ICL Proposal Development Today?

The Present Interprets the Past for the Future

INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Proposal development is competitive team writing…

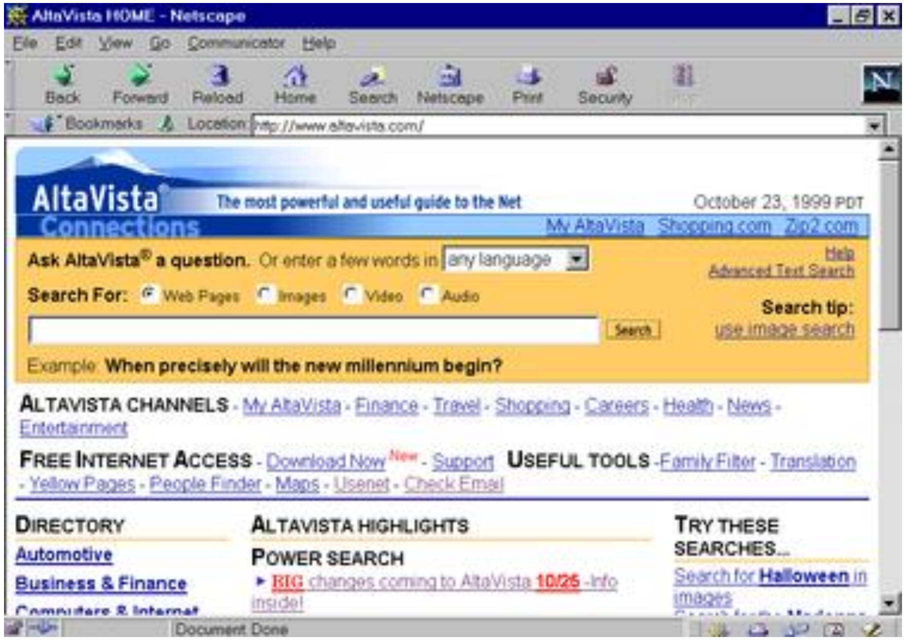
… which is very much like competitive team racing.

proposals as race cars and … agency solicitations as race tracks.

- Proposal narratives (the "story") are the vehicles that carry the ideas and vision (Often you can run them again!)
- Agency solicitations are the race tracks on which they run
- **The PI's are the drivers, "the creative talent"**
- Team's job: Help the driver deliver their best performance
- Good teamwork: communication, cooperation, knowing your teammates
- The analogy is not perfect

When I started at ICL…

- There was no Google
- There were no “Clouds” or “cloud services”
- Your online social network: email
- The preferred writing tool: MS Word
- There was no online proposal submission process
- “Computational Science” was a controversial thing
- But we were riding the Internet Tsunami!!



Some key competitive challenges

- Communication environment
  - E.g. Zoom, Slack, …
- Technological Change on multiple fronts
  - Infrastructure being addressed is changing
  - Tools for building the proposals are changing
  - AI is changing everything!
- Institutional Change
  - The race track itself changes every year (New administration, new requirements, …)
  - Scientific communities are changing: More interdisciplinary collaboration, new methods and tools, etc.
  - AI is changing everything!
- So what were the key lesson's we learned about proposal development basics that we need to adapt to the new world that's coming?

Lesson 1: Develop “the story”

- Tell it in condensed form in the Intro
- Map the RFP requirements to the structure of the document so that reviewers can see where the relevant points are addressed.
- “Should” in the RFP usually means “must”; first, avoid being ruled out. Sometimes they want more than you can give in x pages.
- All good writing is by compression.
- Work on good fit between the story and agency
- You’re trying to create a champion in the room

Lesson 2: Coordinate the budget and the narrative

- The narrative and the budget should tell one coherent story
- What will the agency fund?
- What is required to do the work?
- Consult with team on various “plans”
- Write a strong budget justification

Lesson 3: Project Illustration pg. 1

DARE

**Project Description: Empi
able Hybrid Systems**

1 Introduction

The rapid and ongoing revolution in p
of tuning the performance of a paralle
shows conclusively that the wrong c
to severe performance penalties, the
application executes on scalable hybri
of the program and a description of th
example, the optimum task-granulari
for smaller problems running on few
Additionally, accurately tuning and m
sizes, which quickly becomes prohibit
(DARE) project, we aim to discover a
have helped to pioneer in the field of

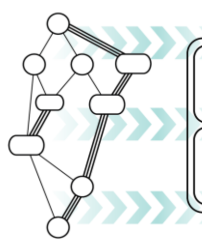


Figure 1: Mutual refinement of topol
DAG-structured task-based applicatio
data need to be transferred between ta

Since the goal of DARE is to i
execution schedules of task-based p
prototype a software workbench—the
we will test the hypothesis that, *by usi
refines the* ***hardware topology model****,
execution schedules that enable us t
structured) task-based programs, ma*
tells us that topology analysis, kerne
cannot be connected in a traditional
propose an incremental / iterative proc

PAPIex

1 Introduction

For more than a decade, the world of high
monitoring library to help implement the
it, you cannot improve it." Widely deploy
the hardware performance counters avail
software infrastructure in every applicati
improving performance can be mission
for experiments in Computer Science. I
systems that combine multicore CPUs a
other interconnects—as well as the eme
duced data movement as a primary progr
PAPI. Many new performance indicators
tending PAPI to monitor performance-cri
processors, including on-chip communic
agement logic, will enable tuning for mo
more importantly, contention for these "
application performance.

At the same time, in order to enable cr
and integrated modeling and analysis o
tire hardware-software system, PAPI
be extended to expose performance me
key software components found in the H
ware stack, such as numerical libraries,
tion frameworks, and parallel runtime
Although they are now treated as blac
having a standard window for viewin
mance events defined by these compon
be critical to achieving the performance
ficiency goals of next generation applic
the scale and complexity of platforms co
grow.

As the name suggests, *the goal of the "P
project is to enhance and extend PAPI
sure the wide range of new hardware
ware events on the extreme scale platfo
will form the basis of exascale computi*
proposed improvements to PAPI-EX wil
to continue its essential role in enabling I
formance analysis and optimization. T
especially true for the large and diverse c
of PAPI-based performance tools on whi
users now depend, including TAU, HPC
As the principal designers of PAPI, and l
well positioned not only to solve the inte
tegrate critical PAPI enhancements that s
The work required to accomplish these o

SparseKaf

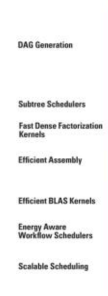
1 Introduction: Motivation and Overview

The use of sparse direct methods in computational sci
LU factorization) can be used to find solutions in n
linear systems, sparse linear least squares, and eigenv
a broad spectrum of large scale applications (see Sec.
and the properties of the hardware platforms on w
Because they hold out the promise of delivering mu
Hybrid Multicore Processors (HMPs)—multicore pr
or Xeon Phi cores—are poised to dominate the landsc
to extreme scale systems that consist of *Multiple Hy*
high-speed network. In the *SparseKaffe* project desc
pyramid, we aim to surmount the research challenges
the power and energy efficiency that these platforms

Those challenges begin with the fact that efficien
neous algorithms—a term that we use to describe alg
the specific strengths of the hardware components av
tions on the critical path of an algorithm may be more
while highly parallel computations, may be more su
matrices demonstrates that a combination of new alg
bring large performance improvements over legacy
equations [43], least square problems [44], eigenval
same time we were already able to achieve a factor
prototype sparse QR factorization [111].

Building on this approach, the name of our proj
*Scheduling and Execution of Kernels with Auto-tuna
computing on heterogeneous architectures*—encapsu
to create fast and efficient sparse direct methods for t

Our goal is to develop
high-performance parallel
sparse direct methods (multi-
frontal QR, Cholesky, and LU
factorization) with irregular
and hierarchical structure
that can exploit MHMPs to
achieve orders of magnitude
gains in computational per-
formance, while also paying
careful attention to the energy
requirements. Such algo-
rithms will enable us to create
a robust suite of open-source,
high-performance software
that others can use to solve
their applications. To achieve
this goal, we will focus efforts
on the following objectives:



Figur

BONSAI

BONSAI: An Open Software Infrastructure for Parallel Autotuning of Computational Kernels

1 Introduction

While hybrid systems that incorporate accelerators, such as GPUs and coprocessors like Xeon Phi, can deliver dramatically better floating-point performance, memory bandwidth, and power efficiency than traditional CPUs, they also make it far more difficult to optimize (or "tune") the execution of the diverse collection of computational kernels that do the majority of heavy lifting in applications across nearly every field of science and engineering. Tuning these kernels for accelerators is far more difficult because, as compared to standard multicore CPUs, these processors are much more complex in terms of concurrency (with thread counts a thousand fold higher), and much more inscrutable to the developer in the way they schedule resources. Consequently, the three traditional approaches to kernel optimization—compiler-based program transformations, manual tuning, and sequential autotuning—all meet with significant limitations when applied to accelerator systems. In the NSF-funded *Benchtesting Environment for Automated Software Tuning* (BEAST) project, we designed, prototyped and validated a novel methodology that makes the rapid and efficient tuning of computational kernels on accelerators both feasible and sustainable [4, 17, 22, 29, 31, 45]. In the *BEAST OpeN Software Autotuning Infrastructure* (BONSAI) project described below, we propose to use the knowledge and the software that resulted from our research to create a modular, easy to use software toolkit that will enable domain scientists and library developers to efficiently explore and optimize the execution of a wide variety of computational kernels on current and future hybrid platforms (Figure 1).

We call our new approach to optimization "benchtesting" because it generalizes the performance testing model used for traditional benchmarking. This generalization combines an abstract kernel specification and corresponding correctness test, as used by traditional benchmarking, with automated test runs and the use of standard data analysis tools, in order to create an autotuning process that overcomes the limitations of conventional optimization strategies on hybrid platforms. The results of the BEAST project showed 1) that this method is general enough to encompass a diverse collection of kernels that have been identified as fundamental to computational science [4, 17, 29]; 2) that it can generate, in a semi-automatic way, implementations of those kernels with excellent overall performance on different accelerator platforms (CUDA for NVIDIA, OpenCL for AMD) [45]; and 3) that it can provide insight into why different implementations of a given kernel have the performance profiles that they do on given architectures [4]. But as we explain in detail below (Sec. 2.2), the key lesson learned was that the secret to unleashing the transformative potential of this methodology is *the application of parallelism to both the compilation and the execution/data collection phases of the benchtesting process*. Accordingly:

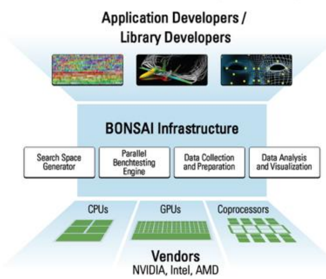


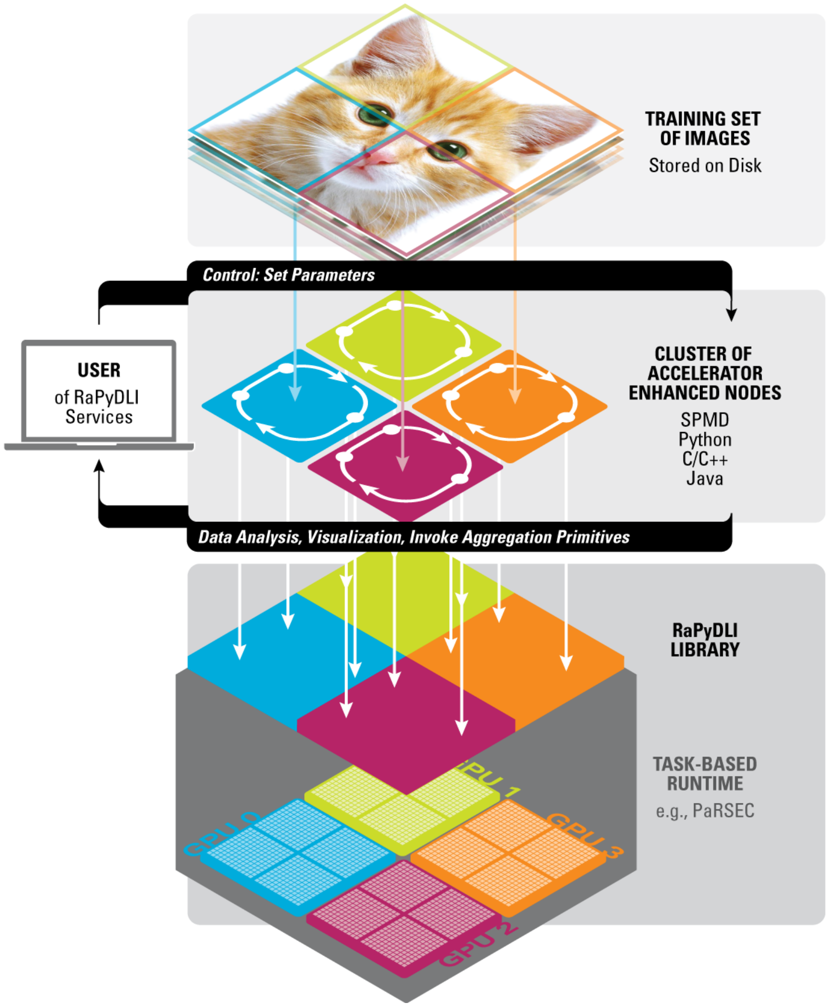
Figure 1: Overview of BONSAI project

The goal of the BONSAI project is to develop a software infrastructure for using parallel hybrid systems at any scale to carry out large, concurrent autotuning sweeps in order to dramatically accelerate the optimization process of computational kernels for GPU accelerators and many-core coprocessors.

2 strategies: Illustrate either big idea or big impact

Lesson 3: Illustration on page 1

- Developing the illustrations helps the PIs understand their story
- Illustration provides a map of the story you're telling
- It's neither a necessary nor a sufficient condition for success
- It's just a competitive edge; proposal must be strong in other respects



Lesson 4: Managing the pipeline

- Submit too many proposals:
  - Burn out; quality of the proposals erodes
  - Not enough research gets done
  - Have to participate in the "ecosystem" in order to succeed.
- Submit too few proposals
  - If we miss a few, the pipeline goes dry, we crash.
- Message: Choose wisely and do it right
  - Match the proposal with the solicitation
  - Half-hearted efforts aren't really worth it
  - Money is not enough: Make sure someone wants to do the work
  - It often takes more than 1 try to get it right
- Sometimes we have to go for it, regardless
- The main question: How do we use AI to do all this stuff more effectively

Some possible uses of AI

- Research current trends and the state of the art
- Create “Red team” proposal review simulator
  - Input the solicitation
  - Describe the expected reviewers (Use reviewer “persona” prompts)
  - Describe the trends and state of the art
  - What questions, criticisms, or suggestions does it see?
- Upgrade proposal formulation and impact
  - With multiple authors, it’s hard to make the story smooth and consistent
  - All good writing is rewriting
  - The painful pieces: Related work, Broader Impacts
- Evaluate fit with and potential for target environments
- How are the agencies using AI in their program development and proposal review process?

Full Disclosure: I asked AI how to use AI

My prompt to ChatGPT:

We are writing a proposal for federal funding in the area of software research for high-performance computing and computational science. We view proposal development as a competitive team writing process. Important elements that go into that process are as follows:

- Evaluate the current state of the art to determine if our proposed project is seen as innovative.
- Develop alternative ways to present the narrative to maximize its impact on expected reviewers.
- Review the draft proposal in light of the funding solicitation to assess whether it meets the criteria for success specified by the solicitation. This review would involve using an AI model for a "Red Team" critique.
- Assess the proposed software technology for how well it fits the targeted environment, such as issues with interoperability, portability, duplicate functionality, and so on.
- Review and analyze previous grants awarded under the targeted federal program to identify common factors that contributed to their success.

Two questions:

1. How, if at all, might AI be used to address these areas of concern?
2. What other tasks is AI being used for, or could be used for, in the writing proposals in this area?