CASA - http://www.casa.umass.edu/

[10] Zink, M., Westbrook, D., Abdallah, S., Horling, B., Lakamraju, V., Lyons, E., Manfredi, V., Kurose, J., Hondl, K. "Meteorological command and control: An end-to-end architecture for a hazardous weather detection sensor network," pp. 37-42 in *Proc. 2005 Workshop on End-To-End, Sense-and-Respond Systems, Applications and Service., International Conference on Mobile Systems, Applications and Services*. USENIX Association, Berkeley, 2005.

[11] Plale, B., Gannon, D., Brotzge, J., Droegemeier, K., Kurose, J., McLaughlin, D., Wilhelmson, R., Graves, S., Ramamurthy, M., Clark, R. D., Yalda, S., Reed, D. A., Joseph, E., Chandrasekar, V. "CASA and LEAD: Adaptive cyberinfrastructure for real-time multiscale weather forecasting," special issue on system-level science, *Computer*, 39(11): 56-63, 2006.

To this end, LEAD is collaborating with SPRUCE, the *S*pecial *PR*iority and *U*rgent *C*omputing *E*nvironment TeraGrid Science Gateway.[12] SPRUCE provides resources quickly and efficiently to high-priority applications that must get computational power without delay.

[12] SPRUCE Science Gateway - http://spruce.teragrid.org/

SPRUCE

SPRUCE facilitates urgent computing by addressing five important concepts: session activation, priority policies, participation flexibility, allocation and usage policies, and verification drills.

SPRUCE uses a token-based authorization system for allocation and tracking of urgent sessions. As a raw technology, SPRUCE has no dictated priority policies; resource providers have full control and flexibility to choose possible urgency mechanisms they are comfortable with and to implement these mechanisms as the providers see fit. To build a complete solution for urgent computing, SPRUCE must be combined with allocation and activation policies, local participation policies for each resource, and procedures to support "warm-standby" drills. These application drills not only verify end-to-end correctness but also generate performance and reliability logs that can aid in resource selection.

Right-of-Way Tokens

Many possible authorization mechanisms could be used to let users initiate an urgent computing session, including digital certificates, signed files, proxy authentication, and shared-secret passwords. In time-critical situations, however, simpler is better. Complex digital authentication and authorization schemes could easily become a stumbling block to quick response. Hence, simple transferable tokens were chosen for SPRUCE. This design is based on existing emergency response systems proven in the field, such as the priority telephone access system supported by the U.S. Government Emergency Telecommunications Service in the Department of Homeland Security.[13] Users of the priority telephone access system, such as officials at hospitals, fire departments, and 911 centers, carry a wallet-sized card with an authorization number. This number can be used to place high-priority phone calls that jump to the top of the queue for both land- and cell-based traffic even if circuits are completely jammed because of a disaster.

[13] Telecommunications Service Priority (TSP) program - http://tsp.ncs.gov/

Figure 3. SPRUCE "right-of-way" token

The SPRUCE tokens (see Figure 3) are unique 16-character strings that are issued to scientists who have permission to initiate an urgent computing session. When a token is created, several important attributes are set, such as resource list, maximum urgency, sessions lifetime, expiration date, and project name. A token represents a unique "session" that can include multiple jobs and that lasts for a clearly defined period. It can also be associated with a group of users, who can be added or removed from the token at any time, providing flexible coordination.

SPRUCE User Eventflow

The SPRUCE eventflow is designed for application teams that provide computer-aided decision support or instrument control. A principal investigator (PI) organizes each application team and selects the computational "first responders," senior staff who may initiate an urgent computing session. First responders are responsible for evaluating the situation in light of the policies for using urgent computing.

As illustrated in Figure 4, the SPRUCE eventflow begins as the result of a trigger, which may be automatic (e.g., an automated warning from weather advisory RSS feed) or human-generated (e.g., a phone call to the PI). SPRUCE token holders are expected to use tokens with discretion and according to coordinated policies, similar to the way that citizens are expected to use good judgment before dialing 911. Token usage will be monitored and reviewed. Administrators can revoke tokens at any time. The first responder begins interaction with the SPRUCE system by initiating a session. Token activation can be done through a Web-based user portal or via a Web service interface. Systems built from the Web service interface can be automated and incorporated into domain-specific toolsets, avoiding human intervention. The initiator of the SPRUCE session can indicate which scientist or set of scientists will be able to request elevated priority while submitting urgent jobs. This set may later be augmented or edited.

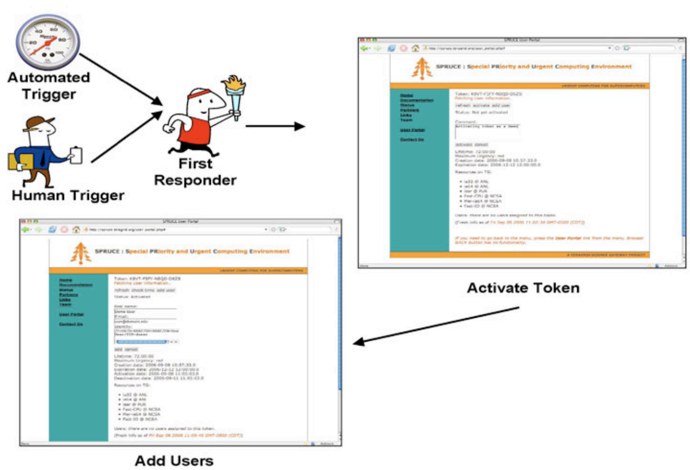


Figure 4. SPRUCE token activation

Resource Selection

Once a token is activated and the application team has been specified, scientists can organize their computation and submit jobs. Naturally, there is no time to port the application to new platforms or architectures or to try a new compiler. Applications must be prepared for immediate use—they must be in "warm standby." All of the application development, testing, and tuning must be complete prior to freezing the code and marking it ready for urgent computation. In the same way that emergency equipment, personnel, and procedures are periodically tested for preparedness and flawless operation, SPRUCE proposes to have applications and policies in warm-standby mode, being periodically tested and their date of last validation logged.

From this pool of warm-standby Grid resources, the team must identify where to submit their urgent jobs. One computing facility site may provide only a slightly increased priority to SPRUCE jobs, while another site may kill all the running jobs and allow an extremely urgent computation to use an entire supercomputer. Current job load and data movement requirements can also affect resource selection. Moreover, how a given application performs on each of the computational resources must also be considered. The SPRUCE advisor, currently under development, determines which resources offer the greatest probability to meet the given deadline. To accomplish this task, the advisor considers a wide variety of information, including the deadline, historical information (e.g., warm-standby logs, local site policies), live data (e.g., current network/ queue/resource status), and application-specific data (e.g., the set of warm-standby resources, performance model, input/output data repositories). To determine the likelihood of an urgent computation meeting a deadline on a given resource, the advisor calculates an upper bound on the total turnaround time for the job. More details on this implementation can be found in [14].

[14] Pete Beckman, Ivan Beschastnikh, Suman Nadella, and Nick Trebon, "Building an Ifrastructure for urgent computing, in *High Performance Computing and Grids in Action*. IOS Press, Amsterdam, 2007.

Prioritized Job Submission

Once the resource is chosen based on the advisor, the job is submitted. SPRUCE provides support for both Globus-based urgent submissions and direct submission to local job-queuing systems. Currently SPRUCE supports all the major resource managers such as Torque, LoadLeveler, and LSF and schedulers such as Moab, Maui, PBS Pro, SGE, and Catalina. The system can support any scheduler with little effort. By extending the Resource Specification Language (RSL) of the Globus Toolkit, which is used to identify user-specific resource requests, the ability to indicate a level of urgency for jobs is incorporated. A new "urgency" parameter is defined for three levels: *critical* (red), *high* (orange), and *important* (yellow). These urgency levels are guidelines that help resource providers enable varying site-local response protocols to differentiate potentially competing jobs. Users with valid SPRUCE tokens can simply submit their original Globus submission script with one additional RSL parameter (of the form "urgency = <level>") to gain priority access. Unlike the Globus RSL, local job queue submission interfaces, such as the PBS command *qsub*,[15] are often not trivially extended to accept new parameters. SPRUCE provides a *spruce_sub* script that accepts an additional command line parameter specifying the job's requested urgency level.

[15] PBS 'qsub' Job Submission Tool - http://www.clusterresources.com/torquedocs21/commands/qsub.shtml

At the core of the SPRUCE architecture is the invariant that urgent jobs may be submitted only while a right-of-way token is active. In order to support this, a remote authentication step is inserted into the job submission tool-chain for each resource supporting urgent computation. Since the SPRUCE portal contains the updated information regarding active sessions and users permitted to submit urgent jobs, it is also the natural point for authentication. When an urgent computing job is submitted, the urgent priority parameter triggers authentication. This authentication is not related to a user's access to resource, which has already been handled by the traditional Grid certificate or by logging into the Unix-based resource. Rather, it is a "Mother, may I" request for permission to queue a high-priority job. This request is sent to the SPRUCE portal, where it is checked against active tokens, resource names, maximum priority, and associated users. Permission is granted if an appropriate right-of-way token is active and the job parameters are within the constraints set for the token. All transactions, successful and unsuccessful, are logged.

Responding to Urgent Computation

All of the above works only when the resource providers support a set of urgent computing policy responses corresponding to different levels of requested urgencies. These policies can vary for every site based on comfort level. The SPRUCE architecture does not define or assume any particular policy for how sites respond to urgent computing requests. This approach complicates some usage scenarios, but it is unavoidable given the way we build Grids from distributed resources of independent autonomous centers and given the diversity of resources and operating systems available for computing. The SPRUCE architecture cannot simply standardize the strategy for responding to urgent computation. Instead, we are left with many possible choices for supporting urgent computation depending on the systems software and middleware as well as on constraints based on accounting of CPU cycles, machine usability, and user acceptance. Given the current technology for Linux clusters and more tightly integrated systems such as the Cray XT3 and the IBM Blue Gene, the following responses to an urgent computing request are possible:

- Scheduling the urgent job as "next-to-run" in a priority queue. This approach is simple and is highly recommended as a possible response for all resource providers. No running computation is killed; the impact on normal use is low. The urgent job will begin when all of the running jobs complete for a given set of CPUs. Unfortunately, this wait could go up to hours or even days.

- Suspending running jobs and immediately launching the urgent job. This will then force some memory paging, but the suspended job could be resumed later. Node crashes and failed network connections can be an obstacle in reviving suspended jobs. The benefit of this policy is that urgent jobs will begin almost immediately, making this option attractive in some cases.

- Forcing a checkpoint/restart of running jobs and re-queuing the urgent job as the next to run. This response is similar to the previous response but safely moves the checkpoint to a location where it can then be used to restart on alternative resources. Architectures supporting system-based checkpoint/restart can be used to support urgent computing where reliable. This checkpointing for large-memory systems could take 30 minutes or more depending on I/O and disk rates.

- Killing all running jobs and queuing the urgent job as next to run. Clearly this response is drastic and frustrating to the users who will lose their computation. Nevertheless, it will ensure that extremely urgent computations begin immediately after running jobs are killed.

Another factor in choosing the response policy is accounting and stakeholder accountability. Certain machines are funded for specific activities, and only a small amount of discretionary time is permitted. Furthermore, in order to improve fairness, some form of compensation (e.g., refunding CPU hours or a one-time higher priority rescheduling) could be provided to jobs that are killed to make room for an urgent job. Another idea is to provide discounted CPU cycles for jobs that are willing to be terminated to make room for urgent computations. In any case, resource providers are encouraged to map all three levels of urgency—critical, high, and important—to clearly defined responses.

SPRUCE Portal

The SPRUCE portal provides a single-point of administration and authorization for urgent computing across an entire Grid. It consists of three parts:

- The Web-based administrative interface allows privileged site administrators to create, issue, monitor, and deactivate right-of-way tokens. It features a hierarchical structure, allowing management of specific sub-domains.

- The Web service-based user interface permits token holders to activate an urgent computing session and to manage user permissions.
- The authentication service verifies urgent computing job submissions. A local site job manager agent queries the remote SPRUCE server to ensure that the submitting user is associated with an active token that gives permission to run urgent jobs on the given resource at the requested urgency.

Both the user interface and the authentication service communicate with the SPRUCE server via a Web services interface. External portals and workflows can become SPRUCE-enabled simply by incorporating the necessary Web service invocations. Users who prefer to use a Web-based interface can use the SPRUCE user portal. All users may monitor basic statistics such as the remaining lifetime of the token and the tokens with which they are currently associated. These interfaces need minimum additional training, making SPRUCE appropriate for emergency situations.

Proof of Concept

LEAD-SPRUCE Urgent Computing aim to predict severe weather during Spring 2007

LEAD applied some of its technology, in real time, for on-demand forecasting of severe weather during the 2007 National Oceanic and Atmospheric Administration (NOAA) Hazardous Weather Test Bed (HWT),[16] which is a multi-institutional program designed to study future analysis and prediction technologies in the context of daily operations. The HWT 2007 spring experiment wes a collaboration among university faculty and students, government scientists, NOAA and private forecasters to further our understanding and use of storm-scale, numerical weather prediction in weather forecasting. LEAD researchers and scientists in coordination with the SPRUCE Urgent Computing team were in a unique position to work with HWT participants to expose this technology to real-time forecasters, students, and research scientists. The 2007 effort addressed two important LEAD-related challenges: (1) the use of storm-resolving ensembles for specifying uncertainty in model initial conditions and quantifying uncertainty in model output, and (2) the application of dynamically adaptive, on-demand forecasts that are created automatically, or by humans, in response to existing or anticipated atmospheric conditions. A key aspect of the spring experiments was that the daily forecasts were evaluated not only by operational forecasters in the NOAA Storm Prediction Center (SPC) but by dozens of faculty and researchers who visited the Hazardous Weather Test Bed in Norman, Oklahoma during the seven-week period. SPC used a formal procedure to evaluate the daily forecasts (additional details may be found in [17]).

The LEAD participation in the HWT 2007 spring experiments is described in detail in [18]. Briefly, the effort sought an initial assessment of the following:

- Quantitative skill of storm-resolving ensemble forecasts compared to their deterministic counterparts at similar (experimental) and coarser (operational) grid spacings
- Predictability of deep convection and organized mesoscale convective systems
- Extent to which dynamically adaptive prediction leads to quantitative forecast improvements, possible negative consequences of adaptation, and an evaluation of strategies for making decisions regarding when, where and how to adapt

16 NOAA HWT – http://www.nssl.noaa.gov/hwt/

17 Weiss, S. J., Kain, J. S., Bright, D. R., Levit, J. J., Carbin, G. W., Pyle, M. E., Janjic, Z. I., Ferrier, B. S., Du, J., Weisman, M. L., Xue, M. "The NOAA Hazardous Weather Testbed: Collaborative testing of ensemble and convection-allowing WRF models and subsequent transfer to operations at the Storm Prediction Center," in *Proc. 22nd Conf. Wea. Anal. Forecasting/18th Conf. Num. Wea. Pred.*, Salt Lake City, Utah, Amer. Meteor. Soc., CDROM 6B.4, 2007.

18 Brewster, K. A., Weber, D. B., Thomas, K. W., Droegemeier, K. K., Wang, Y., Xue, M., Marru, S., Gannon, D., Alameda, J., Jewett, B. F., Kain, J. S., Weiss, S. J., Christie, M. "Use of the LEAD portal for on-demand severe weather prediction," *Sixth Conference on Artificial Intelligence Applications to Environmental Science, 88th Annual Meeting of the American Meteorological Society*, New Orleans, 2007.

- Ability of the TeraGrid to accommodate scheduled, on-demand and urgent computing applications that have strict quality-of-service requirements and use a substantial portion of available resources for an extended period of time

LEAD Scientists conducted on-demand, dynamically adaptive forecasts over regions of expected hazardous weather, as determined by severe weather watches and/or mesoscale discussions among scientists and forecasters at the SPC. The LEAD on-demand forecasts began in the first week of May and continued until June 8, 2007. LEAD scientists Drs. Dan Weber and Keith Brewster, interacted directly with the SPC forecasters and HWT participants to obtain the daily model domain location recommendations and launched the daily forecasts using the LEAD Portal. The 9-hour WRF forecasts consisted of 1000 km x 1000 km regions placed in an area of elevated risk of severe weather occurrence during the 1500-0000 UTC forecast period. The on-demand forecasting process depicted in Figure 5 illustrates the forecasters' interaction with the weather to create a customized forecast process not possible with the current real-time Numrical Weather Prediction NWP scheme.

The on-demand forecasts were initialized by using the 15 UTC LEAD ARPS Data Assimilation System ADAS[19] analysis or 3-hour North American Model-NAM forecast initialized at 1200 UTC interpolated to a horizontal grid spacing of 2-km. The ADAS analysis included radar data and other observations to update the 3-hour NAM forecast from the 12 UTC initial time. One advantage to this on-demand forecast system configuration is the potential rapid turnaround for a convective scale forecast using NAM forecasts updated with mid-morning observations. The period selected, from 1500 UTC to 0000 UTC, overlaps with part of the 2007 HWT forecast and verification period for the larger-scale, numerical forecasts using 2 km and 4 km grid spacing.

[19] Xue, M., Wang, D., Gao, J., Brewster, K., Droegemeier, K. K. "The Advanced Regional Prediction System (ARPS) – storm-scale numerical weather prediction and data assimilation," *Meteor. Atmos. Physics*, 82:139-170, 2007.

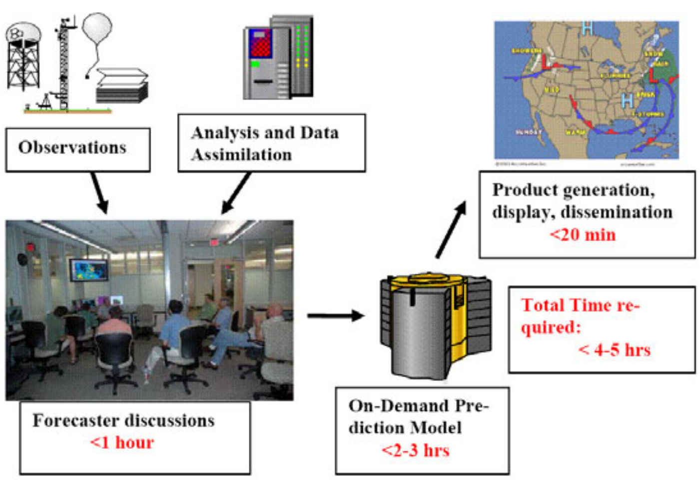


Figure 5. HWT 2007 spring experiments

Each day, one or more six- to nine-hour nested grid forecasts at 2 km grid spacing were launched automatically over regions of expected severe weather, as determined by mesoscale discussions at SPC and/or tornado watches, and one six- to nine-hour nested grid forecast, per day, at 2 km grid spacing was launched manually when and where deemed most appropriate. The production workflows were submitted to the computing resources at the National Center for Supercomputing Applications (NCSA). Because of the load on that machine, including other 2007 HWT computing resource needs, the workflow often waited for several hours in queues, before 80 processors were available to be allocated to the workflow. Moreover, the on-demand forecasts were launched based only on the severity of the weather. If we need a quick turnaround, computing resources have to be pre-reserved and idled, wasting CPU cycles and decreasing the throughput on a busy resource.

In order to tackle this problem, LEAD and SPRUCE researchers collaborated with the University of Chicago/Argonne National Laboratory (UC/ANL) TeraGrid resources to perform real-time, on-demand severe weather modeling. Additionally, the UC/ANL IA64 machine currently supports preemption for urgent jobs with highest priority. As an incentive to use the platform even though jobs may be killed, users are given a 10% discount from the standard CPU service unit billing. Deciding which jobs are preempted is determined by an internal scheduler algorithm that considers several aspects, such as the elapsed time for the existing job, number of nodes, and jobs per user. LEAD was given a limited number of tokens for use throughout the tornado season. The LEAD web portal allows users to configure and run a variety of complex forecast workflows. The user initiates workflows by selecting forecast simulation parameters and a region of the country where severe weather is expected. This selection is done graphically through a "mash-up" of Google maps and the current weather. We deployed SPRUCE directly into the existing LEAD workflow by adding a SPRUCE Web service call and interface to the LEAD portal. Figure 6 shows how LEAD users can simply enter a SPRUCE token at the required urgency level to activate a session and then submit urgent weather simulations.

Figure 6. Launching a urgent computing workflow using SPRUCE token

When isolated supercells were detected in upper midwest on June 7th, LEAD developers helped scientists get quick turnaround using SPRUCE critical priority queues on UC/ANL resources, preempting currently running jobs. The scientists subsequently analyzed the forecasts and compared the 20 UTC radar images for the HWT 2 km and 4 km forecasts (Figure 7). The LEAD on-demand shows distinct differences from other HWT numerical predictions (Figure 8) using the previous day's 21Z SREF data for the ARW2 and ARW4, the resolution and initial condition for the ARW3, and the 15 UTC data and resolution for the LEAD-ADAS urgent computing workflow execution.

Based on a comparison of just the two LEAD forecasts, the ADAS initialized forecast does a better job of handling the main line of convection during the period; in contrast, the NAM-initialized forecast is a little slow in initiating convection on that line in Iowa and produces less intense convection. However, the ADAS-initialized forecast produces some spurious convection early in the run that started in northeast Iowa and quickly moved northeast; the remains of that can be seen in the Upper Peninsula of Michigan at 20 UTC. It is possible that the ADAS analysis resulted in the net convective inhibition being too weak in those areas for this case. At 00 UTC, both LEAD forecasts had a weak secondary boundary to the southeast of the main line running from near Chicago across northern Illinois into northern Missouri. In the ADAS run this appears to be convection on an outflow boundary from the main line, whereas in the NAM-initialized run it seemed to have developed on its own as a weak line. It can be seen from this one example that each method of initialization of the model has its own unique characteristics and it is expected that, in time the best of each can be discerned and an intelligently constructed consensus will produce a superior forecast to what is currently available.

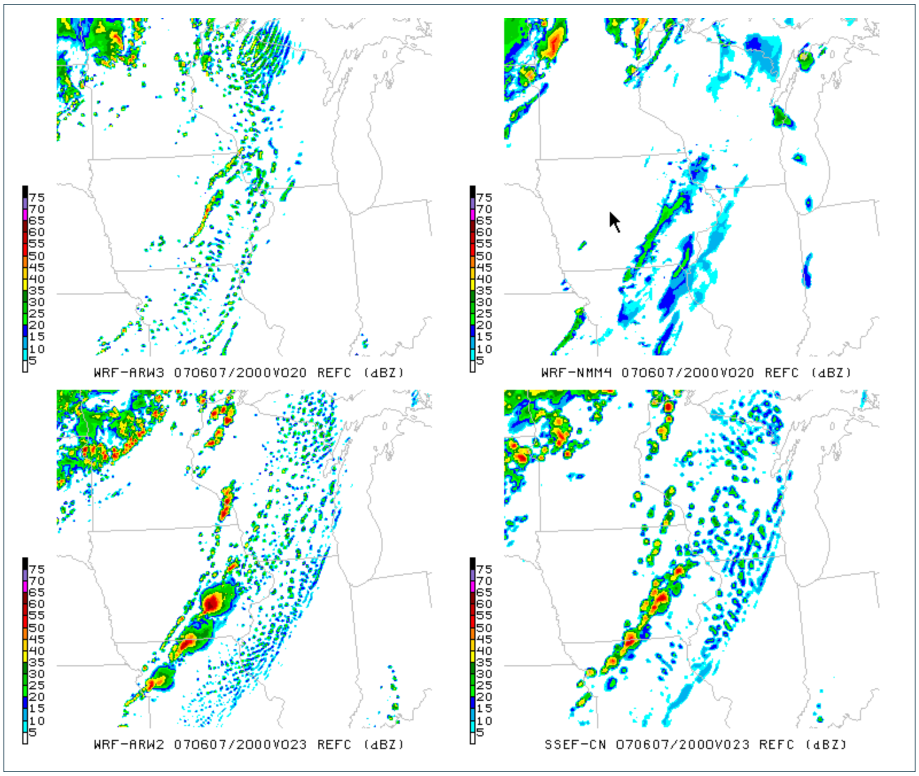


Figure 7. Comparison of four forecasts of radar composite valid at 20 UTC 07 June 2007: (top left) 3 km WRF ARW initialized at 00 UTC; (top right) 4-km WRF NMM initialized at 00 UTC; (bottom left) CAPS 2-km WRF initialized at 21 UTC 06 June 2007; (bottom right) 4 km WRF ARW SREF control forecast initialized at 00 UTC.

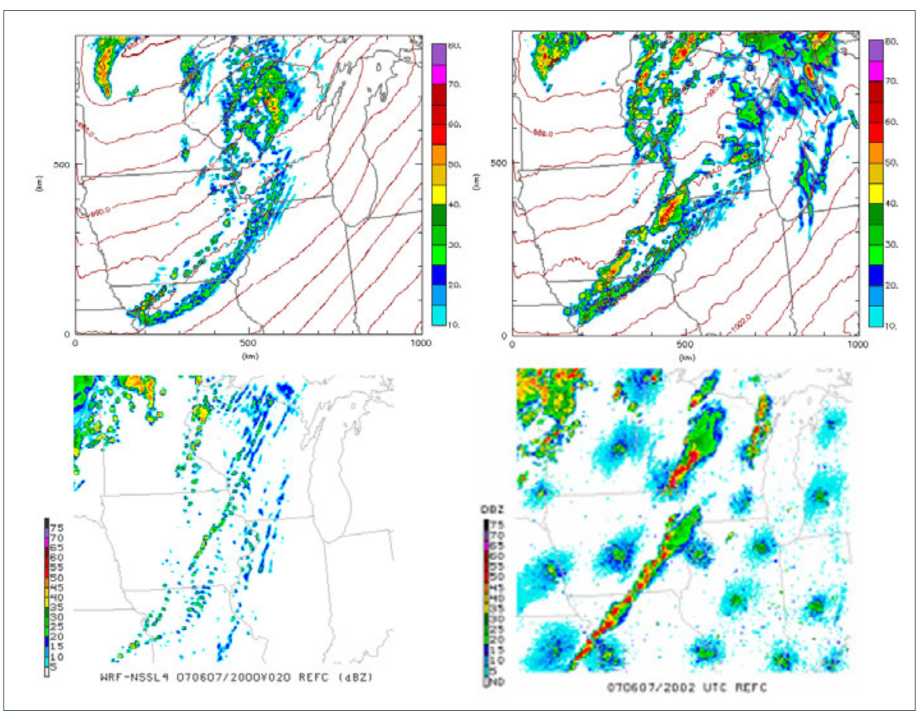


Figure 8. Comparison of four forecasts of radar composite valid at 20 UTC 07 June 2007: (top left) LEAD 2-km WRF initialized from the 3h forecast of the 1200 UTC NAM;(top right) LEAD 2-km WRF initialized from the 15 UTC ADAS analysis; (bottom left) NSSL 4-km WRF initialized at 00 UTC; and (bottom right) Observed radar composite at 2002 UTC.

Conclusion and Future Work

During spring 2007, LEAD cyberinfrastructure integrated with the SPRUCE Urgent Computing tools demonstrated on-demand, dynamically adaptive forecasts —those launched at the discretion of forecasters and over regions of expected hazardous weather as determined by severe weather watches and mesoscale discussions at the NOAA Storm Prediction Center. This collaboration was successful and used preemption capabilities on UC/ANL TeraGrid resources to meet the deadlines for critical runs.

For the 2008 Hazardous Weather Test Bed, we plan to repeat the experiment from 2007, adding 3-6 hours to the length of each on-demand forecasts to cover the evening active thunderstorm period as well as the afternoon. Additionally, we will study the processes by which forecasters determine when and where to (manually) launch on-demand forecasts. We also will continue to evaluate the tradeoffs between varying versus persistent model configurations. We strongly believe that by using urgent computing, the community can test and explore new ways to use applications and resources for critical situations.

Cyberinfrastructure for Coastal Hazard Prediction

Gabrielle Allen
Louisiana State University

Philip Bogden
Gulf of Maine Ocean Observing System

Tevfik Kosar
Archit Kulshrestha
Gayathri Namala
Sirish Tummala
Edward Seidel
Louisiana State University

Introduction

Around half the U.S. population live in coastal areas, at risk from a range of coastal hazards including hurricane winds and storm surge, floods, tornados, tsunamis and rising sea-level. While changes in sea-level occur over time scales measured in decades or more, other hazards such as hurricanes or tornados occur on timescales of days or hours, and early accurate predictions of their effects are crucial for planning and emergency response.

On the 29th August 2005, Hurricane Katrina (Fig. 1) hit New Orleans, with storm surge and flooding resulting in a tragic loss of life and destruction of property and infrastructure (Table 1). Soon after, Hurricane Rita caused similar devastation in the much less populated area of southwest Louisiana, and once again parts of New Orleans were under water. In both cases mandatory evacuations were enforced only 19 hours before the hurricanes made landfall. Speedier and more accurate analysis from prediction models could allow decision makers to evacuate earlier and with more preparation — and such hurricane prediction infrastructure is one goal of the SURA SCOOP Project.

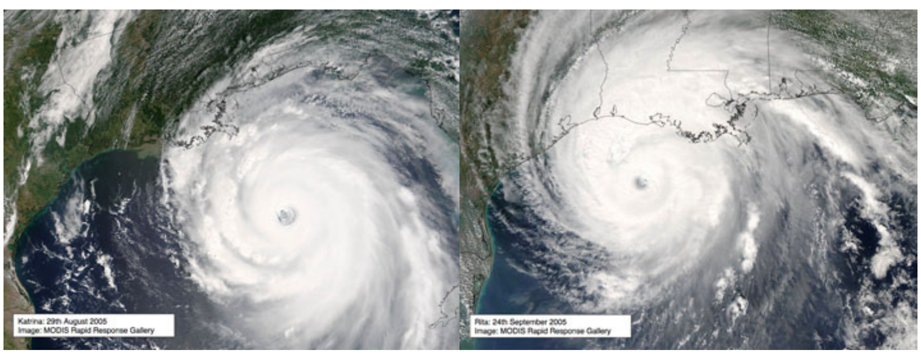


Figure 1. Satellite images of Hurricanes Katrina (left) and Rita (right) which made dramatic landfall on the southeast US coast in 2005. Katrina resulted in the loss of nearly 2000 lives and caused some $120 billion of property damage. The storm size at landfall was 460 miles, with 145mph winds (Category 3), and storm surges of up to 22 feet. [Image credits: MODIS Rapid Response Gallery]

The SCOOP Program[1 2] is creating an open integrated network of distributed sensors, data and computer models to provide a broad array of services for applications and research involving coastal environmental prediction. At the heart of the program is a service-oriented cyberinfrastructure, which is being developed by modularizing critical components, providing standard interfaces and data descriptions, and leveraging new Grid technologies and approaches for dynamic data driven application systems.[3] This cyberinfrastructure includes components for data archiving, integration, translation and transport, model coupling and workflow, event notification and resource brokering.

[1] SURA Coastal Ocean Observing Program (SCOOP) - http://scoop.sura.org/.

[2] Bogden, P. S., Gale, T., Allen, G., MacLaren, J., Almes, G., Creager, G., Bintz, J., Wright, L. D., Graber, H., Williams, N., Graves, S., Conover, H., Galluppi, K., Luettich, R., Perrie, W., Toulany, B., Sheng, Y. P., Davis, J. R., Wang, H., Forrest, D. "Architecture of a community infrastructure for predicting and analyzing coastal inundation," *Marine Technology Society Journal*, 41, No 1: 53-71, 2007.

[3] Allen, G., "Building a Dynamic Data Driven Application System for Hurricane Forecasting," *Proceedings of ICCS 2007*, Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 1034-1041, 2007.

Hurricane Katrina	Hurricane Rita
Date: 23-30 Aug, 2005	Date: 18-26 Sep, 2005
Category 3 landfall (peak winds: 145 mph) on 29th Aug, 6:10 am CDT, near Buras LA.	Category 3 landfall (peak winds: 120 mph) on 24th Sep, 2:40 am CDT, Texas Louisiana Border.
Voluntary evacuation New Orleans: 37 hours before landfall. Mandatory evacuation: 19 hours before landfall.	Mandatory evacuation Galveston: 19 hours before landfall.
Human casualties: 1836 approx. Property damage: 120 billion, New Orleans population reduced by 50%.	Property damage: 35 billion, 10% population displaced from Houston and Galveston.
Storm size (width) at landfall: 460 miles	Storm size (width) at landfall: 410 miles
Radius of hurricane force winds at landfall: 125 miles.	Radius of hurricane force winds at landfall: 85 miles.
Coastal storm surge: 18-22 feet.	Coastal storm surge: 15-20 feet.
Third most powerful hurricane to hit U.S coast. Most expensive. One of five deadliest.	

Table 1. Properties of hurricanes Katrina and Rita.

The SCOOP community currently engages in distributed coastal modeling across the southeastern US, including both the Atlantic and Gulf of Mexico coasts. Various coastal hydrodynamic models are run on both an on-demand and operational (24/7/365) basis to study physical phenomena such as wave dynamics, storm surge and current flow. The computational models, include Wave Watch 3 (WW3), Wave Model (WAM), Simulating Waves Nearshore (SWAN), ADvanced CIRCulation (ADCIRC) model, ElCIRC, and CH3D.[4] In the on-demand scenario, advisories from the National Hurricane Center (NHC) detailing impending tropical storms or hurricanes trigger automated workflows consisting of appropriate hydrodynamical models. The resulting data fields are analyzed and results are published on a community portal, and are also distributed to the SCOOP partners for local visualization and further analysis, as well as being archived for further use in a highly available archive.[5]

This article describes the technologies and procedures used for on-demand ensemble workflows that are used for predicting hurricane impacts, including urgent and prioritized workflows, local policies for compute resources, and providing access to urgent compute resources.

Hurricane Forecast Timeline

Hurricanes develop over warm ocean waters as an area of low pressure which turns into a tropical storm as the circular wind motion becomes organized. As the storm's wind surpasses 74 mph, the storm is classified as a Category 1 Hurricane. Hurricanes typically take around 3-5 days to develop. Coastal modelers begin running models and providing projected hurricane tracks as soon as there is indication that an area of low pressure will turn into a tropical storm. The National Hurricane Center (NHC) consolidates these various model runs and publishes a predicted storm track based on the models and past experience. Every few hours, these model predictions are also coupled with data from real observations, such as coastal data, buoy data, and the data provided by the hurricane hunter aircraft, and then the published track gets revised. This updated data is released to the community via advisories. The NHC publishes an advisory every six hours for every area of interest, specifying forecast track and strength forecast. The track data is picked up as soon as it is available and coastal scientists use this to run other wave, surge and inundation models.

[4] Wave Watch 3 (WW3) - http://polar.ncep.noaa.gov/waves/wavewatch/wavewatch.html,
Wave Model (WAM) - https://www.fnmoc.navy.mil/PUBLIC/WAM/wam det.html,
Simulating Waves Nearshore (SWAN) - http://www.wldelft.nl/soft/swan/,
ADvanced CIRCulation (ADCIRC) model - http://www.adcirc.org/,
ELCIRC - http://www.ccalmr.ogi.edu/CORIE/modeling/elcirc/,
CH3D - http://users.coastal.ufl.edu/~pete/CH3D/ch3d.html.

[5] MacLaren, J., Allen, G., Dekate, C., Huang, D., Hutanu, A., Zhang, C. "Shelter from the Storm: Building a Safe Archive in a Hostile World," In *Proceedings of the The Second International Workshop on Grid Computing and its Application to Data Analysis (GADA'05)*, Agia Napa, Cyprus, 2005. Springer Verlag.

Since the NHC issues an advisory every six hours during a storm event, modelers have a six hour window in which to generate all the results based on the previous advisory and use the results to "hot start" the next runs with data from the next advisory. Figure 2 shows the tracks for the 13th and 14th advisories issued by NHC at 11AM EDT and 5PM EDT, respectively, on August 26, 2005 during Hurricane Katrina. The images highlight that there can be considerable difference in projected paths between advisories. It is believed that the necessary lead time for evacuation is 72 hours and accurate predications about the impact of the storm are dependent on the accuracy of the forecast track. Hence the only way to overcome the inaccuracy in track prediction in order to provide reasonable guarantees of the predictions is to use ensemble modeling.

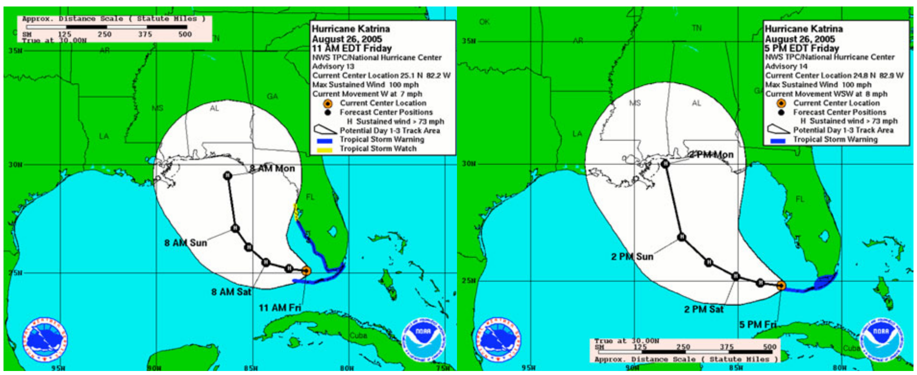


Figure 2. Hurricane Katrina Tracks for consecutive six hour advisories (13 and 14) illustrating the difference in projected paths. Advisory 13 (left) indicates landfall in Florida, while Advisory 14 (right) shows a projected landfall two states west in Mississippi. Image provided by NHC.

The development of low pressure areas and the timelines of these turning into hurricanes can vary from a few hours to a few days. A worst case scenario could have an advance notice of less than 12 hours, making it difficult to quickly obtain resources for an extensive set of investigatory model runs and also making it imperative to be able to rapidly deploy models and analysis data.

One obvious solution would be to dedicate a set of supercomputers for hurricane prediction. This would however require a significant investment to deploy and maintain the resources in a state of readiness, multiple sites would be needed to provide reliability, and the extent of the modeling would be restricted by the size of the machines.

A different solution is to use resources that are deployed and maintained to support other scientific activities, for example the NSF TeraGrid (which will soon be capable of providing over 1 PetaFlops of CPU power), the SURAgrid (developing a community of resources providers to support research in the southeast US), or the Louisiana Optical Network Initiative (LONI) (with around 100 TeraFlops for state researchers in Louisiana). Section 3 describes some of the issues involved when resources are provided to both a broad community of scientists and to support urgent computing.

On-demand Ensembles for Hurricane Forecast and Prediction

The impact of a hurricane is estimated from predicted storm surge height, wave height, inundation and other data. Coastal scientists provide estimations using a probabilistic ensemble of deterministic models to compute the probability distribution of plausible storm impacts. This distribution is then used to obtain a metric of relevance for the local emergency responders (e.g., the maximum water elevation or MEOW) and get to them in time to make an informed decision.[2] Thus, for every cycle there will be an ensemble of runs corresponding to the runs of all the models for each of the set of perturbed tracks. The SCOOP Cyberinfrastructure includes a workflow component to run each of the models for each of the tracks. The NHC advisory triggers the workflow that runs models to generate various products that are either input to other stages of the workflow or are final results that end up as visualized products. Figure 3 shows the SCOOP workflow from start to end and the interactions between various components.

During a storm event, the SCOOP workflow is initiated by an NHC advisory that becomes available on an FTP site that is continuously polled for new data. When new track data is detected, the wind field data is generated that is then pushed to the SCOOP archives using the Logical Data Manager (LDM) to handle data movement. Once the files are received at the archive, the archive identifies the file type and triggers the execution of the wave and surge models. The trigger invokes the SCOOP Application Manager (SAM) that looks up the Ensemble Description File (EDF) to identify the urgency and priority associated with the ensemble members. The urgency and priority of a run and how the SCOOP system uses this information are elaborated in the next section.

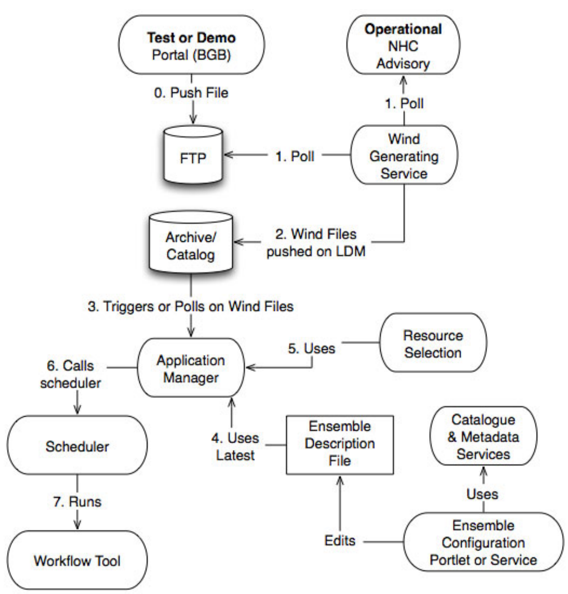


Figure 3. SCOOP workflow showing various components and their interaction. The workflow is initiated either by an NHC advisory, or through a configurable test/demo interface on the SCOOP portal.

Implementation of Urgent Capabilities in the SCOOP Workflow

The SCOOP infrastructure has been designed to support the urgent and dynamic deployment of ensembles of coastal models, where different members of the ensemble can have different urgency levels and different priority levels. Here we describe some of these new capabilities:

SCOOP Application Manager and Priority Aware Scheduler

During a hurricane event, different coastal hydrodynamics models are executed to predict quantities such as wave height and storm surge. The input to the ensemble of models is wind field data obtained from various sources including analytic models and fully 3D computational models.

The SCOOP Application Manager (SAM) allows ensembles to be created either from template configurations for a particular region or scenario, by customized user input through the SCOOP portal, or by future dynamically created ensembles built using storm and region properties. The SAM configures ensembles using parameters for each ensemble member which set: (i) the model to be run; (ii) the track used to calculate forcing wind fields; (iii) the level of urgency; (iv) and the priority of members within the ensemble.

The *urgency* parameter specifies the immediateness of the job to be performed and can be set to one of the keywords red, orange or green. An urgency level of red indicates that the job should run immediately, if necessary by preempting other jobs running on a resource. An urgency of orange indicates the job should run in high priority mode, for example as next to run on a batch queue. Urgency levels of green are used for "normal" jobs, which do not have special access to resources. A second integer-valued parameter for *priority* specifies the order in which jobs should be completed.

The ensemble configuration information is encoded into an XML file called the Ensemble Description File (Fig. 4). The EDF contains the science information about the ensemble member and does not contain any resource specific information.

```
<ensembledescription name="Default_Ensemble_2007"
lastModified="2007-08-28+09:00">
<storm num="12" name="Katrina" date="2005-08-28+09:00"/>
<ensemble size="10" creationTime="2007-04-24+12:29" lengthForecastHrs="120">
  <member id="1" urgency="1" priority="1">
    <track>e01</track>
    <model>WW3</model>
    <forcing>ANA</forcing>
    <region>Gulf</region>
    <config>0.2</config>
    <hotstart>No</hotstart>
    <comment></comment>
  </member>
  <member id="2" urgency="2" priority="2">
    <track>p02</track>
    <model>WW3</model>
    <forcing>ANA</forcing>
    <region>Gulf</region>
    <config>0.2</config>
    <hotstart>No</hotstart>
    <comment></comment>
  </member>
```

Figure 4. Portion of an Ensemble Description File encoding configuration information such as Urgency Level and Priority for each ensemble member.

The SCOOP Application Manager consists of multiple other components such as a resource broker that generates a list of resources and their capabilities such as access to on-demand queues, type of batch system, etc. The Application Manager then sends the list of available resources on which these ensemble members can be executed, which is used by the scheduler to dispatch the job to the resource.

The SCOOP Workflow system consists of two basic components: the SCOOP scheduler and the Workflow manager. The scheduler takes the track execution requests from SAM accompanied with resource availability information (ERDF). The track execution requests are generated by SAM using the information in the EDF files. These requests contain all the information the scheduler needs to generate, schedule, and execute ensemble track subworkflows. A sample track execution request is shown below in Figure 5.

```
[
  filename = "/data/SCOOP/MODEL WIND/ANA/UFL/WANAFp05-UFL ...";
  runtype = "WW3";
  priority = "1";
  urgency = "ondemand";
  x509proxy = "default";
]
```

Figure 5. Scheduler job submission request

The scheduler implements a multi-level priority queue with three levels of queuing: "on demand," "high priority," and "best effort." After the scheduler checks the urgency level and priority of each request, it places them in the correct queue in the correct order. The dispatcher selects the next request to be executed and creates a DAG (Directed Acyclic Graph) based workflow for each request. These DAG based workflows are then submitted to the workflow manager.

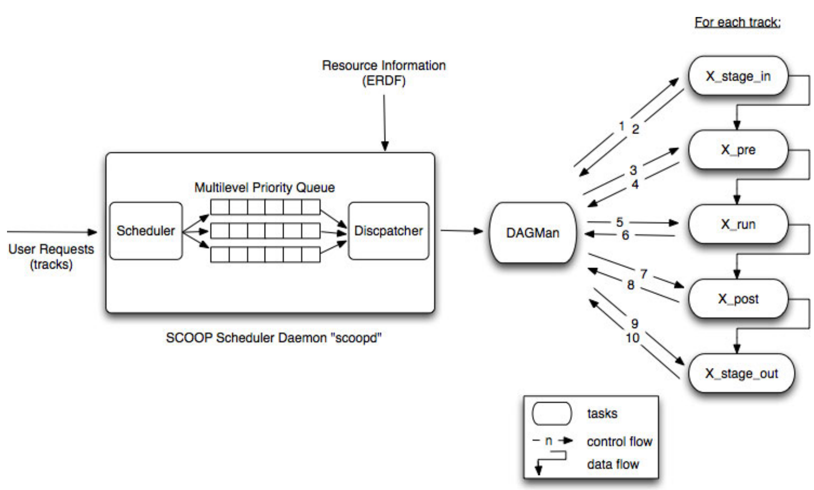


Figure 6. Control and Data flows for SCOOP models (replace X with ADCIRC, WW3, etc.)

As the workflow manager, we are using an enhanced version of Condor DAGMan. This version of DAGMan enables us to visually monitor the execution of the ensemble tracks from a web-based graphical interface. The status of each sub-task of each track, as well as the execution time of each step can easily be monitored using this interface. DAGMan submits these workflows to the queues specified by the Application Manager using Condor-G, which is a grid middleware used to submit jobs to globus gatekeepers. An illustration of the scheduler is shown in Figure 6.

The SCOOP scheduler queues are used mainly for internal prioritization and ordering, though the component also makes sure that the requests are mapped to the right kind of resources and queues. Once the jobs are dispatched, the quality of service received by the job is determined by the local resource policies and procedures, which are discussed in the next section.

Local Resource Policies and Procedures

The SCOOP system uses resources from various different grids such as the Louisiana Optical Network Initiative (LONI), SURAgrid and TeraGrid.[6] All three grids have a very different mode of operation and administration for instance the LONI grid is centrally administered and all policies are enforced on all resources. SURAgrid allows greater flexibility and control and policy-making is in the hands of individual resource providers. The TeraGrid is also composed of resources that are administered by the resource providers. All three grids implement access to on-demand resources in a variety of ways. LONI machines make use of SPRUCE and preempt queues to offer on-demand resources. SURAgrid uses Loadleveler's checkpoint restart mechanism to provide access to on-demand resources by suspending running jobs. TeraGrid has an entire cluster dedicated for on-demand jobs. For SCOOP, so far we have primarily used the LONI and SURAGrid on-demand resources.

[6] Louisiana Optical Network Initiative (LONI): http://www.loni.org/; NSF TeraGrid: http://www.teragrid.org/; Southeastern Universities Research Association Grid (SURAgrid): http://suragrid.sura.org/

On the LONI machines, the processors are divided into two groups, AIX based and Linux based. The machines run their independent schedulers and the processors on each resource are further subdivided into a preemptory pool and a dedicated pool. The preemptive queues feed the Preemptive pool of processors, and the dedicated queues feed the rest of the system. The checkpoint queues, which include all processors in both pools, can be used to submit system-wide jobs. The job restart information should be saved periodically for the jobs in the checkpoint queue as they may be preempted when an urgent job arrives in the preempt queue. On the LONI systems, this is left for the user to do along with choosing an appropriate queue for submission. On the LONI AIX frames, the preempt queue is allowed a maximum of 48 processors. The rest of the available processors are in the dedicated queue called workq. Jobs in the checkpoint queue (checkpt) can run on the entire machine. Also the maximum allowed wall clock time for jobs is typically longer in the workq and shorter in the preempt queues. SURAGrid, as part of the resource agreement with resource providers, has access to 20% of the resource that resources such as Ursa at GSU, offer as dedicated on-demand resources.

Immediate Access to Resources

Using preemption or other mechanisms to enable urgent simulations on supercomputers is not new. However, the traditional procedure for implementing preemption is to run such jobs in a special queue for which access is only granted for a fixed set of users. The policy, queue configuration, and set of users on each machine, particularly at each site, would need to be carefully negotiated (and usually frequently renegotiated). These procedures are usually not documented, thus it is difficult and time consuming to add new users for urgent computing, or to change the configuration of machines, for example to accommodate larger simulations. To resolve some of these issues, Special PRiority and Urgent Computing Environment (SPRUCE)[7] was implemented in the workflow. SPRUCE is a specialized software system to support urgent or event-driven computing on both traditional supercomputers and distributed Grids. It is being developed by the University of Chicago and Argonne National Laboratory and is presently functioning as a TeraGrid science gateway.

[7] SPRUCE: Special PRiority and Urgent Computing Environment - http://spruce.teragrid.org/

SPRUCE uses token based authentication system for resource allocation. Users are provided with right of way tokens, which are unique 16 character strings that can be activated through a web portal. The token is created on the CN value of the administrator. When a token is activated, there are other parameters that are set including

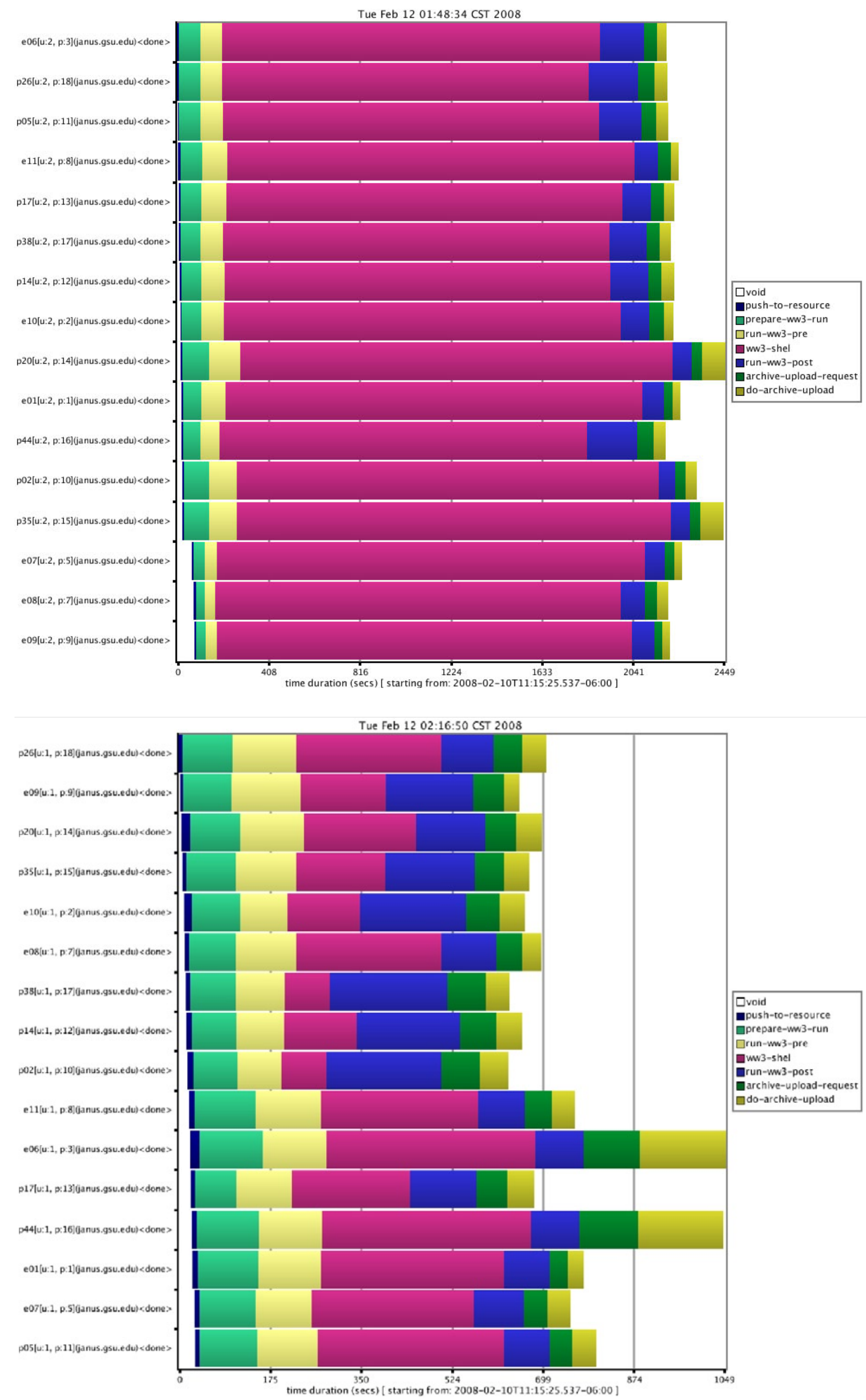
- Resources for urgent jobs: the activated token can be used to access any resource that is specified in this list and can be used by any person registered in it.
- Lifetime of the token: Each token is given a specific time period. Once active, the token can be used during this time period.
- Maximum urgency that can be requested, specified by the colors red, orange and yellow
- People to be notified when the token is used (e.g., the local administrators)

SPRUCE is a grid middleware that integrates with the resource manager on the system. When SPRUCE is installed, the resource manager is equipped with an authentication filter that checks for a valid token on the corresponding user name or the Distinguished Name (DN). If a token is activated, the job is submitted to a queue of higher priority level.

Results

The SCOOP on-demand system was demonstrated at the SuperComputing 2007 conference in Reno, Nevada using the resources of the SURAgrid and LONI. The demo illustrated how a hurricane event triggered the use of on-demand resources, and how the priority-aware scheduler was able to schedule the runs on the appropriate queues in the appropriate order. The guarantee that a member runs as soon as data for it has been generated makes it possible to provide a guarantee that the set of runs chosen as high priority runs will complete before the six hour deadline. Other work in benchmarking the models on different architecture platforms was used to estimate the amount of CPU time that a model would need to complete given the number of on-demand processors available.

SPRUCE was used to acquire the on-demand processors on some resources, and highlighted several different advantages. For example, the SCOOP workflow was no longer tied to being run by certain special users. This also meant that there was no need for negotiating access to the on-demand queues with the resource owners. Also, using SPRUCE provided the resource owners the ability to restrict the usage of the system in on-demand mode, and at the same time providing on-demand resources to anyone who needs them. In the past this could only be done by adding and deleting user access on a case-by-case basis. SPRUCE tokens can now be handed out to users by an allocation committee - thus removing the burden of evaluating the need for on-demand resources by users from the system administrators.



Figures 7(a - top) and 7(b - bottom). Ensemble member execution and wait times for (a)best-effort and (b) on-demand execution.

Figures 7(a) and 7(b) show the execution and wait times for the various stages of execution of the SCOOP workflow. Figure 7(a) shows the execution with only best-effort resources. The pink bars depict the execution and queue wait times of the core Wave Watch III execution on eight processors. It can be seen that the queue wait times account for most of the total time. Figure 7(b) depicts the ensemble execution using on-demand resources. In this case, 16 processors were available for on-demand use, hence two ensemble members ran simultaneously while others waited for these to finish.

A closer look at the 7(b) graph indicates that ensemble members p38 and p02 executed first followed by p14 and e10. The lengths of the pink bars for p14 and e10 are double that of p38 and p02 showing that they began execution right after the first two members finished execution. Comparing the two graphs, the last run finished in about 700 seconds when using on-demand resources compared to a time of about 2100 seconds without on-demand resources. It must be noted that the tests were performed using a short three hour forecast run, which completes in about 90 seconds on the chosen platform.

Conclusions

We have described an initial prototype for implementing urgent workflows for predicting the impacts of hurricanes, which include a new priority-aware scheduler, SPRUCE, for token-based authorization and carefully thought out policies on local resources. Despite the promise of this early work, there are many issues to research and resolve in the domain of urgent computing:

- Procedures for describing, negotiating and guaranteeing different levels of quality of service for urgent computing (e.g., run immediately, run within six hours, next to run).

- Economic models for sites, potentially around the world, to be compensated for providing their resources in an on-demand mode to those using urgent computing. Such an economic model for urgently available resources would then require scientists and authorities to in turn take into account economic cost when designing model suites and balance this against the potential risk.

- Policies and approaches for competing urgent scenarios, at the simplest level dealing with multiple concurrent hurricanes, but also handling for example modeling a Category 5 hurricane in Louisiana at the same time as a forest fire approaching Los Angeles in California.

- The configurations for model ensembles used today are still relatively static, and instead should be dynamically constructed to adapt to both the resource cost and availability and the physical situation (for example, a larger suite of models may be appropriate for a Category 5 hurricane than a Category 1 hurricane, or for a situation where a large number of ensembles are needed to provide appropriate confidence in results).

Acknowledgements
The SCOOP project is a large collaborative effort involving many organizations and researchers to build a cyberinfrastructure, and we would like to acknowledge their many contributions to the work described in this article. This work was funded by the Office of Naval Research (Award N00014-04-1-0721), the National Oceanic and Atmospheric Administration's NOAA Ocean Service (Award NA04NOS4730254) and the NSF DynaCode project (Award 0540374). SCOOP acknowledges computational resources provided by SURAgrid, LONI and TeraGrid.

Urgent Computing in Support of Space Shuttle Orbiter Reentry

Benjamin S. Kirk
NASA Lyndon B. Johnson Space Center

Grant Palmer
Chun Tang
NASA Ames Research Center

William A. Wood
NASA Langley Research Center

1. Introduction

On February 1, 2003, the Space Shuttle Orbiter *Columbia* suffered catastrophic structural failure during reentry, tragically killing all seven crewmembers on board. An extensive investigation into the accident was conducted in the ensuing months and identified that foam debris-induced damage to the reinforced-carbon-carbon wing, leading edge thermal protection system was the most probable root cause of the failure. During the course of the investigation, the *Columbia Accident Investigation Board* (CAIB) made a number of recommendations, which NASA agreed to implement before returning the Shuttle fleet to flight.

One of these recommendations, R3.8-2, addressed the need for computer models to evaluate thermal protection system damage that may result from debris impact. It reads:

> *Develop, validate, and maintain physics-based computer models to evaluate Thermal Protection System damage from debris impacts. These tools should provide realistic and timely estimates of any impact damage from possible debris from any source that may ultimately impact the Orbiter. Establish impact damage thresholds that trigger responsive corrective action, such as on-orbit inspection and repair, when indicated.*[1]

Implementing this recommendation was no small task, and involved hundreds of personnel from NASA, Boeing, The United Space Alliance, and other organizations. The result of this effort was the creation of a family of analysis tools that are used during the course of a Shuttle flight to assess the aerothermal, thermal, and structural impacts of a given damage site. These tools necessarily cross disciplines because, ultimately, the health of the vehicle depends on the coupled interaction of these three fields. The suite of tools spans the range of complexity from closed-form, analytical models to three-dimensional, chemical nonequilibrium Navier-Stokes simulation of geometrically complex configurations.

The focus of this article is to overview the damage assessment process, which is now a standard part of every Shuttle mission. The primary focus will be one aspect of this process, namely the rapid development of high-fidelity, aerothermal environments for a specific damage configuration using computational fluid dynamic (CFD) models.[2 3] The application of such models requires immediate and reliable access to massively parallel computers and a high degree of automation in order to meet a very aggressive schedule. The remainder of this article is outlined as follows: Section 2 provides an overview of the damage assessment process and required timeline, Section 3 describes the role of high-performance computing in rapidly generating aerothermal environments and associated challenges, Section 4 details the specific example of damage that occurred on STS-118 during the summer of 2007, and Section 5 provides some observations and general conclusions, which may be applicable to any process which demands urgent computational simulation.

[1] Gehman, H. W. et al, "Columbia Accident Investigation Board, Volume 1," Government Printing Office, August, 2003.

[2] Tang, C., Saunders, D., Trumble, K., and Driver, D., "Rapid Aerothermal Simulations of Damage and Repair during a Space Shuttle Mission," *AIAA Paper No. 2007-1783*, April 2007.

[3] Campbell, C., Driver, D., Alter, S., Fasanella, E., Wood, W., and Stone, J., "Orbiter Gap Filler Bending Model for Re-Entry," *AIAA Paper No. 2007-0413*, Jan. 2007.

2. Typical Flight Support Scenario

2.1 Data Acquisition

NASA and its commercial partners instituted a number of process and data-acquisition improvements during the two-and-a-half year lapse between the *Columbia* tragedy and *Discovery*'s historic return-to-flight mission. These improvements were specifically designed to identify and assess the severity of damage sustained to the thermal protection system during launch and on-orbit operations. The majority of such damage has historically been caused by foam or ice shed from the Orbiter/External Tank/Solid Rocket Booster ascent stack, but a limited amount of damage has also been attributed to micrometeor and orbital debris hypervelocity impacts.

A number of ground and air-based imagery assets provide video coverage of the vehicle's ascent to orbit. These imagery data are intensely reviewed during the hours after launch to identify potential debris strike events. Multi-band radar assets are also deployed on land and at sea during the launch phase to identify any off-nominal signatures, which may be related to debris impact. Additionally, the wing-leading-edge structural subsystem of each Orbiter was instrumented with a suite of accelerometers to aid in the detection of potential debris strikes.

Once the vehicle is in orbit, there are additional procedures that are executed to help identify potential damage. On the second day of flight, two crewmembers perform a detailed scan of the reinforced-carbon-carbon wing leading edge and nose cap. This scan is specifically designed to detect very small damages that could potentially cascade into a catastrophic failure sequence during the extremely high temperatures of reentry.

Prior to docking with the *International Space Station* on the third day of flight, the Orbiter executes a specific maneuver designed to aid in damage detection. The vehicle essentially performs a back flip while approximately 600 meters away from the Station. During this procedure two Station crewmembers perform photography of the vehicle. The imagery resolution is such that 7 cm damage can be identified anywhere on the vehicle, with damage as small as 2 cm identifiable in specific areas of interest. Imagery experts and hardware technicians provide the essential damage descriptions that are taken as input to a cross-disciplinary analysis. A composite lower-surface image that was obtained during *Discovery*'s return-to-flight is shown in Figure 1.

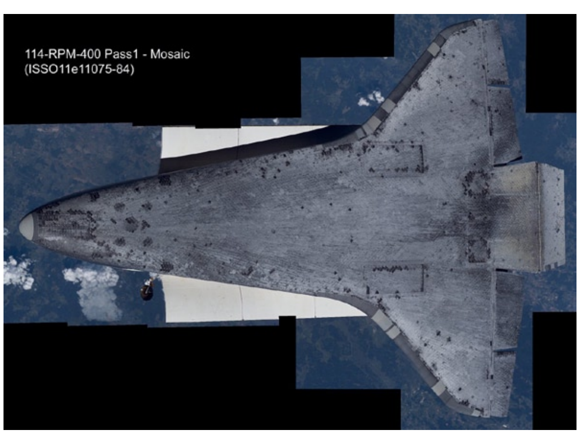


Figure 1. Composite image of Obiter lower surface taken during *Discovery*'s return-to-flight.

Each of the previously mentioned data acquisition tools is used on every mission. These data often provide the damage assessment team sufficient data to clear the vehicle for reentry. This is not always the case, however, and additional assets can be used to perform a focused inspection of a particular damage site that may be of concern. One such data set will be presented later.

2.2 Damage Assessment

It is at the end of flight day three when all of these data are available to analysts on the ground that the damage assessment process begins in earnest. From flight days three to five the coupled aerothermal-thermal-stress analysis process is performed for each identified damage site. The goal is to disposition each site as acceptable or unacceptable for reentry based on a set of well-defined structural and thermal limits. If a site is deemed unacceptable for reentry, it is the damage assessment team, in conjunction with on-orbit operations personnel, who work together to design, implement, and affect a repair procedure.

The first step in this process is to determine the aerothermal environment induced by a specific damage site. This includes any local changes in heat transfer that may result, as well as global effects such as early boundary-layer transition that may affect the downstream portion of the vehicle. Principally, empirically based correlations are applied to each site. These correlations are based on extensive test and analysis data that were performed pre-flight for physically relevant and geometrically similar conditions.[4] As with any empirical correlation, however, questions of suitability for a particular case invariably arise and must be addressed. This is the primary area where high-fidelity analysis is used during the nominal process.

[4] Everhart, J., "Supersonic/Hypersonic Laminar Heating Correlations for Rectangular and Impact-Induced Open and Closed Cavities," *AIAA Paper No. 2008-1283*, January 2008.

These aerothermal environments are then used as boundary conditions in transient thermal analysis for each site. The two primary goals of the thermal analysis are (i) to identify any material exceedances that may occur (e.g., exceeding allowable temperatures for aluminum structure), and (ii) to provide a damage-specific environment that can be used in stress analysis.

Assuming that a damage site has not exceeded material limits, the possibility still exists for local buckling due to thermal stress, for example. In this way the thermal environment is taken as input to a stress assessment that evaluates the potential for such effects. It is only when the end-to-end process is applied to a given site and presents no issues that the damage is deemed acceptable for reentry.

If the baseline process identifies an issue with any damage site, additional analysis is performed and the site is also considered as a candidate for on-orbit repair. It is in such high-risk scenarios that high-fidelity analysis and high-performance computing is particularly valuable.

2.3 Time Criticality

The intent is that the pre-flight mission timeline occurs uninterrupted while this process is executed on the ground. The nominal damage assessment process is scripted and well-rehearsed as to fit in a nominally 24-hour timeline. This is absolutely essential to mission success. In this way any damage that may require repair is identified and reported to the Mission Management Team by no later than the fifth day of flight. It is at this point during the flight when the schedule for the remainder of the mission is finalized. In particular, if a repair must be executed, it must be identified at this point so that adequate resources (e.g., breathable oxygen, water, spacecraft power) can be allocated. Identifying a problem late in the mission may be useless as there may not be adequate resources available to affect a repair.

3. High-Fidelity CFD Analyses and High-Performance Computing

It is in this compressed timeline that high-fidelity analysis must be performed if it is to be of value to the overall process. Additionally, the data environment is highly dynamic, as new characterizations of a damage site are continuously acquired. The timeline is such that a high-fidelity analysis must have a turnaround time of approximately eight hours or less for it to be useful. This requirement poses a number of challenges.

3.1 Analysis Tools & Supporting Infrastructure

The two primary CFD codes used by the reentry aerothermodynamic community at NASA are the *LAURA* and *DPLR* codes from Langley and Ames Research Centers, respectively. Both codes are block-structured, finite volume solvers that model the thermochemical nonequilibrium Navier-Stokes equations. *LAURA*[5] was originally written in Fortran 77 and was highly optimized for the vector supercomputers of the day. Subsequent modifications to the code have incorporated MPI for distributed-memory parallelism. *DPLR*,[6] written in Fortran 90, is a relatively newer code and was designed from its inception to use MPI on distributed-memory architectures with cache-based commodity processors.

The *Columbia* supercomputer, installed and maintained by the NASA Advanced Supercomputing Division (NAS), is the primary resource used for these analyses. *Columbia* is composed of 20 SGI Altix nodes, each of which contains 512 Intel, Itanium-2 processors. (*Columbia* was ranked 20th on the November 2007 Top 500 supercomputer ranking.) Prior to each launch, NAS personnel reserve one node for dedicated mission support and alert the user community that additional resources may be reallocated if necessary. *Columbia* is augmented with department-level cluster resources to provide redundancy (albeit at reduced capability) in case of emergency.

Institutional policies preclude major modification to either the software environment on the machines or the supporting network infrastructure in a "lockdown" period leading up to launch. This helps assure that resources are available and function as intended when called upon. This restriction prevents overzealous firewall modifications from precluding access to resources, to provide but one example.

3.2 Solution Procedure & Quality Control

High-fidelity analysis is engaged in earnest when a request is made from the Damage Assessment Team, which operates primarily at the Mission Control Center at Johnson Space Center in Houston, TX. A geometric description is provided to the analysis team that can be discretized into a computational grid. The analysis team has developed a number of rapid-turnaround, grid generation schemes based on both algebraic and partial-differential-equation techniques. In particular, Gridgen scripts have been created to model common types of damage scenarios (such as a cavity formed by debris impact or protruding gap filler). These technologies allow for high-quality, block-structured grids to be generated automatically in less than an hour.

The primary goal of these simulations is to determine the aerothermal environment induced by a given damage in relation to a reference, undamaged state. Accordingly, a number of simulations of the entire Orbiter have been pre-computed at relevant reentry conditions.[7 8] These results are archived on a 7 TB disk array at NAS and are mirrored across the agency for redundancy. These global solutions provide both a reference for undamaged configuration and a convenient starting point for local analysis.

[5] Gnoffo, P. A. and Cheatwood, F. M., "User's Manual for the Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA)," *NASA Technical Memorandum TM-4674*, National Aeronautics and Space Administration, 1996.

[6] Wright, M.J., Candler, G.V., and Bose, D., "Data-Parallel Line Relaxation Method for the Navier-Stokes Equations," *AIAA Journal*, Vol. 36, No. 9, 1998, pp. 1603-1609.

[7] Reuther, J., McDaniel, R., Brown, J., Prabhu, D., Saunders, D., and Palmer, G., "External Computational Aerothermodynamic Analysis of the Space Shuttle Orbiter at STS-107 Flight Conditions," *AIAA Paper No. 2004-2281*, June 2004.

[8] External Aerothermal Analysis Team, "Smooth Outer Mold Line Aerothermal Solution Database for Orbiter Windside Acreage Environments During Nominal Entry Conditions," *NASA Johnson Space Center Engineering Note EG-SS-06-01*, 2005.

Due to the predominantly hyperbolic nature of the governing equations, many areas on the vehicle are amenable to a local analysis approach, which considers only the damage site in isolation with upstream boundary conditions imposed using solutions from the reference dataset. Consequently, the resulting grid (hence the computing time) is significantly smaller than a global simulation of the entire vehicle with the damage site. This approach has proved invaluable in reducing the turnaround time of obtaining high fidelity CFD solutions (see reference [2] for more details). These improvements have enabled the mission support teams to either compute more cases or use less computing resources.

In recent Shuttle missions, ground-based arc-jet experiments have been performed, for evaluating material performance. However, any ground test can at best approximate the real aerothermal environment because no facility can duplicate the extreme flight conditions during reentry. Increasingly, this high-fidelity analysis capability has been used to help characterize ground-based testing and provides an invaluable tool for comparing and contrasting the test and flight environments.

Finally, rigorous quality control procedures have been implemented that fit into the aggressive timeline. This is a critical component of any computational simulation that is used in engineering design, but its importance is elevated for situations that are critical for risk analysis. Specifically, in this context an erroneous solution can be worse than just a waste of resources – it can actually be dangerous because simulation data are often used to judge the relative risk of two scenarios. Erroneous data could possibly lead decision-makers to actually choose the *riskier* of the two options. For the case of aerothermal analysis, a number of quantitative quality-control steps have been instituted to avoid this scenario. For example, simulations performed at the same conditions using both *LAURA* and *DPLR* are used as a quality control check. Additionally, metrics for quantifying the iterative solver for grid convergence are computed as part of the solution process. Finally, a team member who was not involved in producing the result subjects each simulation to a predefined quality control process.

3.3 Communication Channels & Result Dissemination

Our experience has pointed out the importance of direct communication channels between the analysis team and those who ultimately make decisions as a result of these analyses. As mentioned previously, the coordination between the aerothermal, thermal, and stress components of the damage assessment process occurs at Johnson Space Center in Houston. The individual analysts, however, are spread out on both coasts at Langley and Ames, and are therefore very much removed from the end-users of the data.

To address this communications gap we require that two members of the aerothermal CFD analysis team be present at Johnson Space Center throughout the damage assessment phase of the mission. Two individuals allow 24-hour coverage, which is essential for our application. These team members provide a critical liaison between the mission operations center and the analysts in the field. They essentially "speak the language" of the personnel performing the analysis and ensure that any known limitations or concerns are adequately presented to the larger damage assessment team. Additionally, the reverse communication channel is also satisfied, alerting the analysts to any additional data that may need to be incorporated into their high-fidelity simulation.

Equally important, we think, is that the analysts understand exactly how the data they are producing is used in the larger overall damage assessment process. We therefore require that each analyst observe the process first-hand before participating in a mission. This can be either through participating in a mission simulation or observing an actual mission on-site at the Johnson Space Center.

4. Illustrative Example: STS-118 Tile Damage

A piece of foam insulation broke off the Shuttle Endeavour during the ascent portion of the STS-118 mission in August 2007. The foam struck the thermal protection system tiles on the aft end of the windward surface of the vehicle. The impact caused a 7.5 cm long by 5 cm wide cavity that was discovered during the docking back-flip maneuver mentioned previously. Detailed imagery analysis was performed and indicated that the damage extended all the way through one of the affected 3 cm thick tiles.

Figure 2 shows the actual image taken during the maneuver that served as the initial input to the damage assessment process. This damage was of immediate concern because it potentially exposed the sensitive tile bond line to the heat of reentry. The 6-inch square tiles are primarily composed of silica and are bonded to the underlying aluminum skin with a felt "strain isolation" pad. This arrangement allows the structure and tiles to expand separately when heated during reentry.

Figure 2. STS-118 tile damage as seen during the rendezvous pitch maneuver

The damage configuration posed a number of potential problems that had to be addressed. The obvious question is whether or not this damage might allow a local structural burn-through and, if so, what the impacts would be. Additionally, since a portion of the insulating tile was removed, the bond line may overheat. This could allow the entire tile to be lost. Finally, increased local heating might cause excessive stress in the underlying aluminum skin due to thermal expansion.

Because of the potential severity of the damage, additional data were requested to help better characterize the damage. A detailed, three-dimensional scan of the damage was performed once the Orbiter was docked with the Space Station using Laser Doppler Range Imaging hardware. The data were downlinked to the damage assessment team in the form of a "point cloud" as shown in Figure 3.

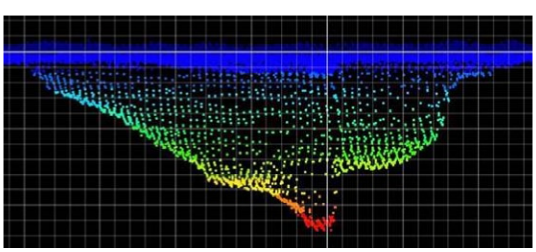


Figure 3. On-orbit LDRI scan of the STS-118 tile damage.

These data revealed that the cavity geometry was rather unique, and can be thought of as a cavity within a cavity. The deepest portion of the cavity extends to the insulation material between the two adjacent tiles. The neighboring tile is gouged roughly to the densified, lower portion of the tile. This configuration is fairly complex from an analysis point of view, and was somewhat out-of-family with the damages that had been used during experimental testing to develop a rapid-assessment, cavity-heating model. Consequently, the Orbiter Aerothermal CFD Team was asked to analyze the configuration to help provide the most accurate environment possible.

Due to initial uncertainty in the damage configuration, the analysis leads (at Johnson Space Center) requested that the analysis team (at Ames and Langley Research Centers) perform analysis on two different geometric configurations. Each configuration was analyzed at five different times along the predicted reentry trajectory. One of these configurations is shown on the left in Figure 4. The flow is from left to right, and the streamlines within the cavity are colored by temperature. The simulations showed that the majority of the high-energy flow bypassed the cavity altogether. Additionally, the critical exposed bond material was largely protected from the flow. The same set of streamlines is overlaid upon the scanned geometry and shown for reference in the right portion of the figure. The geometric similarity between the analyzed configuration and the true flight configuration is remarkable, and a unique capability offered by our urgent computing process put in place.

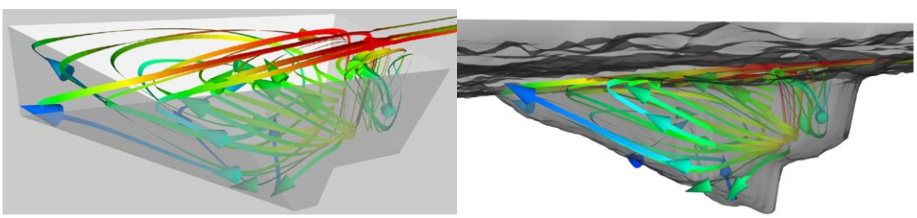


Figure 4. Configuration analyzed (left), with streamlines overlaid on actual damage (right).

The initial results from the damage assessment process were promising, but questions still remained about the material response. An arc-jet test was designed specifically to address this concern. The scanned damage was machined directly into an existing, pre-instrumented tile array and tested in an approximate flight environment. Arc-jets are particularly well suited to this type of testing, but a key question is how the test conditions relate to the true flight conditions. The high-fidelity analysis process was able to help here as well by simulating the as-tested configuration.

Based on the results of the complete aerothermal/thermal/stress analysis cycle, the decision was made to reenter the Orbiter as-is. The cavity is shown post-flight in Figure 5. It is clear from the figure that the damage did not progress during reentry. The correct decision was made.

It is worth mentioning, however, that a repair effort was being pursued in parallel to the nominal damage, assessment process. In the event a repair was warranted the urgent analysis process undoubtedly would have been engaged again to help assess and define repair

Figure 5. Post-flight image. Note the exposed red, felt-like material at the bottom of the cavity.

requirements. This places a large burden on the analysis community, as they must carefully evaluate many possible scenarios. However, given the compressed timeline imposed by manned spaceflight with limited consumables, there is no alternative to this seeming chaotic, parallel-path approach.

5. Observations & Lessons Learned

The rapid aerothermal analysis capability put in place during NASA's return-to-flight efforts has proven a, critical component of the damage assessment process which aims to assure the Shuttle is "go" for reentry. On multiple occasions, the Orbiter aerothermal analysis team has demonstrated the ability to meet the aggressive schedule demanded by real-time space operations support. In the case of STS-118, insights gained through this capability helped demonstrate that repair was not necessary, allowing the primary mission objectives to be achieved while ensuring crew safety. Given that Shuttle flights typically carry seven crewmembers, are estimated at $500 million a piece, and each Orbiter costs in excess of $1 billion, it is hard to underestimate the programmatic value of making the right decision in such circumstances.

Instituting this capability required the efforts of many people over a period of years. Key to its success was the dedication of these individuals and the tireless efforts of the overall team. The capability that has been put in place continues to evolve and benefits from experience gained each flight. We believe this is a critical aspect of using urgent computing to support high-stakes, real-time decisions. In our experience, it required three full-up system tests (in the form of pre-flight mission simulations) to effectively shake out the process, to illustrate strengths, and to identify and address weaknesses.

A highly automated process, robust quality control procedures, and dedicated, on-demand access to world-class resources are all prerequisites that help enable this capability. Equally important, and perhaps more surprisingly, are the human factors involved. Our experience is that timely generation of accurate results is critical, but proper interpretation and communication of those results is equally as critical. For our application, we require that analysis leads be co-located with the end users of the analysis data.

Acknowledgements

We would like to acknowledge the Orbiter Aerothermal CFD Team whose tireless efforts over a period of several years have led to this capability which has proven so useful. The current team, and those recently affiliated, includes: Chun Tang, Grant Palmer, Kerry Trumble, Todd White, David Saunders, Jay Hyatt, Bill Wood, Maria Pulsonetti, Bil Kleb, Rick Thompson, Peter Gnoffo, Adam Wise, Jose Padilla, David Driver, Dinesh Prabhu, Steve Alter, Max Larin, and Jeremiah Marichalar. Additionally, Cathy Schulbach, Chuck Niggley, and the NAS operations staff have been invaluable in their support of this team over the past three years.

Life or Death Decision-making: The Medical Case for Large-scale, On-demand Grid Computing

Steven Manos
Stefan Zasada
Peter V. Coveney
University College London

Overview

Patient-specific medicine is the tailoring of medical treatments based on the characteristics of an individual patient. Decision support systems based on patient-specific simulation hold the potential of revolutionising the way clinicians plan courses of treatment for various conditions, such as viral infections and lung cancer, and the planning of surgical procedures, for example in the treatment of arterial abnormalities. Since patient-specific data can be used as the basis of simulation, treatments can be assessed for their effectiveness with respect to the patient in question before being administered, saving the potential expense of ineffective treatments and reducing, if not eliminating, lengthy lab procedures that typically involve animal testing.

In this article we explore the technical, clinical and policy requirements for three distinct patient-specific biomedical projects currently taking place: the patient-specific modelling of HIV/AIDS therapies, cancer therapies, and addressing neuro-pathologies in the intracranial vasculature. These patient-specific medical simulations require access to both appropriate patient data and the computational and network infrastructure on which to perform potentially very large-scale simulations. The computational resources required are supercomputers, machines with thousands of cores and large memory capacities capable of running simulations within the time frames required in a clinical setting; the validity of results not only relies on the correctness of the simulation, but on its timeliness. Existing supercomputing site policies, which institute 'fair share' system usage, are not suitable for medical applications as they stand. To support patient-specific medical simulations, where life and death decisions may be made, computational resource providers must give urgent priority to such jobs, and/or facilitate the advance reservation of such resources, akin to booking and prioritising pathology lab testing.

1. Introduction

Recent advances in advance reservation and cross-site run capabilities on supercomputers mean that, for the first time, computation can be envisaged in more than a scientific research capacity so far as biomedicine is concerned. One area where this is especially true is in the clinical decision-making process; the application of large-scale computation to offer real-time support for clinical decision-making is now becoming feasible. The ability to utilise biomedical data to optimise patient-specific treatment means that, in the future, the effectiveness of a range of potential treatments may be assessed before they are actually administered, preventing the patient from experiencing unnecessary or ineffective treatments. This should provide a substantial benefit to medicine and hence to the quality of life of human beings.

Traditional medical practice requires a physician to use judgement and experience to decide on the course of treatment best suited to an individual patient's condition. While the training and experience of physicians hone their ability to decide the most effective treatment for a particular ailment from the range available, this decision making process often does not take into account all of the data potentially available. Indeed in many cases, the sheer volume or nature of the data available makes it impossible for a human to process as part of their decision making process, and is therefore discarded.

For example, in the treatment of HIV/AIDS, the complex variation inherent within data generated by analysis of viral genotype resulting in a prediction of phenotype (in terms of viral sensitivity to a number of treatments) makes the selection of treatment for a particular patient based on these predictions fairly subjective.

Patient-specific medical simulation holds the promise of evaluating tailored medical treatment based on the particular characteristics of an individual patient and/or an associated pathogen. Furthermore, approaches using simulation are based on the development of theories and models from which deductions can be made, as is the standard approach in the physical sciences and engineering. In reality, biology and medicine are still too poorly understood for deductive approaches to replace inductive ones so, in the foreseeable future, both will continue to sit side by side.[1] However for clinical acceptance, verification and validation of these techniques need to be addressed. The patient-specific simulation approach contrasts with more traditional use of computer systems to support clinical decision making, such as 'classic' expert systems, which take a Baconian approach, allowing a clinician to infer the cause of symptoms or the efficacy of a particular treatment regime based on historical case data. An example of such a system is the MYCIN expert system,[2] designed to suggest possible bacterial causes of a patient's infection by asking a clinician a series of 'yes' or 'no' questions.

[1] Coveney, P. V., Fowler, P. W. "Modelling biological complexity: a physical scientist's perspective," *The Journal of the Royal Society Interface*, 2005, 2, 267280.

[2] Shortlife, E. H. *Computer-Based Medical Consultations: MYCIN*, American Elsevier, 1976..

While the details vary widely between medical conditions, several basic elements are common to all fields of patient-specific medical simulation in support of clinical decision-making. Data is obtained from the patient concerned, for example from an MRI scan or genotypic assay, which is used to construct a computational model. This model is then used to perform a single simulation, or can form the basis of a complex workflow of simulations of a proposed course of treatment; for example, molecular dynamics simulations of drugs interacting with a range of viral proteins, and the results of the simulation are interpreted to assess the efficacy of treatment under consideration. The use a of simulation to assess a range of possible treatments based on data derived from the patient who is to be treated will give the physician the ability to select a treatment based on prior (simulated) knowledge of how the patient will respond to it.

2. Infrastructure requirements

The patient-specific medical simulation scenarios touched on above require access to both appropriate patient data and to the infrastructure on which to perform potentially very large numbers of complex and demanding simulations. Resource providers must furnish access to a wide range of different types of resource, typically made available through a computational grid, and to institute policies that enable the performance of patient-specific simulations on those resources. A computational grid refers to a geographically distributed collection of supercomputing resources, typically connected by high-capacity networking infrastructure, and we define grid computing as *distributed computing conducted transparently across multiple administrative domains.*[3] For the purpose of this article, grids can also include other resources, such as medical imaging equipment and data visualisation facilities.

[3] Coveney, P. V. "Scientific Grid Computing," *Philosophical Transactions of the Royal Society A*, 2005, 363(1833), 1707-1713.

In order to make patient-specific simulations useful to a physician, results need to be obtained within a clinically useful timeframe, which ranges from instantaneous results to weeks, depending on the scenario. In addition to expediency of access to patient data, consideration must also be given to policy and procedures that ensure maintenance of patient confidentiality. For such an enterprise to succeed, grid computing will need to focus not only on the provision of large 'island' compute machines but also on the performance characteristics of the networks connecting them. The process of clinical decision making, requiring access to relevant data, timely availability of computational results, visualisation, data storage, and so on, requires infrastructure that can facilitate the transfer of gigabytes of data within clinically relevant timeframes.

2.1 The need for resource reservation and urgent computing

When used as part of the clinical decision making process, computational resources often need to support more exotic scheduling policies than simple first come, first served, batch scheduling, which is the typical scenario seen in high-performance research computing today. Clinicians who require interactive access to machines (for example, for steering and visualisation, as well as cross-site applications, for example when performing cerebral blood flow simulations using the HemeLB code discussed later in this article) also need to be able to both schedule time on specific resources - compute and networking - and access tools to allow them to easily co-reserve those resources, so they are available when needed. This in turn leads to a demand on resource providers to implement policies and tools that allow such reservations to be made as needed and when required, so that such methodologies can be incorporated into a user's normal research activities, rather than just providing such facilities on an *ad hoc* basis. Moreover, the resources provided by a single grid may not always be sufficiently powerful or appropriate to run large-scale distributed models, and resources provided by multiple grids may need to be federated in order for a particular investigation to be conducted.

If these resources need to be used interactively, the problem of reservation becomes compounded since each grid has its own policies and systems for making advanced reservations, if it has any at all. Additionally, the high performance network provision between grids may also be limited or non-existent. Nevertheless, such obstacles must be overcome to make efficient use of available federated systems.

The key factor that transcends all of the current patient-specific medical simulation scenarios described in this article is the need to turn simulations around fast enough to make the result clinically relevant. This in turn means that the results can be obtained and interpreted within a timeframe on which a clinical decision is made; for example, in the HIV case described later this is roughly two weeks – the time it takes to get the results of a genotypic assay. To achieve the required turn around factor, such simulations cannot be run in a resource's normal batch mode; they need to be given a higher priority and they require some form of on-demand computing to succeed.

3. Different paradigms for on-demand computing

We consider two different urgent computing paradigms in order to make use of supercomputing resources, provided by a grid, in clinical scenarios; the advance reservation of CPU time on a compute resource at some specific point in the future, and the pre-emption of running jobs on a machine by some 'higher priority' work. The two paradigms apply to slightly different situations; the former would be of most use when a clinician knows in advance that a simulation needs to be performed at a specific time, for example an interactive brain blood-flow simulation run for a surgeon while planning or conducting a surgical procedure. The second paradigm is most useful when a medical simulation needs to be performed urgently, but the need for the simulation is not known in advance. An example of this latter simulation would be where a clinician encounters a HIV patient and urgently needs to compute the efficacy of a series of inhibitor drugs in relation to the patient's specific HIV mutation.

There is crossover between the two different urgent computing paradigms considered, and to a certain extent both apply to each of the situations mentioned. We favour a combination of both paradigms, to give clinicians and scientists the greatest amount of flexibility possible for the work they need to conduct. We discuss the technical aspects of the two paradigms in greater detail below.

3.1 Advance reservation

Several systems exist to allow users to easily co-reserve time on grid resources. GUR (Grid Universal Remote)[4] is one such system, developed at San Diego Supercomputer Center (SDSC). The GUR tool is a python script, which builds on the *ssh* and *scp* commands to give users the ability to make reservations of compute time and co-schedule jobs. GUR is installed on the SDSC, National Center for Supercomputing Applications (NCSA) and Argonne National Laboratory (ANL) TeraGrid IA-64 systems, and is expected to be available at other TeraGrid sites soon.

[4] Marcusiu, D., Margo, M., Yoshimoto, K., Kovatch, P. "Automatic Co-Scheduling," *TeraGrid Conference*, June 12-15, Indianapolis, USA, 2006.

HARC (Highly Available Robust Co-scheduler) is one of the most robust and widely deployed open-source systems that allows users to reserve multiple distributed resources in a single step.[5] These resources can be of different types, including multiprocessor machines and visualisation engines, dedicated network connections, storage, the use of a scientific or clinical instrument, and so on. HARC can be used to co-allocate resources for use at the same time, for example, within a scenario in which a clinical instrument is transferring data over a high-speed network link to remote computational resources for real-time processing. It can also be used to reserve resources at different times for the scheduling of workflow applications. We envisage clinical scenarios within which patient-specific simulations can be timetabled and reserved in advance, via the booking of an instrument, the reservation of network links and storage facilities, followed by high-end compute resources to process data, and finally the use of visualisation facilities to interpret the data for critical clinical decisions to be made.

[5] MacLaren, J. "HARC: The Highly-Available Resource Co-allocator," In *Proceedings of GADA07*, LNCS 4804 (OTM Conferences 2007, Part II). Springer-Verlag, 2007.

Currently, HARC can be used to book computing resources and lightpaths across networks based on GMPLS (Generalised Multi-protocol Label Switching) with simple topologies. HARC is also designed to be extensible, so new types of resources can be easily added; it is this that differentiates HARC from other co-allocation solutions. There are multiple deployments of HARC in use today: the US TeraGrid, the EnLIGHTened testbed in the United States, the regional North-West Grid in England, and the National Grid Service (NGS) in the UK. We use HARC on a regular basis to make single and multiple machine reservations, within which we are able to run numerous applications including HemeLB (see Section 4.1).

3.2 Emergency Computing

SPRUCE (SPecial PRiority and Urgent Computing Environment)[6] is an urgent computing solution that has been developed to address the growing number of problem domains where critical decisions must be made quickly with the aid of large-scale computation. SPRUCE uses simple authentication mechanisms, by means of transferable 'right of way' tokens. These tokens allow privileged users to invoke an urgent computing session on pre-defined resources, during which time they can request an elevated priority for jobs. The computations can be run at different levels of urgency; for example, they can have a 'next to run' priority, such that the computation is run once the current job on the machine completes, or 'run immediately,' such that existing jobs on the system are removed, making way for 'emergency' computation in a pre-emptive fashion, the most extreme form of urgent computing. The neurovascular blood-flow simulator, HemeLB (discussed in Section 4.1) has been used with SPRUCE in a 'next to run' fashion on the large scale Lonestar cluster at the Texas Advanced Computing

[6] Beckman, P., Beschastnikh, I., Nadella, S., Trebon, N. "Building an Infrastructure for Urgent Computing," *IOS*, 2007.

Center (TACC), and was demonstrated live on the show floor at SuperComputing 2007, where real-time visualisation and steering were used to control HemeLB within an urgent computing session.

The TeraGrid also provides a contrasting solution to the need to run urgent simulations on its resources. SDSC provide an 'On-Demand' computer cluster, made available to researchers via the TeraGrid, to support scientists who need to make use of urgent scientific applications. The cluster is configured to give top priority to urgent simulations, where results of the simulation are needed to plan responses to real-time events. When the system is not being used for on-demand work, it runs normal batch compute jobs, similar to the majority of other TeraGrid resources. Many of the current urgent scenarios considered cover the need to anticipate the effects of natural disasters, such as earthquakes and hurricanes, by performing simulations to predict possible consequences while the event is actually happening. Patient-specific medical simulations present another natural set of use cases for the resource.

4. Patient-specific computational medicine

In this section, we discuss three examples of patient-specific medicine where computational approaches are showing promise. Although the overall pathologies in each case are similar from patient to patient, the underlying details of the pathology can differ dramatically. In the case of HIV/AIDS and cancer treatments, the underlying mutations of these conditions are related to the genotype of the patient, and in the case of neurovascaular pathologies, the cerebral vascular structure differs considerably between individuals, so that each person will exhibit different blood flow dynamics. All these cases exemplify 'patient-specific' approaches, since the treatment is based on genotypic and/or phenotypic information obtained from the patient.

4.1 Grid enabled neurosurgical imaging using simulation

Cardiovascular disease is the cause of a large number of deaths in the developed world.[7] Cerebral blood flow behaviour plays a crucial role in the understanding, diagnosis and treatment of this disease. The problems are often due to anomalous blood flow behaviour in the neighbourhood of bifurcations and aneurysms within the brain; however, the details are not very well understood.

Experimental studies are frequently impractical owing to the difficulty of measuring flow behaviour in humans; however, X-ray and magnetic resonance imaging angiography (MRA) enable non-invasive static and dynamical data acquisition.[8] Indeed, some studies have revealed relationships between specific flow patterns around walls and cardiovascular diseases such as atherosclerosis.[9]

Today, such imaging methods represent a very important tool for diagnosis of various cardiovascular diseases, together with the design of cardiovascular reconstructions and devices for the enhancement of blood flow. Notwithstanding these advances in measurement methods, modelling and simulation undoubtedly have a crucial role to play in haemodynamics. Simulation, for example, furnishes the clinician with the possibility of performing non-invasive virtual experiments to plan and study the effects of certain courses of (surgical) treatment with no danger to the patient, offering support for diagnosis, therapy and planning of vascular treatment.[10] Modelling and simulation also offer the prospect of providing clinicians with virtual patient-specific analysis and treatments.

[7] The World Health Report. "Reducing risks, promoting healthy life," 2002; http://www.who.int/whr/en/.

[8] Goyen, M., Laubb, G., Ladd, M. E., Debatin, J. F., Barkhausen, J., Truemmler, K.-H., Bosk, S., Ruehm, S. G. "Dynamic 3D MR angiography of the pulmonary arteries in under four seconds," *Journal of Magnetic Resonance Imaging*, 2001, 13, 372377.

[9] Thubrikar, M. J., Robicsek, F. "Pressure-Induced Arterial Wall Stress and Atherosclerosis," *Annals of Thoracic Surgery*, 1995, 59, 15941603.

[10] Taylor, C. A., Draney, M. T., Ku, J. P., Parker, D., Steele, B. N., Wang, K., Zarins, C. K. "Predictive medicine: Computational techniques in therapeutic decision-making," *Computer Aided Surgery*, 1999, 4, 231247.

Reaching the goal of blood flow modelling and simulation is dependent on the availability of computational models of sufficient complexity and power. Furthermore, the neurovascular system varies between every single person, and so any computational approach will require patient-specific data. The aforementioned imaging techniques are used to provide data for such simulations. Furthermore, the computational fluid 'solver' used must itself be numerically highly efficient and provide scientists and neurosurgeons with the ability to manipulate and visualise the associated large data sets. The intricate geometry of the fluid vessels and treatment of fluid boundary conditions at such walls are also very difficult for traditional continuum fluid dynamics models to handle. Instead, a lattice-Boltzmann (LB) method, coined HemeLB, offers an attractive alternative. A major feature of HemeLB is real-time rendering and computation; fluid flow data is rendered *in-situ* on the same processors as the LB code, and sent, in real-time, to a lightweight client on a clinical workstation (Figure 1). The client is also used to steer the computation in real time, allowing the adjustment of physical parameters of the neurovascular system, along with visualisation-specific parameters associated with volume rendering, isosurface rendering, and streamline visualisation.

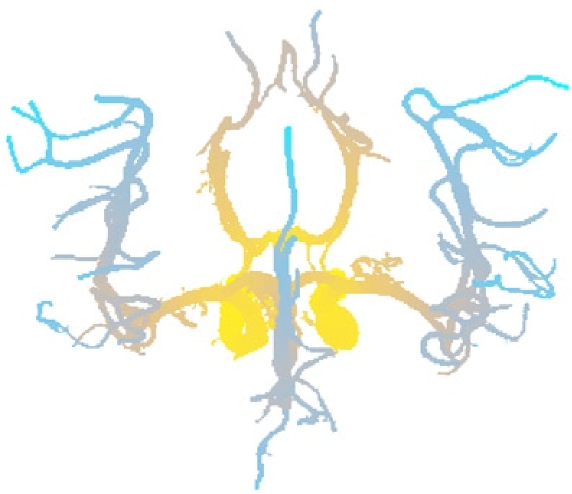


Figure 1. A snapshot of the real-time simulation and visualisation of neurovascular blood flow using HemeLB, as displayed in the client software. Here the pressure field is shown, where increasing pressure is depicted from blue to yellow. This view can be rotated and adjusted in real time using computational steering.

HemeLB is intended to yield patient-specific information, which helps plan embolisation of arterio-venous malformations and aneurysms, amongst other neuropathologies. Using this methodology, patient-specific models can be used to address issues with pulsatile blood flow, phase differences and the effects of treatment, all of which are potentially very powerful both in terms of understanding neurovascular patho-physiology and in planning patient treatment.

The software environment used in this project aims to bring to the forefront details and processes clinicians need to be aware of, such as (i) the process of image segmentation to obtain a 3D neurovascular model, (ii) the specification of pressure and velocity boundary conditions, and (iii) the real-time rendered image (Figure 2). Clinicians are not concerned with *where* simulations are running, nor the details of reservations, thus features such as advanced reservations and emergency computing capabilities, job launching and research selection are all done behind the scenes. This environment is particularly important given the time scales involved in the clinical decision making process in the treatment of aterio-venous malformations and aneurysms. From the acquisition of a 3D dataset (which is typically 2 to 4 GB in size), to the next embolisation, a time scale of 15 to 20 minutes is typical, and for such computational approaches to be clinically relevant, we have to fit into this time scale. There are also preventative scenarios that can be envisioned; patients could be subjected to such simulations in advance of vascular pathologies developing, averting future problems with interventional treatments.

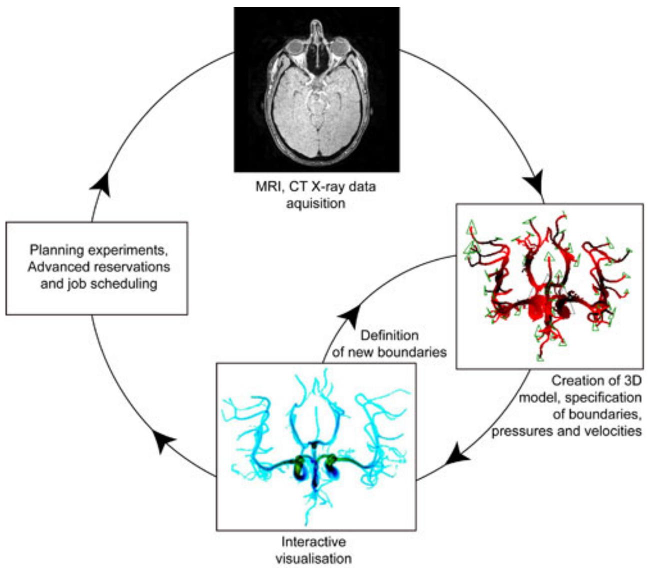


Figure 2. The workflow cycle for the simulation of neurovascular blood flow from the viewpoint of a clinician. Patient-specific data is acquired in the form of MRI or X-ray CT scans, and a 3D model of the neurovascular structure is created. Real-time simulation and interactive visualisation is used to examine blood flow through the brain. At this point, new boundaries and other physical changes can be made to the vascular structure, where the clinician can observe changes in pressure and velocity throughout the vasculature prior to operating. Many of the details, such as machine reservations and job submission are completely hidden from the surgeon/clinician/consultant.

4.2 Patient-specific HIV drug therapy

A major problem in the treatment of AIDS is the development of drug resistance by the human immuno-deficiency virus (HIV). HIV-1 protease is the enzyme that is crucial to the role of the maturation of the virus and is therefore an attractive target for HIV/AIDS therapy. Although several effective treatment regimes have been devised which involve inhibitors that target several viral proteins,[11] the emergence of drug resistant mutations in these proteins is a contributing factor to the eventual failure of treatment.

Doctors have limited ways of matching a drug to the unique profile of the virus as it mutates in each patient. A drug treatment regimen is prescribed using knowledge-based clinical decision support software, which attempts to determine optimal inhibitors using existing clinical records of treatment response to various mutational strains. The patient's immune response is used as a gauge of the drug's effectiveness and is periodically monitored so that ineffective treatment can be minimised through an appropriate change in the regimen. The FP6 EU project 'Virolab' is attempting to enhance the efficacy of clinical decision support software, through a unification of existing databases, as well as integration with means of assessing drug resistance at the molecular level.[12]

[11] Alfano, M. and Poli, G. (2004). "The HIV Lifecycle: Multiple Targets for Antiretroviral Agents," *Drug Reviews Online*. 1:83-92.

[12] Sloot, P.M.A., Boukhanovsky, A.V., Keulen, W., Tirado-Ramos, A., Boucher, C.A. "A Grid-based HIV Expert System," *Journal of Clinical Monitoring and Computing*, vol. 19, nr. 4-5, October 2005.

At the molecular level, it is the biochemical binding affinity (free energy) with which an inhibitor binds to a protein target that determines its efficacy. Experimental methods for determining biomolecular binding affinities are well established and have been implemented to study the *in-vitro* resistance conferred by particular mutations. These in turn add invaluable information to any decision support system, but are limited as studies are performed usually on key characteristic mutations and not with respect to the unique viral sequence of a patient. An exhaustive experimental determination of drug binding affinities in a patient-specific approach is far too costly and time-consuming to perform in any clinically relevant way.

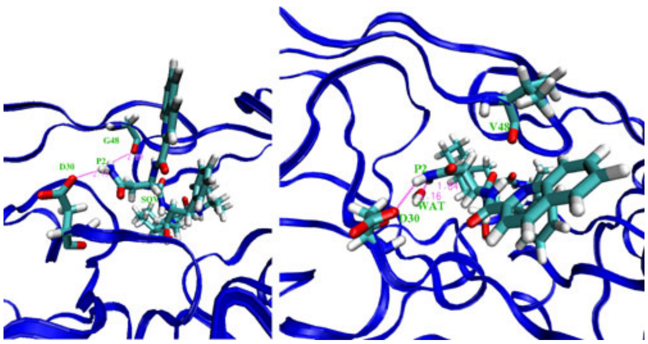


Figure 3. Snapshots of saquinivar binding with the active site of the L90M mutant wildtype of HIV protease (left), and the G48V HIV mutation (right). This image is taken from [13] with permission.

Computational methods also exist for determining biomolecular binding affinities. In a recent study,[13] the effectiveness of the drug saquinavir was tested against the wildtype HIV-1 protease, along with three drug-resistant strains using free energy methods in molecular dynamics (MD) simulations (Figure 3). The protocol implemented by the study gave accurate correlations to similar experimentally determined binding affinities. Furthermore, the study made use of a tool, the Binding Affinity Calculator (BAC), for the rapid and automated construction, deployment, implementation and post processing stages of the molecular simulations across multiple supercomputing, grid-based resources. The BAC is built on top of the Application Hosting Environment (AHE),[14] a web services environment designed to hide the complexity of application launching from the scientific end user of the grid. The AHE makes use of Globus Toolkit versions 2 and/or 4 for job submission, and GridFTP for data transfer between resources.

BAC automates binding affinity calculations for all nine drugs currently available to inhibit HIV-1 protease and for an arbitrary number of mutations away from a given wildtype sequence. Although the applicability of the method in the saquinavir-based study still needs to be established for all other inhibitors, the scope of BAC is enormous as it offers an automated *in-silico* method for assessing the drug resistance for any given viral strain. The turn around time using BAC for such studies is seven days (per drug/protease system) with optimal computational resources; this is more than suitable for the timescales required for effective clinical decision support. Given enough computational power such that binding affinity calculations can be routinely applied, the potential to achieve patient-specific HIV decision support may then become realistic.

13 Stoica, I., Sadiq, K. S., Coveney, P. V. "Rapid and Accurate Prediction of Binding Free Energies for Saquinavir-Bound HIV-1 Proteases," *J. Am. Chem. Soc.*, January, 2008, DOI: 10.1021/ja0779250

14 Coveney, P. V., Saksena, R. S., Zasada, S. J., McKeown, M., Pickles, S. "The Application Hosting Environment: Lightweight Middleware for Grid-based Computational Science," *Computer Physics Communications*, 2007, 176, 406418.

4.3 Treating cancer with patient-specific chemotherapeutic drug targeting

The identification and treatment of cancer exists on various levels, from the large-scale view of tumour growth down to individual molecular interactions. The treatment of cancer often takes two directions, using targeted radiotherapy to kill malignant cells, while also using tumour-growth inhibitors in an attempt to selectively target and kill tumourous cells. The effectiveness of particular chemotheraputic treatment differs from patient to patient, with some courses of treatment not being effective at all.

A new generation of anticancer drugs are part of an approved scheme called 'targeted therapy,' in which anticancer drugs are directed against cancer-specific molecules and signalling pathways. These are designed to interfere with a specific molecular target, usually a protein that plays a crucial role in tumour cell growth and proliferation. Receptor tyrosine kinases (RTKs) are an example; they are cell surface proteins that can be used as targets to control tumour growth in various preclinical treatment models. Tyrosine kinase inhibitors (TKI) interfere with the related cell signalling pathways and thus allow target-specific therapy for selected malignancies. In fact, some TKIs have been approved for use in cancer therapy, and others are in various stages of clinical trials.

RTKs have been found to be over-expressed or mutated in tumour cells, and these mutations allow cancer cells to develop drug resistance. Clinical studies have shown a strong correlation between a reduction in the response to treatment with TKIs and the presence of these mutations, where the resistance is introduced by preventing or weakening the binding of the receptor to the targeted TKI.

The binding of the tumour-growth inhibitors to cell receptors is identical to small molecule-protein or protein-protein interactions. Molecular dynamics techniques can be used to study these interactions in atomistic detail, and to predict the effect of different receptors and mutations on inhibitor binding affinities. Using patient-specific data, such as the RTK mutation, which is expressed on tumourous cells, MD techniques can be used to rank the binding affinities, and therefore the effectiveness of various treatments against a patient-specific case.

Using a grid-infrastructure, turnaround times can be dramatically accelerated. MD simulations, particularly for the case of various inhibitors and possibly various targets, can be independently run by being farmed off to various grid resources. Providing turnaround times of five days will ensure that the findings are clinically relevant and become part of the clinical decision making process. One of the aims within this project is to develop a work-flow tool, which will use the AHE to permit the automated running of such patient-specific simulations, hiding the unnecessary grid details from clinicians.

5. Discussion

Patient-specific medical simulation holds the promise of revolutionising the diagnosis and treatment of many different medical conditions, by making use of advanced simulation techniques and high performance compute resources. For computational medicine to be of use in modern clinical settings, the timeliness with which results are delivered is of primary concern. Results need to be generated in a timeframe that is useful to the clinician initiating the simulation results; that is, they must be generated in time to inform the treatment regime or procedure under consideration. In the case of neurosurgical treatments, this is in the order of 15 to 20 minutes. In the case of HIV or cancer pathology reports, this is in the order of 24 to 48 hours.

Due to the urgent requirements of patient-specific simulations, the current standard model of high performance compute provision, the batch queue model, is of no use. Simulations have to fit into existing clinical processes; clinical processes cannot be altered to adapt to a batch compute model, as very often a simulation will be used to inform an urgent life or death decision. Because of this, technologies that enable and facilitate urgent computing are of great relevance to the emerging field of patient-specific simulation. Advance reservation tools such as HARC and urgent computing systems such as SPRUCE are essential for making patient-specific medical simulation a reality when using general purpose, high performance compute resources that typically run a wide range of different tasks.

Systems such as HARC and SPRUCE were initially conceived to support the submission of very infrequent on-demand jobs, for example climate models, that typically are only run in emergency situations, such when a hurricane is looming; running simulations at this frequency on general purpose compute resources, such as those available on the TeraGrid, have a negligible effect on the users of the system as a whole. In the case of an urgent simulation using SPRUCE, a limited set of users who had their jobs pre-empted would not notice anything different. Patient-specific medical simulations are of a different nature; a successful patient-specific simulation technique will likely have thousands, or even tens of thousands, of possible patients that it could be performed for. The possible level of compute time required will dwarf the current urgent-computing policies and resources in place.

Patient-specific medical simulation raises several moral, ethical and policy questions that need to be answered before the methodologies can be put to widespread use. Firstly there is the question of the availability of resources to perform such simulations. The compute power currently made available through general purpose scientific grids, such as the TeraGrid or UK NGS, is not enough to satisfy the potential demand of medical simulation. The scarcity of resources raises the question of how such resources will be allocated. Which patients will benefit from medical simulations? Will it be based on the ability to pay? Secondly there is the question of data privacy. Sensitive clinical information is often kept on highly secure hospital networks, and the owners and administrators of such networks are often loath to let any data move from it onto networks over which they have no control, which is necessary if the data is to be shipped to a remote site and used in a simulation. Using such data on 'public' grid resources requires it to be suitably anonymised, so that even if it were to fall into the wrong hands it could not be traced back to the patient it was taken from.

We believe that as such tasks become more widespread and embedded in the clinical process, the market will start to address the first question raised above. Already, many companies are starting to provide utility compute services, such as Amazon's Elastic Compute Cloud,[15] which allows the public to purchase computational cycles. If a market was created for running medical simulations on demand, then we believe it likely that utility compute providers will move to supply the necessary compute services. Although it is uncertain how a pricing model will work in reality, it is likely that the utility compute model will drive down the costs of such simulations, and where the performance of simulations is shown to make a treatment regime more efficient, it is likely that the cost could be met from the money saved. The second question needs to be addressed by medical data managers and government regulators. Once enabling policies have been developed, the process of routinely anonymising data and shipping it from a hospital network or storage facility will become routine. Such a system of anonymisation is being implemented in the neurovascular project discussed, involving discussions with technical network administrators and management from the UK National Health Service (NHS).

[15] Amazon Elastic Compute Cloud – http://www.amazon.com/gp/browse.html?node=201590011

It is essential that a dialogue is joined between governments, researchers, health professionals and business into how the infrastructure needed to perform patient specific medical simulations can be performed on a routine basis. The benefits of performing such simulations are too great to be ignored and, in addition to the case studies presented, we believe that computational simulation will be used in more and more medical scenarios. In the vision that patient-specific medical simulations become a day to day reality in the treatment of patients, vast quantities of simulation data will be available alongside traditional medical data. With parallel advances in data warehousing, data-mining and computational grids, the enhancement of medical practice using simulation will one day become a reality.

Acknowledgments

We acknowledge the work of Kashif Sadiq and Marco Mazzeo, based at the UCL Centre for Computational Science, for their work on the HIV and cerebral bloodflow case studies. We are grateful to EPSRC for funding much of this research through RealityGrid and RealityGrid Platform grants (GR/R67699 and EP/C536452), the GENIUS project (EP/F00561X), and also Integrative Biology (GR/S72023). Our work was partially supported by the National Science Foundation under MRAC award DMR070014N, which provided access to numerous computational resources on the US TeraGrid. This research has also been partially supported by the EU-funded ViroLab project (IST-027446) and has benefitted greatly through involvement of PVC in the EU-funded STEP Coordination Action (no. 027642).

An Interaction Based Composable Architecture for Building Scalable Models of Large Social, Biological, Information and Technical Systems

Karla Atkins
Christopher L. Barrett
Richard Beckman
Keith Bisset
Jiangzhou Chen
Stephen Eubank
Annette Feng
Xizhou Feng
Steven D. Harris
Bryan Lewis
V.S. Anil Kumar
Madhav V. Marathe
Achla Marathe
Henning Mortveit
Paula Stretz
Virginia Polytechnic Institute and State University

Introduction

This article describes our ongoing efforts to develop a global modeling, information and decision support cyberinfrastructure (CI) that will provide scientists and engineers novel ways to study large complex socio-technical systems. It consists of the following components:

i. High-resolution scalable models of complex socio-technical systems;
ii. Service-oriented architecture and delivery mechanism for facilitating the use of these models by domain experts;
iii. Distributed coordinating architecture for information fusion, model execution and data processing; and
iv. Scalable data management architecture and system to support model execution and analytics
v. Scalable methods for visual and data analytics to support analysts.

To guide the initial development of our tools, we are concentrating on agent-based models of inter-dependent societal infrastructures, spanning large urban regions. Examples of such systems include: regional transportation systems; regional electric power markets and grids; the Internet; ad-hoc telecommunication, communication and computing systems; and public health services. Such systems can be viewed as organizations of organizations. Indeed, functioning societal infrastructure systems consist of several interacting public and private organizations working in concert to provide the necessary services to individuals and society. Issues related to privacy of individuals, confidentiality of data, data integrity and security all arise while developing microscopic models for such systems. See [1, 2, 3] for additional discussion (also see Figure 1).

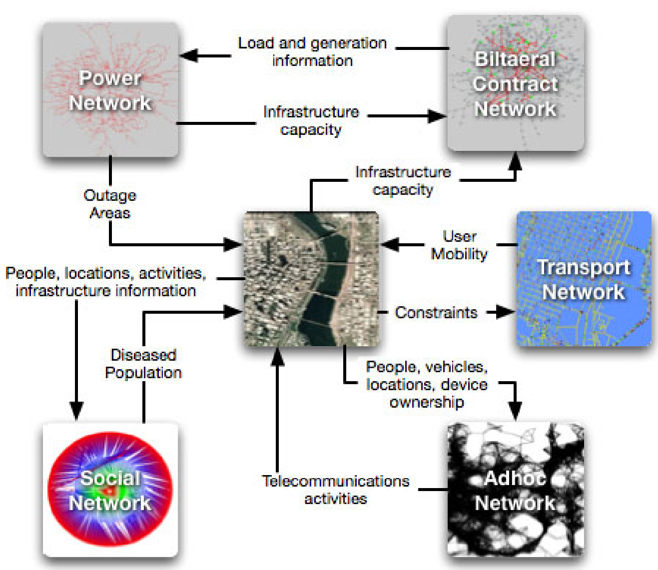


Figure 1. Schematic of societal infrastructure systems (adapted from [2]).

[1] Albert, R., Barabasi, A. "Statistical mechanics of complex networks," *Rev. Mod. Phys.* 74, pp. 47-97, (2002).

[2] Barrett, C., Eubank, S., Anil Kumar, V., Marathe, M. "Understanding Large Scale Social and Infrastructure Networks: A Simulation Based Approach," *SIAM news*, March 2004, Appears as part of Math Awareness Month on The Mathematics of Networks.

[3] "Capturing Complexity Through Agent-Based Modelling," Special Issue of the *Proceedings of the National Academy of Sciences*, Vol. 99 (suppl. 3), 2002.

An Interaction Based Composable Architecture for Building Scalable Models of Large Social, Biological, Information and Technical Systems

The need to represent functioning population centers during complex incidents such as natural disasters and human initiated events poses a very difficult scientific and technical challenge that calls for new state-of-the-art technology. The system must be able to handle complex co-evolving networks with over 300 million agents (individuals), each with individual itineraries and movements, millions of activity locations, thousands of activity types, and hundreds of communities, each with local interdependent critical infrastructures. The system must be able to focus attention on demand and must support the needs of decision makers at various levels. The system must also support related functions such as policy analysis, planning, course-of-action analysis, incident management, and training in a variety of domains (e.g., urban evacuation management, epidemiological event management, bio-monitoring, population risk exposure estimation, logistical planning and management of isolated populations, site evacuations, interdependent infrastructure failures).

Constructing large socio-technical simulations is challenging and novel, since, unlike physical systems, socio-technical systems are affected not only by physical laws but also by human behavior, regulatory agencies, courts, government agencies and private enterprises. The urban transportation system is a canonical example of such interaction; traffic rules in distant parts of the city can have an important bearing on the traffic congestion in downtown, and seemingly "reasonable" strategies such as adding a new road somewhere might worsen the traffic. The complicated inter-dependencies within and among various socio-technical systems, and the need to develop new tools, are highlighted by the failure of the electric grid in the northeastern U.S in 2003. The massive power outage left people in the dark along a 3,700 mile stretch through portions of Michigan, Ohio, Pennsylvania, New Jersey, New York, Connecticut, Vermont and Canada. Failure of the grid led to cascading effects that slowed down Internet traffic, closed down financial institutions and disrupted the transportation; the New York subway system came to a halt, stranding more than 400,000 passengers in tunnels.[4]

The CI we are building was motivated by the considerations to understand the complex inter-dependencies between infrastructures and the society as described above. Over the past 15 years and in conjunction with our collaborators, we have established a program for modeling, simulation and associated decision support tools for understanding large socio-technical systems. The extremely detailed, multi-scale computer simulations allow users to interact among themselves as well as interact with the environment and the networked infrastructure. The simulations are based on our theoretical program in discrete dynamical systems, complex networks, AI and design and analysis of algorithms (see [2,5,6,7,8] and the references therein).

Until 2003, much of our efforts were concentrated on building computational models of individual infrastructures, see [2]. Over the last 7-10 years, significant advances have been made in developing computational techniques and tools that have the potential of transforming how these models are delivered to and used by the end users.[9] [10] [11] This includes, web services, grid computing and methods to process large amounts of data. With the goal of harnessing this technology, since 2005, we have expanded the scope of our effort. In addition to building scalable models, we have also begun the development of an integrated CI for studying such inter-dependent, socio-technical systems. It consists of mechanisms to deliver the access to these models to end users over the web, development of a data management environment to support the analysis and data, and a visual analytics environment to support decision-making and consequence analysis (see [2,6]). The CI will provide social scientists unprecedented Internet-based access to data and models pertaining to large social organizations. In addition, the associated modeling tools will generate new kinds of synthetic data sets that cannot be created in any other way (e.g., direct measurement). The data generated by these methods will protect the privacy of individuals as well as the confidentiality of

4 US-Canada Power System Outage Task Force. Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations April 2004. http://www.ksg.harvard.edu/hepg/Blackout.htm

5 Barrett, C. L., Bisset, K., Eubank, S., Kumar, V. S. A., Marathe, M. V., Mortveit, H. S. "Modeling and Simulation of Large Biological, Information and Soci-Technical Systems: An Interaction-Based Approach," *Proc. Short Course on Modeling and Simulation of Biological Networks, AMS Lecture Notes, Series: PSAPM*, 2007.

6 Barrett, C., Eubuank, S., Marathe, M. "Modeling and Simulation of Large Biological, Information and Socio-Technical Systems: An Interaction Based Approach," in *Interactive Computation: The New Paradigm*, D. Goldin, S. Smolka and P. Wegner Eds. Springer Verlag, (2005).

7 Eubank, S., Guclu, H., Anil Kumar, V. S., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N. "Modeling Disease Outbreaks in Realistic Urban Social Networks," *Nature*, 429, pp. 180-184 May (2004).

8 Barrett, C., Mortveit, H., Reidys, C. "Elements of a Theory of Computer Simulation: III," *Applied Mathematics and Computation*, pp. 325–340, (2001).

9 Fox, G. "Grids Challenged by a Web 2.0 and Multicore Sandwich," *CCGrid 2007* Keynote Talk Rio de Janeiro Brazil May 15 2007.

10 Hayden, L., Fox, G., Gogineni, P. "Cyberinfrastructure for Remote Sensing of Ice Sheets," *Proceedings of TeraGrid 2007 Conference,* Madison Wisconsin June 4-8 2007.

11 *Grid Computing: Making the Global Infrastructure a Reality* edited by Fran Berman, Geoffrey Fox and Tony Hey. Wiley Publishers, March 2003.

data obtained from proprietary datasets. This will enable social scientists to investigate entirely new research questions about functioning societal infrastructures and the individuals interacting with these systems. Everything, from the scope and precision of socio-technical analysis to the concept of collaboration and information integration, will change, as a dispersed framework that supports detailed interdependent interaction of very large numbers of complex individual entities that come into use and evolve. The tools will also allow policy makers, planners, and emergency responders unprecedented opportunities for coordination of and integration with the information for situation assessment and consequence management. This is important for planning and responding in the event of a large-scale disruption of the societal infrastructures.

Challenges

Current-generation High Performance Computing (HPC) based modeling environments for socio-technical systems are complex, large-scale special-purpose systems. These systems are readily accessible only to a limited number of highly specialized technical personnel in narrowly defined software applications. In other words, currently, web oriented distributed information CI for representing and analyzing large socio-technical systems simply do not exist. The vision of the CI under development is to change the model of delivery of HPC-oriented models and analytical tools to analysts. Just as the advent of search engines (e.g., Google) radically altered research and analysis of technical subjects across the board, the goal of the CI is to make HPC resources seamless, invisible, and indispensable in routine analytical efforts by demonstrating that HPC resources should be organized as an evolving commodity, and made accessible in a fashion as ubiquitous as Google's home page. Recently, grid based global cyber-infrastructures for modeling physical systems have been deployed. This is only a first step in developing what we would call *semantic complex system modeling*; we hope to make similar progress in the context of socio-technical systems. Developing such an infrastructure poses unique challenges that are often different than the ones faced by researchers developing the CI for physical systems. We outline some of these below.

1. **Scalability**: *The CI must be globally scalable.* The scalability comes in three forms: (i) allowing multiple concurrent users, (ii) processing huge quantities of distributed data and (iii) ability to execute large national-scale models. For example, simulating dynamical processes usually occurs on unstructured, time varying networks. Unlike many physical science applications, the unstructured network is crucial in computing realistic estimates, e.g., disease dynamics in an urban region. The unstructured network represents the underlying coupled, complex social contact and infrastructure. We need to simulate large portions of the continental United States – this implies a time varying dynamic social network of over 250 million nodes.

2. **Coordination**: *The CI should allow computational steering of experiments.* The systems needed by stakeholders are geographically distributed, controlled by multiple independent, sometimes competing, organizations and are occasionally dynamically assembled for a short period of time. Currently, web-enabled simulations and grid computing infrastructures for physical simulations have concentrated on massively parallel applications and loose forms of code coupling wherein large-scale experiments are submitted as batch jobs. Computational steering of simulations based on analysis is usually not feasible due to the latencies involved. Several socio-technical systems of interest can be formally modeled as partially observed Markov decision processes (POMDP) and large n-way games; a key component of POMDP and such games is that actions taken by an observer change the system dynamics (e.g., isolating critical workers during an epidemic). In other words, the underlying complex network, individual behavior and dynamics of particular processes over the network (e.g., epidemic)

co-evolve. Such mechanisms require computational steering, which in turn requires CI for coordinating resource discovery of computing and data assets; AI-based techniques for translating user level request to efficient workflows; re-using data sets whenever possible and spawning computer models with required initial parameters; and coordination of resources among various users. *Computational steering* also occurs at a coarser level as well — in this case due to extremely large design space, we need to consider adaptive experimental designs. This is by and large infeasible on today's grid computing environments

3. **Data & Information Processing**: *The CI should facilitate efficient data and information fusion and analysis.* The Internet has enabled the sharing of data in a simple and cost effective way, from the producers' side. Consumers of the data must still locate the appropriate data and deal with multiple incompatible data formats. The heterogeneity, volume and geographic distribution of data implies that social scientists, without the proper tools and use of database techniques, will be left to write custom programs that will tend to be less efficient than well crafted database and middleware methods. Unlike simulations of physical systems, models of socio-technical systems are usually data-intensive. Moreover, the data sets are being continually collected, refined, integrated and aligned to support ongoing analysis. Analogous to physical simulations, the output data is large and processing it is a computational challenge. More importantly, a POMDP model of socio-technical systems implies that a lot of data mining and analysis has to be done in concert with the simulation. This implies stringent computational requirements.

4. **User Support**: Development of appropriate analysis frameworks for users are needed, including user interfaces, high level formalisms to set up experiments, and visual and data analytics, which include methods for integrating heterogeneous databases to support multi-view visualization (e.g., disease spread in a geographic region and epidemic curves); methods for visualizing and analyzing large co-evolving coupled networks; and data mining and knowledge discovery tools to support analytical processes.

In addition, we need to develop environments and tools for simulation assisted decision support and consequence analysis. This includes methods for presenting results of analysis and simulations so as to avoid confirmation bias and framing effects, simulation based micro-economic analysis of decisions, and methods in risk analysis for ranking assets and understanding the inherent uncertainties in modeling such systems.

Overall Architecture & Current Status

Figure 2 shows a conceptual architecture of the overall system that we are developing. Simfrastructure assumes the role of coordination between all the constituent components. This includes high resolution models for simulating large socio-technical systems, SimDM: a distributed data management environment, the underlying data and compute grids that provide low level data and compute services. Simfrastructure uses (tuple/java)-spaces to achieve the desired coordination goals. Currently, we have operating models for public health, commodity markets, transportation, integrated telecommunication networks, urban populations and built infrastructure. These models can all run on high performance computing platforms. We are currently extending them to work on grid-like architectures developed as a part of the NSF funded Teragrid initiative.

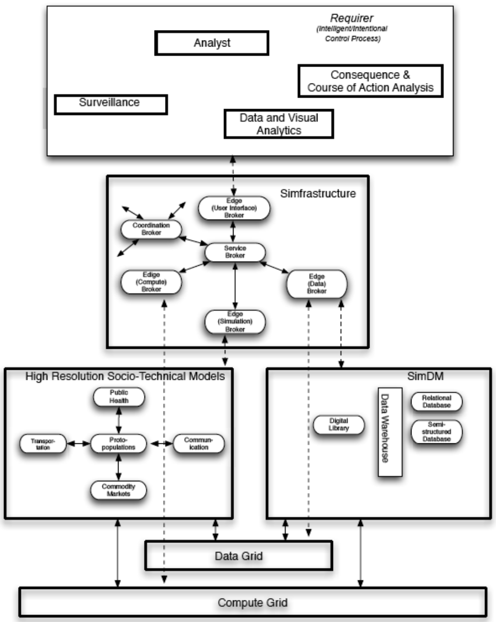


Figure 2. Overall cyber-infrastructure architecture for simulation based decision support and consequence analysis of large societal infrastructures.

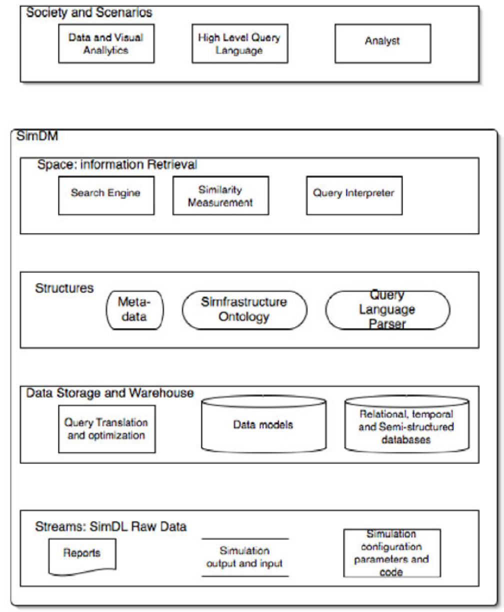


Figure 3. Schematic architecture of SimDM: data management module based on the 5S specification.

Current Infrastructure Models: Currently, modules for public health, telecommunications, commodity markets and urban social networks have been developed and integrated within the Simfrastucture framework [5, 6, 12, 13, 14]. Simdemics is a module that supports public health epidemiology. Simdemics itself has three different implementations of methods that simulate the spread of infectious diseases. Sigma is a highly scalable, web-based, service-oriented modeling framework for analyzing large generic markets for commodities such as electricity, oil, corn, as well as for allocating distributed computer resources in a utility data center. It supports large-scale synthetic and human economic experiments and studies related to bargaining, learning, cooperation, and social and risk preferences. It can also be used as a tool to study appropriate designs for marketing bandwidth in an unlicensed radio spectrum. The integrated tele-communication modeling environment consists of analytical and simulation-based modeling tools for design and analysis of next-generation computing and communication systems that are based on packet switched network technology. Examples of such systems include mesh networks deployed in urban and rural communities, vehicular ad hoc networks, hybrid, cellular, mesh and sensor networks. Finally, the module for generating urban social contact networks generates high fidelity synthetic networks consisting of people, locations and their interactions. The kinds of interactions determine the specific social network that is created. In addition to these, TRANSIMS an urban transport module developed by our group and team members at Los Alamos National Laboratory, is also integrated within the framework. A key feature of the overall architecture is its ability to easily integrate other infrastructure models within the framework. For instance, the Urban infrastructure suite (UIS) developed at Los Alamos can be integrated within the current framework.

SimDM is an integrated data management environment. It follows the 5S (Streams, Structures, Spaces, Scenarios, and Societies) framework that defines the meta-model for a minimal digital library. A conceptual architecture is shown in Figure 3 above. See [15, 16] for additional discussion.

[12] Atkins, K., Barrett, C., Beckman, R., Bisset, K., Chen, J., Eubank, S., Anil Kumar, V. S., Lewis, B., Macauley, M., Marathe, A., Marathe, M., Mortveit, H., Stretz, P. "Simulated Pandemic Influenza Outbreaks in Chicago," *NIH DHHS Study Final report*, 2006.

[13] Barrett, C., Smith, J. P., Eubank, S. "Modern Epidemiology Modeling," *Scientific American*, March (2005).

[14] Barrett, C., Beckman, R., Berkbigler, K., Bisset, K., Bush, B., Campbell, K., Eubank, S., Henson, K., Hurford, J., Kubicek, D., Marathe, M., Romero, P., Smith, J., Smith, L., Speckman, P., Stretz, P., Thayer, G., Eeckhout, E., Williams, M. D. "TRANSIMS: Transportation Analysis Simulation System," *Technical Report LA-UR-00-1725*, Los Alamos National Laboratory Unclassified Report, 2001. An earlier version appears as a 7 part technical report series LA-UR- 99-1658 and LA-UR-99-2574 to LA-UR-99-2580.

[15] Bailey-Kellog, C., Ramakrishnan, N., Marathe, M. "Spatial data mining to support pandemic preparedness," *SIGKDD Explorations* 8: 80-82, 2006.

[16] Barrett, C., Bisset, K., Eubank, S., Fox, E., Ma, Y., Marathe, M., Zhang, X. "A Scalable Data Management Tool to Support Epidemiological Modeling of Large Urban Regions," *European Conference on Digital Libraries (ECDL)*, pp. 546-548, 2007.

It stores streams of textual bits from files or databases and audio/video sequences. Challenges arise from enforcing proper structures over heterogeneously structured digital objects with close conceptual relationships. In our prototype implementation, we used RDF-based metadata, which defines semantic contents of objects and relationships among them. The metadata constructs a knowledgebase for Simfrastructure, on which a browsing service could be based. Simfrastructure objects contain both textual information and real number parameters.

Simfrastructure serves the role of coordinating these diverse simulations, the data management tool, the visual and data analytic tools, and the end users. It is organized around the Software as a services (SAS) paradigm. Currently, it uses Javaspaces as the implementing construct, but the basic concepts are generic and readily implemented using other similar languages. We will use the terminology in Gelernter and Carriero[17][18] for describing Simfrastructure. The *asynchronous ensembles* in our architecture consist of simulation models, databases, GUIs and analytical tools. The basic concept within Simfrastructure is that of *brokers: coordinating processes responsible for achieving a desired workflow by appropriately invoking appropriate asynchronous ensembles*. Brokers use *associative memory* for communicating data objects between them. In Javaspaces this is called a blackboard. For computational efficiency and security, these blackboards are generally distributed and organized hierarchically. Brokers are also organized hierarchically; this hierarchy captures calling rules. As in tuple-spaces or Javaspaces, brokers place appropriate data objects in the associative memory. Our architecture uses the generative communication paradigm; brokers act as coordinators for this purpose. Brokers are responsible for understanding what information needs to be communicated between various asynchronous ensembles. They are lightweight processes that are assigned the task of requesting information from various ensembles, communicate information/data between ensembles by using blackboard and in the end achieve a given workflow. Important parameters are that of computational efficiency, memory requirement and accuracy. In our envisioned architecture, achieving a given functionality has to account for these parameters; brokers call appropriate computation and evaluation processes to conclude if the data object returned conforms to the required specification. We have chosen to use a tuple-space model in contrast to a message passing model for our coordinating system for the following reasons:

- Brokers in general will not know how a specific request can be satisfied, therefore, it uses common associative memory as a way to broadcast its request. Brokers that can invoke appropriate processes to fulfill these requests using **in** or **rd** like primitives available in Linda tuple spaces. This was one of the central features of tuple-space like constructs and is very useful in our setting.
- The broker requests are highly asynchronous; in general, requests are generated on demand when a specific analysis needs to be done by an analyst. At that time, we have very little control over the specific computing and data resources at our disposal.
- Broker based architecture allows us to develop solutions that protect participating institutions' IP and security requirements. Since all communication happens among brokers and not directly between services, organizations need not have knowledge either of the entire problem or all of the resources being used to solve the problem. By using a trusted third-party to host the computation, one organization may provide a proprietary model that uses proprietary data from a second party, without either organization needing a trust relationship with the other.

[17] Gelernter, D., Carriero, N. "Coordination Languages and their significance," *CACM*, 1992, 35(2).

[18] Gelernter, D., Carriero, N. "Linda in Context," *CACM*, 1989, 32(4).

An Interaction Based Composable Architecture for Building Scalable Models of Large Social, Biological, Information and Technical Systems

In the current implementation of Simfrastructure, we have three kinds of brokers:

- *Edge brokers*: mediate access to a particular resource (simulation, data, service, etc), removing the need for the resource to communicate directly with any other resource
- *Service brokers*: coordinate the edge brokers in response to satisfy a given query
- *Coordination brokers*: coordinate the overall workflow

There are several kinds of edge brokers: simulation edge brokers meant for interacting with simulations; data management edge brokers meant to access the data management module; the visual and data analytics brokers; surveillance data access brokers, etc. These edge brokers can in turn call data and compute grid primitives as necessary. A formal grammar specifies the call structure and the various access rules. Simfrastructure in concert with the simulation models and SimDM are designed to support computational steering; this is crucial for the class of applications we are interested in studying.

Illustrative Scenarios

It is valuable to go through an illustrative scenario of how the CI might be used in practice by analysts and policy makers for planning and response to natural or human-initiated crises. The CI been used to support several user defined studies over the past 10 years; see [3, 7, 19]. The scenario illustrates the need to address each of the challenges discussed above. It also highlights the need to solve the complete problem rather than solving it piecemeal. Solutions need to be practical and usable.

[19] Atkins, K., Barrett, C., Beckman, R., Bisset, K., Chen, J., Eubank, S., Anil Kumar, V. S., Lewis, B., Marathe, A., Marathe, M., Mortveit, H., Stretz, P. "An analysis of layered public health interventions at Ft. Lewis and Ft. Hood during a pandemic influenza event," *TR-NDSSL-07-019*, 2007.

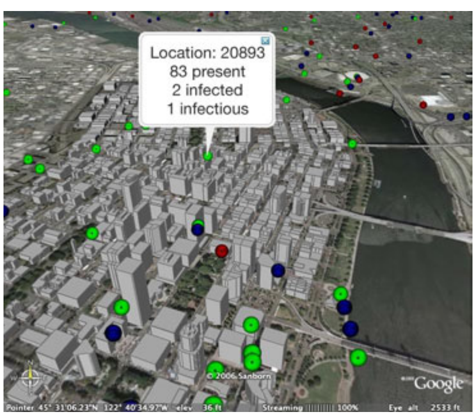


Figure 4. A sample graphic available to the analyst when using the cyber-infrastructure in the illustrative scenarios. Note the spatial detail that can be generated.

Situational Awareness and Consequence Analysis in the Event of Epidemics

In this scenario, during a heat wave in a city, an adversary shuts down portions of the public transit system and a hospital emergency room during the morning rush hour. They spread a harmless but noticeable aerosol at two commuter rail stations. These events, occurring nearly simultaneously, foster a chaotic, if not panic-stricken, mood in the general public. Disinformation released via the mass media enhances the perception of an attack. Simulations of epidemics on social contact networks combined with simulations of urban population mobility and other infrastructure simulations can be used for situation assessment and course of action analysis. Simfrastructure first calls the data broker to see if the network is already available. Assuming the answer is no, Simfrastructure then calls the population mobility broker for constructing the dynamic social network. This in turn will involve fusing information about census data, data on individual activity, and location data from commercial and public databases.

These are the people who would show up first for treatment if indeed a chemical or biological attack had occurred. They also would serve as the subpopulation to seed our epidemiological simulations. Simfrastructure calls data mining tools and Simdemics to achieve these tasks.

Biases in their demographics compared to a random sample of the population will induce persistent biases in the set of people infected at any time. We estimated the demand at hospitals, assuming that people would arrive at a hospital near their home or current location. We also estimated the demographics of casualties under an alternative scenario during only a heat wave. Historically, the most likely casualties of a heat wave are elderly people living alone with few activities outside the home.

This information, combined with demographic and household structure data, allowed us to estimate demand for health services created by the heat wave by demographic and location. For situation assessment, we noted the obvious differences between these two demand patterns. In an actual event, comparison with admissions surveillance data would allow quick disambiguation between the two situations. We estimated the likely spread of disease for several different pathogens by demographic and location. Furthermore, we implemented several suggested mitigating responses, such as closing schools and/or workplaces, or quarantining households with symptomatic people. Knowledge of the household structure permits an exceptionally realistic representation of the consequences of these actions. For example, if schools are closed, a care-giver will also need to stay home in many households.

Conclusions and Summary

We described our work in progress that aims to build a scalable CI to study large socio-technical networked systems. The goal of the CI is to provide seamless access to HPC-based modeling and analysis capability for routine analytical efforts. It consists of (i) high-resolution models, tools for decision making, and consequence analysis, (ii) service-oriented architecture and delivery mechanism for facilitating the use of these models by domain experts, (iii) distributed coordinating architecture for information fusion, model execution and data processing, and (iv) scalable methods for visual and data analytics to support analysts.

Due to space considerations, we have not discussed peta-scale computing and data grids that will serve as the underlying technology. Much remains to be done to develop the CI. Researchers across the world are developing new tools in web services, tools and CI for various problem domains.[9] [10] [11] We hope to build on these advances.

Acknowledgements

The work is partially supported by an NSF HSD grant, the NIH MIDAS project, the CDC center of excellence, DoD and a Virginia Tech internal grant. We thank our current and past collaborators, on related topics, especially, Douglas Roberts, Aravind Srinivasan, Geoffery Fox, Arun Phadke and Jim Thorp, students in the Network Dynamics and Simulation Science Laboratory and members of the TRANSIMS and NISAC projects at Los Alamos National Laboratory. Finally, we thank Suman Nadella and Pete Beckman for inviting us to submit an article for this *CTWatch Quarterly* issue.

Supercomputing On Demand: SDSC Supports Event-Driven Science

Paul Tooby
Dong Ju Choi
Nancy Wilkins-Diehr
San Diego Supercomputer Center

Somewhere in Southern California a large earthquake strikes without warning, and the news media and the public clamor for information about the temblor -- Where was the epicenter? How large was the quake? What areas did it impact?

A picture is worth a thousand words – or numbers – and the San Diego Supercomputer Center (SDSC)[1] at UC San Diego is helping to provide the answers. Caltech computational seismologist Jeroen Tromp can now give the public movies that tell the story in a language that's easy to understand, revealing waves of ground motion spreading out from the earthquake -- and he can deliver these movies in just 30 minutes with the help of a supercomputer at SDSC. But he can't do it by submitting a job to a traditional computing batch queue and waiting hours or days for the results.

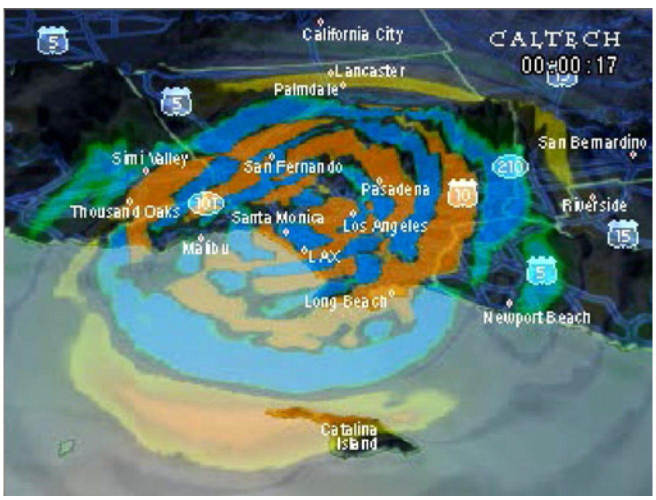


Figure 1. Frame from a movie of a "virtual earthquake" simulation of the type that will be run on SDSC's new OnDemand system to support event-driven science. The movie shows the up-and-down velocity of the Earth's surface as waves radiate out from a magnitude 4.3 earthquake centered near Beverly Hills, California. Strong blue waves indicate the surface is moving rapidly downward, while red/orange waves indicate rapid upward motion. Courtesy of Joroen Tromp, ShakeMovie, Caltech.

Tromp is an example of the new users in today's uncertain world who require immediate access to supercomputing resources.[2] To meet this need, SDSC has introduced OnDemand, a new supercomputing resource that will support event-driven science.[3]

"This is the first time that an allocated National Science Foundation (NSF) TeraGrid supercomputing resource will support on-demand users for urgent science applications," said Anke Kamrath, director of User Services at SDSC. "In opening this new computing paradigm we've had to develop novel ways of handling this type of allocation as well as scheduling and job handling procedures."

In addition to supporting important research now, this system will serve as a model to develop on-demand capabilities on additional TeraGrid systems in the future. TeraGrid is an NSF-funded large-scale production grid linking some of the nation's largest supercomputer centers for open scientific research including SDSC.

[1] San Diego Supercomputer Center (SDSC) - http://www.sdsc.edu/

[2] SDSC Allocations - http://www.sdsc.edu/us/allocations/

[3] SDSC On-Demand cluster - http://www.sdsc.edu/us/resources/ondemand/

Urgent applications that will make use of OnDemand range from making movies of Southern California earthquakes to systems that will help give near real-time warnings based on predicting the path of a tornado or hurricane, or foretell the most likely direction of a toxic plume released by an industrial accident or terrorist incident.

When an earthquake greater than magnitude 3.5 strikes Southern California, typically once or twice a month, Tromp's simulation code needs to use 144 processors of the OnDemand system for about 28 minutes. Shortly after the earthquake strikes a job is automatically submitted and immediately allowed to run. The code launches and any "normal" jobs running at the time are interrupted to make way for the on-demand job.

SDSC computational expert Dong Ju Choi worked extensively with Tromp to ensure that the simulation code will run efficiently in on-demand mode on the new system.

"SDSC's new OnDemand system is an important step forward for our event-driven earthquake science," said Tromp. "We're getting very good performance that will let us cut the time to deliver earthquake movies from about 45 to 30 minutes or less, and every minute is important."

The movies that result from the computations are made available as part of the ShakeMovie project in Caltech's Near Real-Time Simulation of Southern California Seismic Events Portal.[4] But behind the scenes of these dramatic earthquake movies, a great deal of coordinated activity is rapidly taking place in a complex, automated workflow.

[4] ShakeMovie, Caltech's Near Real-Time Simulation of So. Calif. Seismic Events Portal - http://shakemovie.caltech.edu/

The system springs to life every time an earthquake occurs in Southern California. When an event takes place, thousands of seismograms, or ground motion measurements, are recorded at hundreds of stations across the region, and the earthquake's epicenter, or location, as well as its depth and intensity are determined.

The waiting ShakeMovie system at Caltech collects these seismic recordings automatically over the Internet. Then, for events greater than magnitude 3.5, to fill in the gaps between the actual ground motion recorded at specific locations in the region, the scientists use the recorded data to guide a computer model that creates a "virtual earthquake," giving an overall view of the ground motion throughout the region.

The animations rely on the SPECFEM3D_BASIN software, which simulates seismic wave propagation in sedimentary basins. The software computes the motion of the earth in 3-D based on the actual earthquake recordings and what is known about the subsurface structure of the region, which greatly affects the wave motion -- bending, speeding or slowing, and reflecting energy in complex ways.

After the full 3-D wave simulation is run on the OnDemand system at SDSC and a system at Caltech for redundancy, data that captures the surface motion (displacement, velocity, and acceleration) are collected and mapped onto the topography of Southern California, and rendered into movies. The movies are then automatically published via the portal, and an email is sent to subscribers, including the news media and the public.

Figure 2. OnDemand cluster at SDSC

OnDemand is a Dell cluster with 64 Intel dual-socket, dual-core compute nodes for a total of 256 processors. The 2.33 GHz, 4-way nodes have 8 GB of memory. The system, which has a nominal theoretical peak performance of 2.4 Tflops, is running the SDSC-developed Rocks open-source Linux cluster operation software and the IBRIX parallel file system. Jobs are scheduled by the Sun Grid Engine.

OnDemand also makes use of the SPRUCE system developed by a team at Argonne National Laboratory. SPRUCE provides production-level functionality, including access controls, reporting, and fine-grained control for urgent computing jobs. An organization can issue tokens to its user groups who have been approved for urgent computing runs. Different colors (classes) of SPRUCE tokens represent varying urgency levels. A yellow token will put the requested job in the normal queue in the Sun Grid Engine scheduler; an orange token goes to the high priority queue; and a job submitted with a red token will preempt running jobs if necessary.

The researchers are working to develop additional capabilities. Currently, jobs with the least amount of accumulated CPU are the first to be preempted. In the future, preempted backfill jobs may be held and restarted when appropriate, without being killed, and investigation of checkpoint and restart systems is ongoing.

Backfill jobs consist of a variety of regular user jobs, primarily parallel scientific computing and visualization applications using MPI. Users who run on the OnDemand cluster are made aware of the cluster's mission to prioritize jobs that require immediate turnaround.

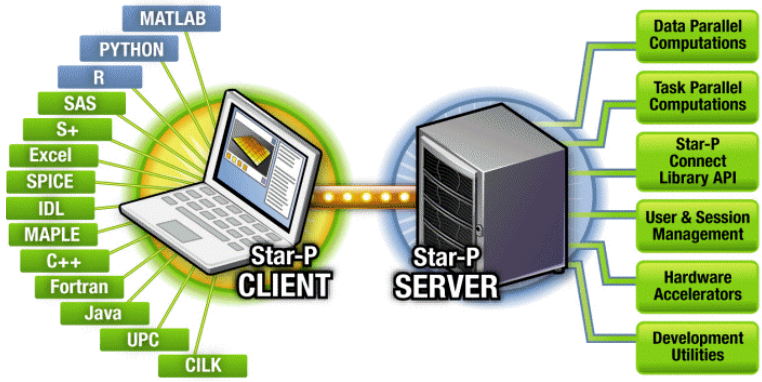


Figure 3. The Star-P extends easy access to supercomputing to a much wider range of researchers.

One of the most interesting and successful applications using OnDemand is a commercial application called Star-P,[5] which extends easy access to supercomputing to a much wider range of researchers. Users can code models and algorithms on their desktop computers using familiar applications like MATLAB, Python and R, and then run them interactively on SDSC's OnDemand cluster through the Star-P platform. This eliminates the need to re-program applications to run on parallel systems, so that programming that took months can now be done in days, and simulations that took days on the desktop can now be done in minutes. Lowering the barrier to supercomputing resources will let researchers jumpstart research that otherwise wouldn't get done.

[5] Star-P at Interactive Supercomputing - http://www.interactivesupercomputing.com/

Star-P supports researchers by allowing them to transparently use HPC clusters through a client (running on their user desktop environment) and server framework (running in an HPC cluster environment). For example, existing MATLAB users on a PC desktop can now achieve parallel scalability from the same MATLAB desktop interface with a simple set of STAR-P commands. This has enabled many users to achieve the tremendous speed-ups that advanced research groups see by laboriously reprogramming applications using MPI.

Researchers on SDSC's OnDemand are using STAR-P in a variety of application areas, including science, engineering, medical and financial disciplines. Several research groups have seen true performance breakthroughs through STAR-P, which fundamentally changes the type of problems they are able to explore. A close collaboration with SDSC also won the Interactive Supercomputing HPC Challenge at SC 07.

SDSC and its academic and industrial partners, including Argonne National Laboratory and Interactive Supercomputing, are aggressively continuing to improve the cluster environment to enhance this urgent computing service. The accumulating experience at SDSC using OnDemand is playing a critical role as a testbed as the team works to further develop the urgent computing paradigm and robust infrastructure.

CTWatch QUARTERLY

ISSN 1555-9874 VOLUME 4 NUMBER 1 MARCH 2008

URGENT COMPUTING: EXPLORING SUPERCOMPUTING'S NEW ROLE

GUEST EDITOR **PETE BECKMAN**

AVAILABLE ON-LINE:
http://www.ctwatch.org/quarterly/

E-MAIL CTWatch QUARTERLY:
quarterly@ctwatch.org

CTWatch is a collaborative effort

http://icl.cs.utk.edu/ http://www.ncsa.uiuc.edu/ http://www.sdsc.edu/

CTWatch Quarterly is a publication of

http://www.ci-partnership.org/

Sponsored by

http://www.nsf.gov/