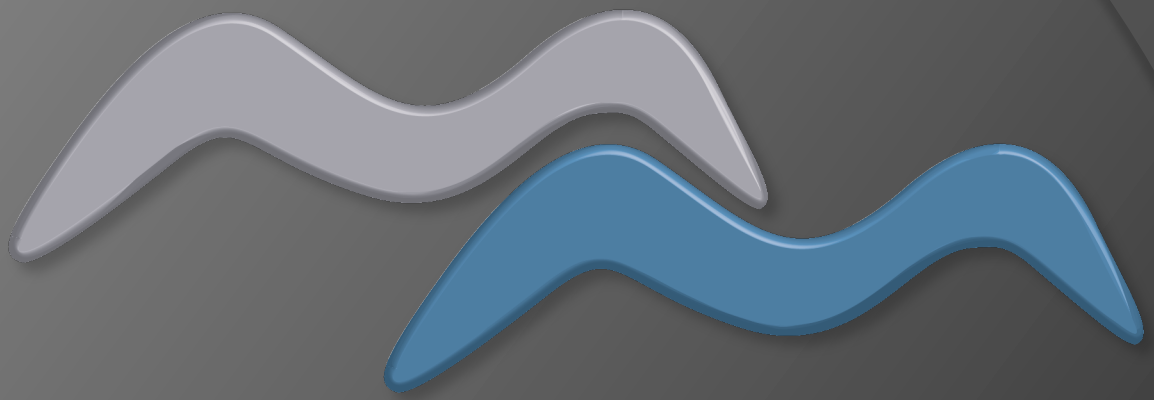

minimal **metrics**

Systems Performance @ Sandia

Philip J. Mucci

phil@minimalmetrics.com

About Me

- Started at Thinking Machines in 1990/1991
- BACS Johns Hopkins
- MSCS University of Tennessee under Jack Dongarra. Active Messages for PVM and PAPI.
- Visiting Scientist IBM Research, LBNL, KTH
- Director of Engineering for .com
- Performance Architect at SiCortex
- (Part Time) Research Consultant at ICL
- Minimal Metrics LLC

About Minimal Metrics

- Consultancy founded in 2011
  - High performance solutions in HPC and the Enterprise
- What we do:
  - Detailed performance analysis
  - Numerical library development
  - Parallel run-time and communication library development
  - Linux kernel tuning, device drivers and BSP
  - Parallel algorithm design
  - System design, technology and procurement guidance.
- We are 100% referral based.

"You'll know when you need us…"

Partners and Collaborators

- Reservoir Labs
- ParaTools Inc
- Scalable Informatics
- Google
- HPE
- IBM
- Cray
- Redhat
- Suse
- University of Tennessee
- University of Oregon
- University of Indiana
- University of New Mexico
- University of Texas A&M
- University of Buffalo
- KTH
- Rice
- Renci

PAPI: Intel Named Events

- Previous event maps had names derived from early Intel documentation.
  - Those names and definitions diverged
  - Intel started publishing event tables silently a few years ago
- Libpfm modified to synchronize Intel's names
  - As identified in Intel Reference Manuals and tools
- Allows all PAPI-based tools to use Intel's methodologies.

PAPI News

- In 2015, PAPI received an NSF SI2 grant for continued development.
- The major (funded) tasks are:
  - Bring papiex back into Open Source (and into this century)
  - PAPI sampling API
  - New PAPI high-level API
  - Integration into an Exascale run-time system or systems
  - Counter validation toolkit
- Minor (unfunded) include:
  - Real build, unit and functional testing
  - Issue tracking
  - Improved documentation

Other PAPI news

- KNL support in process (under NDA)
- ARMv8 64-bit support only for Xgene.
- POWER8 support (done)
- Sparc64 support (in-process)
- Libpfm modifications to merge Intel's event tables with Libpfm's event tables.
  - So one can use Intel's docs, tools and methods with PAPI based tools.

PAPIEX

- *Provides transparent, passive and near-zero overhead monitoring of any application without recompilation*
  - Parallelism: MPI, Pthreads, OpenMP
  - Compute and NUMA (performance counters)
  - Data movement: MPI and I/O
  - Locking and synchronization
  - Memory usage
  - Simple and advanced run modes
  - Trivial to use!

PAPIEX

A Simple Command Line PAPI Tool

- An easy to use command line interface for monitoring and measuring an application.
- As easy to use as /usr/bin/time
- Simple/passive instrumentation API
- Works (identically) out of the box on serial, OpenMP, Pthreads, MPI and Hybrid codes
- Follows entire process/thread tree
- Easy to read textual reports with automatic aggregation. Job, Process, Thread

PAPIEX

- Reports sets of metrics and derived ratios
  - Useful for diagnosis and characterization.
- In addition to PAPI metrics it also reports:
  - MPI communication
  - MPI synchronization
  - Thread synchronization
  - File I/O
  - Memory utilization

PAPIEX

- Also produces machine readable output.
- Stable enough for integration into batch systems. (PerfMiner)
- Easy to extend to additional PAPI components and run-time systems.
- Scalable to tens of thousands of ranks.
- <1% overhead – only instruments slow points, no sampling.

Sample output from Lulesh

```
papiex version                : 1.0.0
papiex build                  : Dec 23 2015/11:41:38
Executable                    : /home/pjmucci/lulesh/luleshoo
Processor                     : Intel(R) Xeon(R) CPU E5-2698 v3 @
2.30GHz
Clockrate (MHz)               : 3600.000000
Hostname                      : shepard-lsm1
Options                       :
MULTIPLEX,MEMORY,PAPI_TOT_INS,PAPI_LST_INS,PAPI_BR_INS,PAPI_LD_IN
S,PAPI_SR_INS,PAPI_TO
T_CYC,PAPI_RES_STL,PAPI_L1_DCM,PAPI_L1_ICM,PAPI_TLB_DM,PAPI_TLB_I
M,PAPI_L2_DCM,PAPI_L2_ICM,PAPI_CA_INV,PAPI_STL_ICY,PA
PI_FUL_ICY,PAPI_BR_CN,PAPI_BR_MSP,PAPI_L2_DCA,PAPI_L2_ICA,NEXTGEN
Domain                        : User
Parent process id             : 80183
Process id                    : 80191
Start                         : Thu Jan 14 12:58:08 2016
Finish                        : Thu Jan 14 12:58:47 2016
Num. of tasks                 : 8

Global derived data:

IPC ..........................................  1.67371e+00
Load Store Ratio .............................  2.94039e+00
Instructions Per Dcache Miss .................  5.38994e+01

Time:
Wallclock seconds ............................  3.88193e+01
IO seconds ...................................  4.41301e-01
No Issue Stall seconds .......................  3.52104e+00
Resource Stall seconds .......................  1.95831e+02

Cycles:
Cycles In Domain .............................  1.57857e+12
Real Cycles ..................................  3.54279e+12
Running Time In Domain % .....................  4.45572e+01
Virtual Cycles ...............................  2.14767e+12
IO Cycles % ..................................  4.48426e-02
MPI Cycles % .................................  3.30127e+00
MPI Sync Cycles % ............................  6.41025e-01
Thread Sync Cycles % .........................  1.02536e-03

Instructions:
Total Instructions ...........................  2.64208e+12
Memory Instructions % ........................  5.24352e+01
Memory Instructions % ........................  5.24352e+01
Branch Instructions % ........................  6.46326e+00

Memory:
Load Store Ratio .............................  2.94039e+00
L1 Data Misses Per 1000 Load Stores ..........  3.53829e+01
L1 Data Misses Per 1000 Load Stores ..........  3.53829e+01
L1 Instruction Misses Per 1000 Instructions ..  5.52103e-02
L2 Data Misses Per 1000 L2 Load Stores .......  2.68006e+02
L2 Instruction Misses Per 1000 L2 Instructions   9.52746e+02
Data TLB Misses Per 1000 Load Stores .........  1.74406e-01
Instruction TLB Misses Per 1000 Instructions .  4.54513e-03

Stalls:
Resource Stall Cycles % ......................  4.46602e+01

No Issue Cycle % .............................  8.02989e-01
Full Issue Cycle % ...........................  3.20748e+01
Branch Misprediction % .......................  1.99332e-01
```

Sample output from Lulesh

```
Global counts data:
Event                                   Sum          Min          Max          Mean         CV
IO cycles ..............................  1.58868e+09  1.79825e+08  2.32669e+08  1.98585e+08  8.29237e-02
MPI Sync cycles ........................  2.27102e+10  8.53654e+08  4.52004e+09  2.83877e+09  3.46667e-01
MPI cycles .............................  1.16957e+11  9.87963e+08  2.15312e+10  1.46197e+10  4.35992e-01
Mem. heap KB ...........................  4.58576e+05  5.63800e+04  5.79560e+04  5.73220e+04  7.99719e-03
Mem. library KB ........................  8.40640e+04  1.05080e+04  1.05080e+04  1.05080e+04  0.00000e+00
Mem. locked KB .........................  0.00000e+00
Mem. resident peak KB ..................  3.53036e+05  4.39560e+04  4.43800e+04  4.41295e+04  3.44141e-03
Mem. shared KB .........................  4.29840e+04  5.37200e+03  5.38000e+03  5.37300e+03  4.92416e-04
Mem. stack KB ..........................  4.38800e+03  5.44000e+02  5.52000e+02  5.48500e+02  6.76044e-03
Mem. text KB ...........................  1.47200e+03  1.84000e+02  1.84000e+02  1.84000e+02  0.00000e+00
Mem. virtual peak KB ...................  0.00000e+00
PAPI_BR_CN .............................  1.35132e+11  1.29001e+10  1.86272e+10  1.68915e+10  9.78988e-02
PAPI_BR_INS ............................  1.70764e+11  1.50636e+10  2.39208e+10  2.13455e+10  1.19306e-01
PAPI_BR_MSP ............................  3.40387e+08  1.48287e+07  8.58754e+07  4.25484e+07  4.84782e-01
PAPI_CA_INV ............................  3.64323e+10  3.64042e+09  6.58882e+09  4.55404e+09  2.36237e-01
PAPI_FUL_ICY ...........................  5.06323e+11  5.73630e+10  6.65820e+10  6.32904e+10  4.13402e-02
PAPI_L1_DCM ............................  4.90186e+10  4.97736e+09  8.02939e+09  6.12733e+09  1.73027e-01
PAPI_L1_ICM ............................  1.45870e+08  1.62856e+07  1.99725e+07  1.82337e+07  5.84540e-02
PAPI_L2_DCA ............................  3.57609e+10  4.08337e+09  4.78990e+09  4.47011e+09  4.94141e-02
PAPI_L2_DCM ............................  9.58413e+09  9.93431e+08  1.65480e+09  1.19802e+09  1.82430e-01
PAPI_L2_ICA ............................  1.45870e+08  1.62856e+07  1.99725e+07  1.82337e+07  5.84540e-02
PAPI_L2_ICM ............................  1.38977e+08  1.49945e+07  1.91577e+07  1.73721e+07  7.14455e-02
PAPI_LD_INS ............................  1.03379e+12  1.24601e+11  1.35564e+11  1.29224e+11  2.73702e-02
PAPI_LST_INS ...........................  1.38538e+12  1.67603e+11  1.80794e+11  1.73172e+11  2.45523e-02
PAPI_RES_STL ...........................  7.04993e+11  8.38043e+10  9.91089e+10  8.81242e+10  5.27713e-02
PAPI_SR_INS ............................  3.51583e+11  4.30028e+10  4.52294e+10  4.39479e+10  1.63802e-02
PAPI_STL_ICY ...........................  1.26758e+10  1.25323e+09  1.86245e+09  1.58447e+09  1.11680e-01
PAPI_TLB_DM ............................  2.41618e+08  1.86280e+07  3.41084e+07  3.02023e+07  1.54060e-01
PAPI_TLB_IM ............................  1.20086e+07  1.03715e+06  2.24464e+06  1.50107e+06  2.19540e-01
PAPI_TOT_CYC ...........................  1.57857e+12  1.91610e+11  2.03771e+11  1.97321e+11  2.34800e-02
PAPI_TOT_INS ...........................  2.64208e+12  3.21143e+11  3.44709e+11  3.30260e+11  2.36111e-02
Real cycles ............................  3.54279e+12  4.42738e+11  4.43020e+11  4.42849e+11  1.95304e-04
Real usecs .............................  1.54391e+09  1.92940e+08  1.93063e+08  1.92989e+08  1.95305e-04
Thr Sync cycles ........................  3.63265e+07  2.93024e+06  7.10849e+06  4.54081e+06  3.10946e-01
Virtual cycles .........................  2.14767e+12  2.60080e+11  2.76769e+11  2.68459e+11  2.28375e-02
Virtual usecs ..........................  5.96575e+08  7.22444e+07  7.68804e+07  7.45719e+07  2.28375e-02
Wallclock usecs ........................  3.88193e+07  3.88039e+07  3.88296e+07  3.88148e+07  2.01547e-04
```

mpipex

- Extended mpiP
- Per Rank MPI Call Statistics plus
  - Message size histogram
  - Point to point statistics: count, size, time
  - High-performance, small-footprint build
  - 2 CSV output files
- Small patches to original mpiP code base
- Honors MPI_Pcontrol()

hpcex

- You do this:

```
mpirun –np 8 hpcex –e UOPS_ISSUED.STALL_CYCLES@1000000 –e
PAPI_L1_TCM@1000000 lulesh –s 48
```

- It does this:

```
# recover binary structure
hpcstruct ./lulesh
# setup events and thresholds
mpirun –np 8 hpcrun ––event UOPS_ISSUED.STALL_CYCLES@1000000
––event PAPI_L1_TCM@1000000 lulesh –s 48
# process output into database
hpcprof –S lulesh.hpcstruct –I ./'*' lulesh-hpctoolkit-db
# To view:
# hpcviewer lulesh-hpctoolkit-db
```

Next for papiex

- OMPT interface
  - Per-platform event maps with -a
  - OMPT: OMP profiling layer
  - OpenCL/CUDA profiling layer
- KNC/KNL/Power8/ARM64 testing
- Memory allocator
- Instrumentation improvements

HPX Performance

- Question: How would one tune such a code?

"*The single most important impediment to good parallel performance is **still** single-node performance*"

William Gropp, Argonne National Lab.

- Start by measuring tasks, lightweight, user-level thread.

Measurement in HPX

- PMU counters work either at the thread level or the core level.
  - The latter is privileged
- We can intercept pthread calls but…
  - It's execution doesn't resemble work in program order!
  - A single thread can execute many tasks in an arbitrary order!
- State must be managed in/by the task scheduler!
  - And eventually record some kind of context (or root)

APEX

- The introspection and runtime adaptation component for the OpenX stack.
  - Fully integrated into both versions of HPX at the right place
- Autonomic Performance Environment for Exascale (University of Oregon): http://www.nic.uoregon.edu/~khuck/APEX-Scalable-Tools-Workshop-2015.pdf

Funded by SciDAC X-Stack project: "XPRESS" #DE-SC0008638.

APEX Extensions

- Handle multi-dimensional data
  - Only time before
- Add PAPI support
  - Overhead of PAPI of concern, but acceptable for prototype.
- Output generation
  - Emits CSV and human-readable file at the end of the run.
- Additional callbacks.

Output sample

Action	#calls	minimum	mean	maximum	total	stddev	% total	PAPI_TOT_CYC	PAPI_TOT_INS
broadcast_call_shutdown_fun...	2	--n/a--	5.12e-05	--n/a--	1.02e-04	--n/a--	0.000	2.45e+05	4.18e+04
broadcast_call_startup_func...	2	--n/a--	3.94e-05	--n/a--	7.87e-05	--n/a--	0.000	1.79e+05	3.55e+04
call_shutdown_functions_action	2	--n/a--	2.64e-04	--n/a--	5.29e-04	--n/a--	0.000	1.49e+06	1.77e+06
call_startup_functions_action	2	--n/a--	7.49e-04	--n/a--	1.50e-03	--n/a--	0.000	2.30e+06	1.59e+06
create_performance_counter_...	1	--n/a--	1.06e-04	--n/a--	1.06e-04	--n/a--	0.000	2.68e+05	1.36e+05
hpx::lcos::local::dataflow:...	450000	--n/a--	6.27e-04	--n/a--	2.82e+02	--n/a--	83.248	8.05e+11	8.13e+11
hpx_main	21	--n/a--	8.38e-02	--n/a--	1.76e+00	--n/a--	0.520	8.23e+08	1.22e+09
load_components_action	2	--n/a--	2.25e-03	--n/a--	4.51e-03	--n/a--	0.001	9.41e+06	1.05e+07
performance_counter_get_cou...	1	--n/a--	2.43e-05	--n/a--	2.43e-05	--n/a--	0.000	6.09e+04	1.18e+04
performance_counter_start_a...	1	--n/a--	2.31e-05	--n/a--	2.31e-05	--n/a--	0.000	5.10e+04	8.32e+03
pre_main	1	--n/a--	1.18e-03	--n/a--	1.18e-03	--n/a--	0.000	3.11e+06	3.99e+06
primary_namespace_bulk_serv...	32	--n/a--	1.36e-05	--n/a--	4.36e-04	--n/a--	0.000	9.98e+05	2.71e+05
primary_namespace_service_a...	4	--n/a--	1.80e-05	--n/a--	7.21e-05	--n/a--	0.000	1.73e+05	3.34e+04
run_helper	1	--n/a--	2.83e-04	--n/a--	2.83e-04	--n/a--	0.000	3.10e+05	1.25e+05
symbol_namespace_service_ac...	9	--n/a--	3.00e-05	--n/a--	2.70e-04	--n/a--	0.000	6.63e+05	1.33e+05

Human-readable output

```
"task","num calls","total cycles","total microseconds","PAPI_TOT_CYC","PAPI_TOT_INS"
"broadcast_call_shutdown_functions_action",2,255876,102,245051,41768
"broadcast_call_startup_functions_action",2,196797,79,179118,35532
"call_shutdown_functions_action",2,1321415,529,1487698,1770180
"call_startup_functions_action",2,3747616,1499,2301975,1585683
"create_performance_counter_action",1,263872,106,267755,135655
"hpx::lcos::local::dataflow::execute",450000,705076722423,282007120,805365897395,812803811673
"hpx_main",21,4401925230,1760623,823088150,1215484203
"load_components_action",2,11273052,4509,9410400,10529432
"performance_counter_get_counter_value_action",1,60712,24,60903,11815
"performance_counter_start_action",1,57685,23,51017,8318
"pre_main",1,2958140,1183,3108772,3990503
"primary_namespace_bulk_service_action",32,1091095,436,997978,271430
"primary_namespace_service_action",4,180309,72,173037,33363
"run_helper",1,706812,283,310269,125023
"symbol_namespace_service_action",9,674929,270,662582,132997
```

CSV output

Future Development

- Simple command-line tool
  - Configure measurements and extract data
  - papiex?
- Develop low-overhead PMU access methods
  - rdpmc() from ring 3 + mmap()
- Additional platforms
  - ARM64, KNL, GPGPU (x86_64, KNC, POWER8)
- Additional programming models
  - MPI-T, UPC++, TBB, Cilk, OCR, Legion, Qthreads, OmpSs, GASPI, etc. (HPX, HPX-5, OpenMP)