allinea

High performance tools to debug, profile, and analyze your applications

Monitoring and improving application energy efficiency

Mark O'Connor, VP Product Management

allinea FORGE

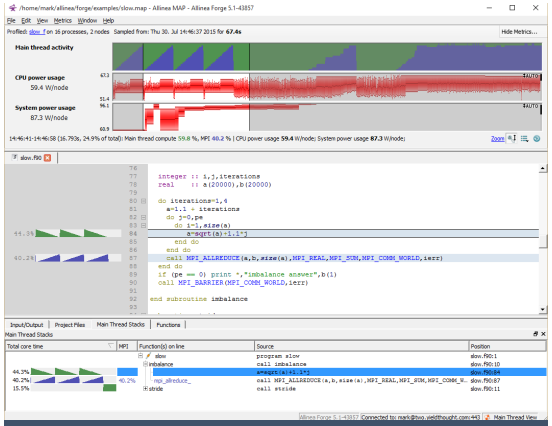
allinea DDT

allinea MAP

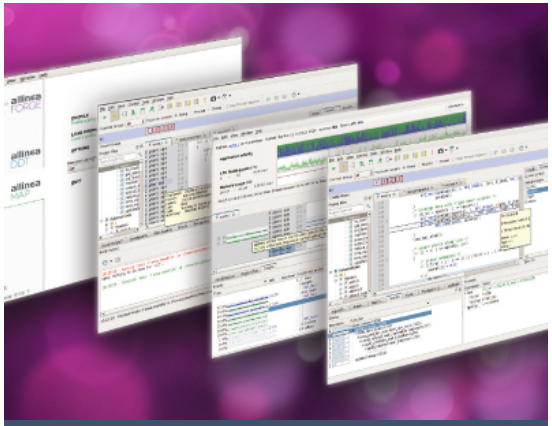
allinea PERFORMANCE REPORTS

Monitoring and improving application energy efficiency

Background: why am I here? (and other important life questions)



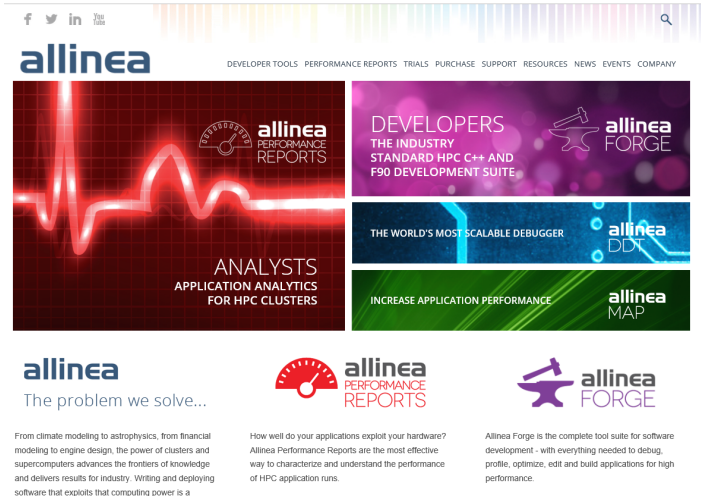
Today: what can we **do** with application-level energy profiling?



Active research: energy-aware scheduling, superoptimizing compilers, …

allinea

Allinea Software



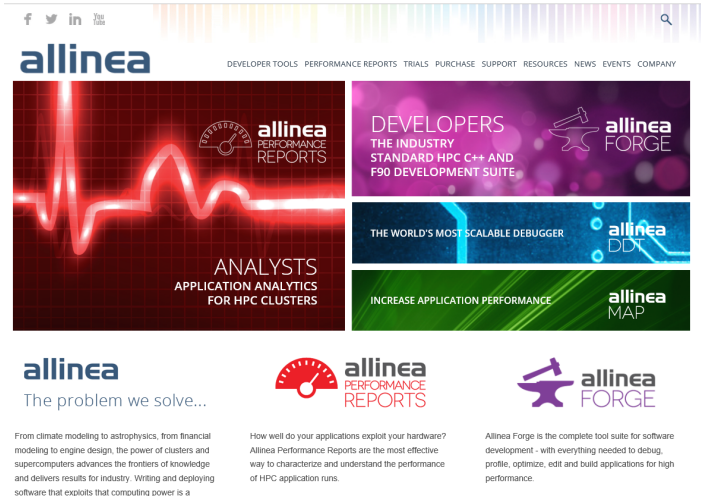
Why am I here?

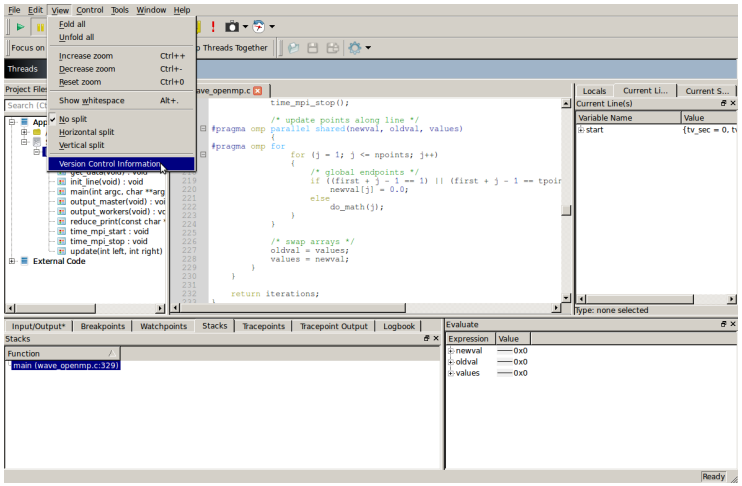
High-performance software tools for HPC

- Performance
- Reliability
- Ease of use

allinea

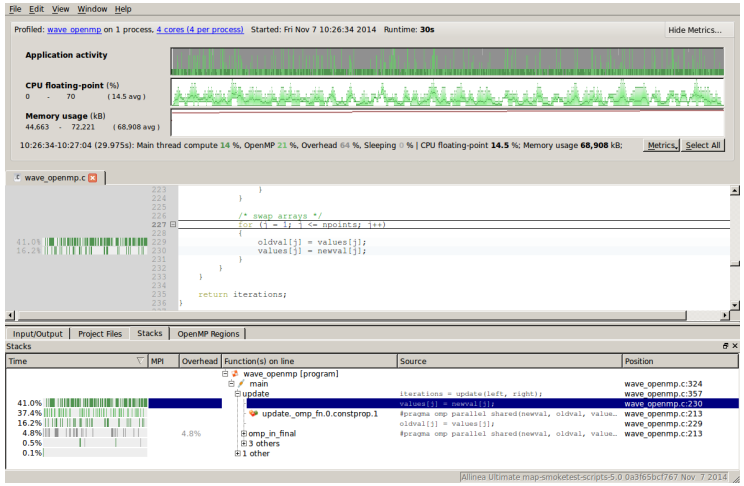
Allinea Forge: Development Tools





allinea DDT

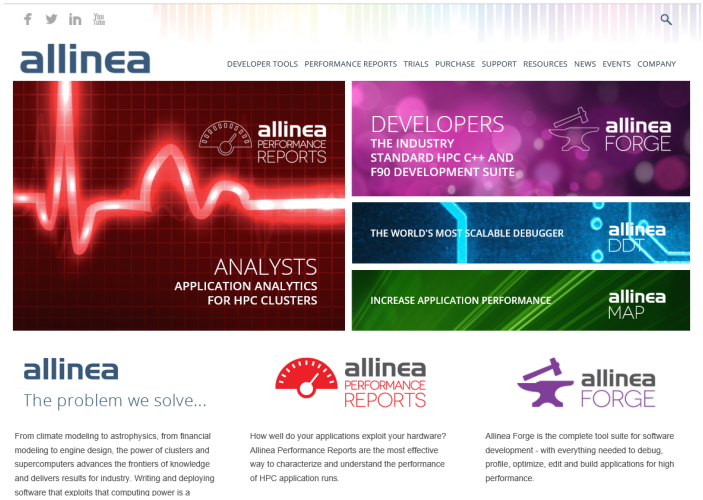
For debugging and profiling codes



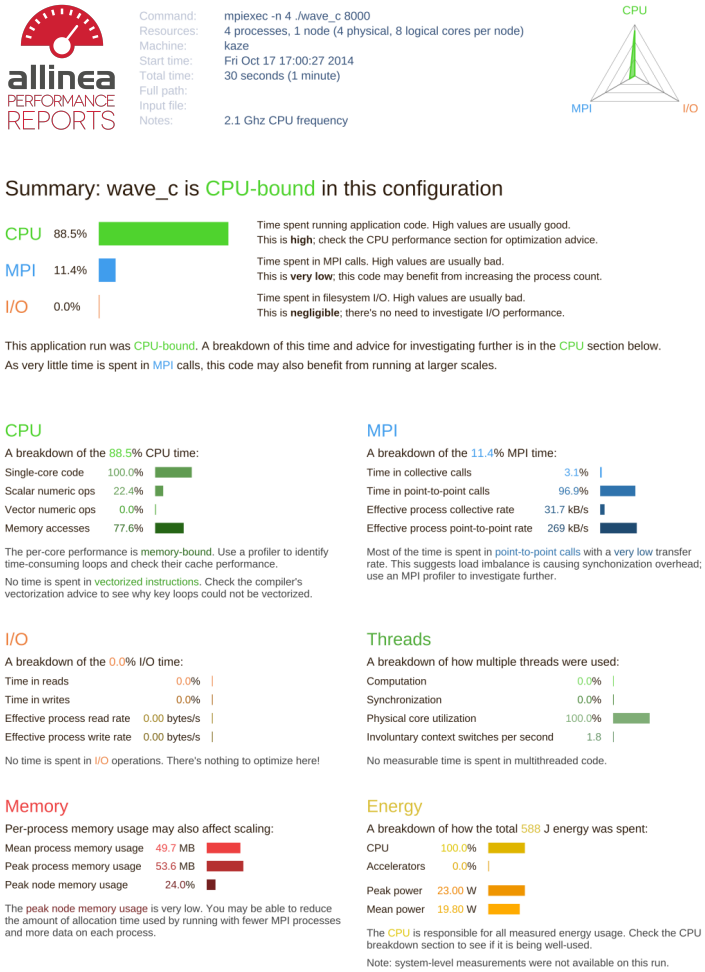
allinea MAP

allinea

Allinea Performance Reports

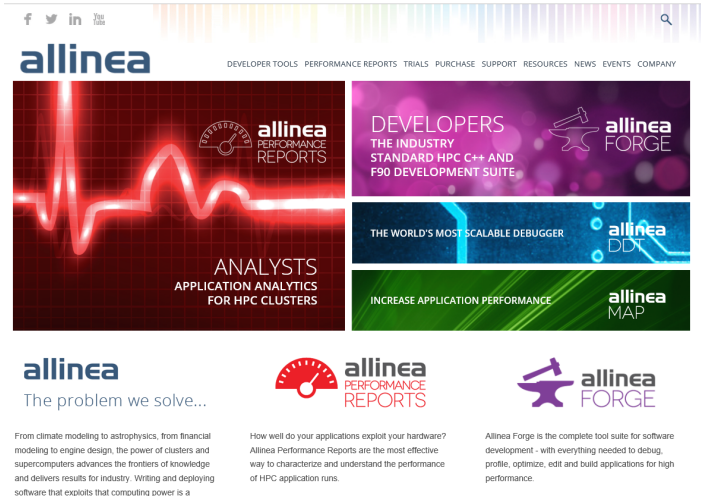


For monitoring and tuning applications



allinea

Allinea Software



Customers

Los Alamos NATIONAL LABORATORY EST. 1943

Argonne NATIONAL LABORATORY

OAK RIDGE National Laboratory

NERSC NATIONAL ENERGY RESEARCH SCIENTIFIC COMPUTING CENTER

University of California Lawrence Livermore National Laboratory

cea

HLRS

BAE SYSTEMS

TOTAL

BLUE WATERS SUSTAINED PETASCALE COMPUTING

TACC

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

WesternGeco

BERKELEY LAB

ICHEC Irish Centre for High-End Computing

lrz

IFERC

GENCI

ONERA THE FRENCH AEROSPACE LAB

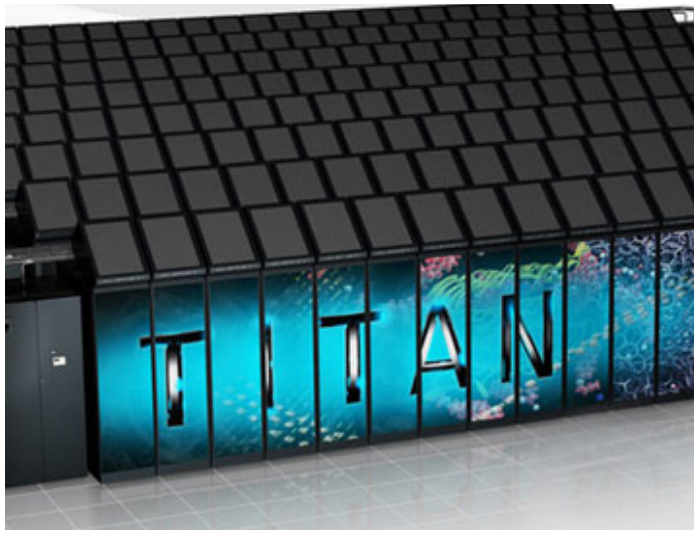
SHARCNET

archer

SNIC

allinea

System-level energy profiling

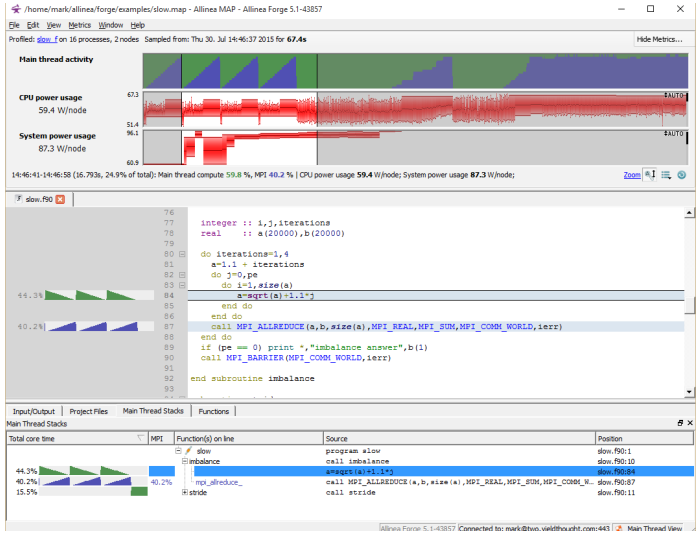


What does **system**-level profiling change?

- Future system planning
- Understanding of peak load characteristics
- CPU clock speed / turbo boost
- Use of accelerators
- PUE computation and reporting

allinea

Application-level energy profiling



What does **application-** level profiling change?

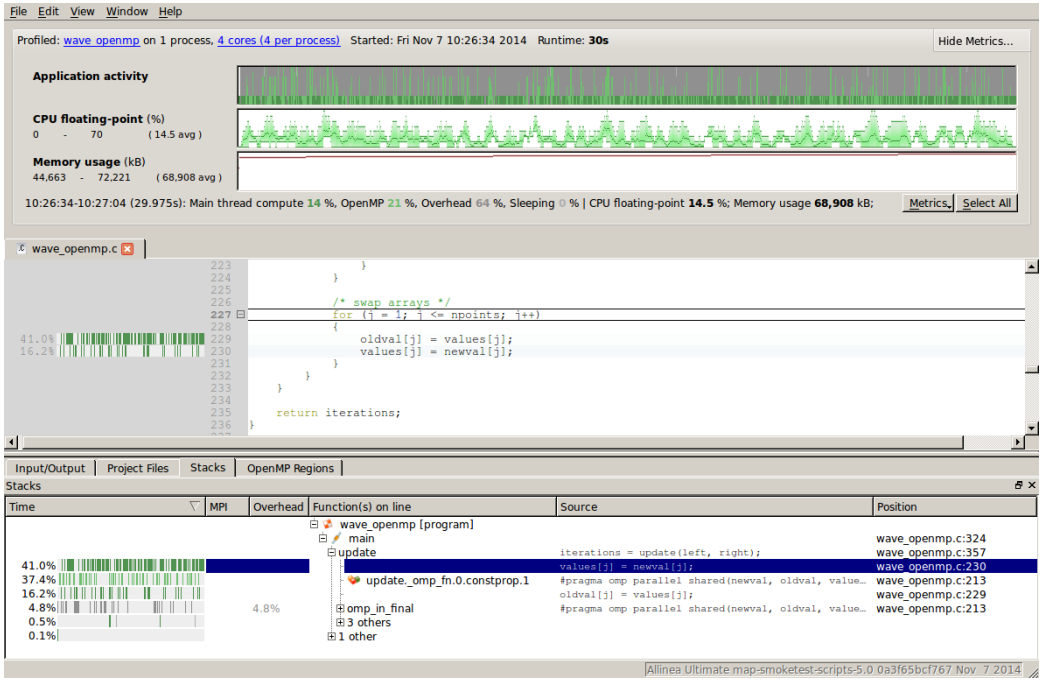
- Show trapped capacity per application run
- Base operational requirements on key application behavior
- Job size and node placement
- Predictive scheduling (power cap, temperature, I/O, …)
- Awareness + visibility in user and IT teams
- Application source code improvements

allinea

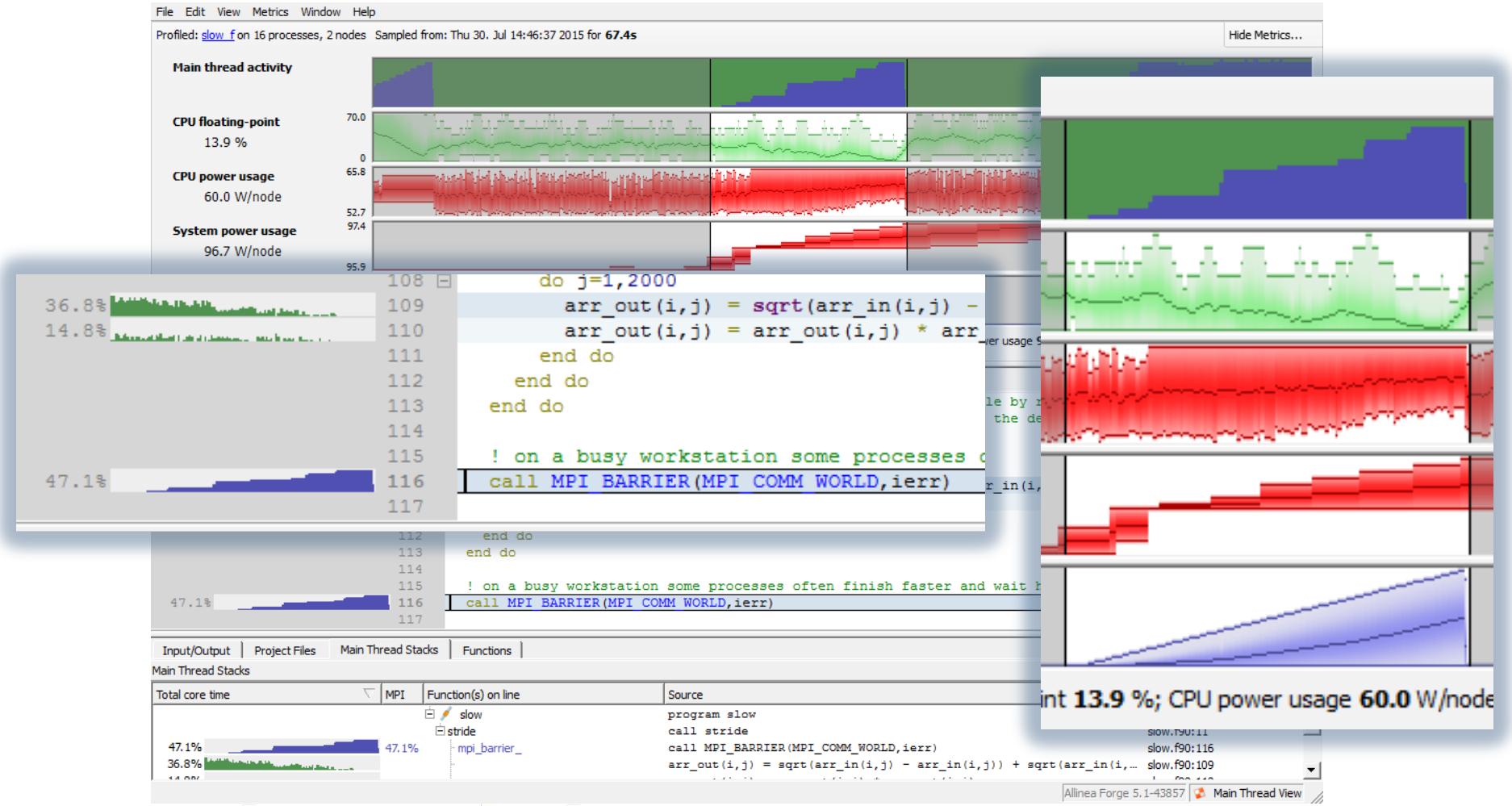
Application-level Profiling for Developers

A quick introduction to Allinea MAP:

- Small data files
- <5% slowdown
- No instrumentation or recompilation
- Proven at over 10,000 cores



Energy Profiling for Developers



Energy Profiling for Developers

CPU fp vector
24.9 %
70.0
0
CPU power usage
58.5 W/node
67.3
51.7

14:46:56-14:47:07 (11.533s, 17.1% of total): Main thread compute **100.0** %, MPI **0.0** % | CPU fp vector **24.9** %; CPU power usage **58.5** W/node;

Hide Metrics...
AUTO
AUTO
Zoom

slow.f90

```
104   ! note: some compilers are able to optimize this trivi
105   ! the inner loops - in that case recompile with -O0 in
106   do l=1,150
107     do i=1,8000
108       do j=1,2000
109         arr_out(i,j) = sqrt(arr_in(i,j) - arr_in(i,j)) +
110         arr_out(i,j) = arr_out(i,j) * arr_out(i,j)
111       end do
112     end do
113   end do
114
```

0.1%
78.0%
21.8%

ample by reordering
of the default -O3

(arr_in(i,j) + arr_in(i,j))

CPU fp vector
34.5 %
90.0
0
CPU power usage
59.3 W/node
68.4
52.4

14:47:15-14:47:21 (6.204s, 9.2% of total): Main thread compute **100.0** %, MPI **0.0** % | CPU fp vector **34.5** %; CPU power usage **59.3** W/node;

slow.f90

```
113   end do
114
115   ! on a busy workstation some processes often finish faster and wait here
116   call MPI_BARRIER(MPI_COMM_WORLD,ierr)
117
118   ! efficient memory access pattern (incrementing i in the inner loop)
119   do l=1,150
120     do j=1,2000
121       do i=1,8000
122         arr_out(i,j) = sqrt(arr_in(i,j) - arr_in(i,j)) + sqrt(arr_in(i,j)
123         arr_out(i,j) = arr_out(i,j) * arr_out(i,j)
124       end do
```

84.0%
16.0%

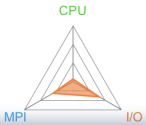
Position
slow.f90:1
slow.f90:11
slow.f90:109
slow.f90:110

5.1-43857 Main Thread View

Application-level Profiling for Users and Admins

allinea PERFORMANCE REPORTS

Executable: MADbench2
Resources: 16 processes, 1 node
Machine: sandybridge2
Start time: Mon Nov 4 12:27:50 2013
Total time: 109 seconds (2 minutes)
Full path: /tmp/MADbench2
Notes: 12-core server / HDD / 16 readers + writers



Summary: MADbench2 is I/O-bound in

The total wallclock time was spent as follows:

CPU	4.8%	Time spent running appl This is **low**; it may be wo
MPI	41.3%	Time spent in MPI calls. This is **average**; check t
I/O	53.9%	Time spent in filesystem This is **high**; check the I

This application run was I/O-bound. A breakdown of this time and advi

CPU

A breakdown of how the 4.8% total CPU time was spent:

Scalar numeric ops	4.9%
Vector numeric ops	0.1%
Memory accesses	95.0%
Other	0.0

The per-core performance is memory-bound. Use a profiler to identify time-consuming loops and check their cache performance.

No time was spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

I/O

A breakdown of how the 53.9% total I/O time was spent:

Time in reads	3.7%
Time in writes	96.3%
Estimated read rate	272 Mb/s
Estimated write rate	7.06 Mb/s

Most of the time is spent in write operations, which have a very low transfer rate. This may be caused by contention for the filesystem or inefficient access patterns. Use an I/O profiler to investigate which write calls are affected.

MPI

Of the
Time i
Time i
Estima
Estima

All of t
This su
synchr
MPI pr

Mer

Per-pr

Mean process memory usage	160 Mb
Peak process memory usage	173 Mb
Peak node memory usage	17.2%

The peak node memory usage is low. You may be able to reduce the total number of CPU hours used by running with fewer MPI processes and more data on each process.

CPU

A breakdown of how the 4.8% total CPU time was spent:

Scalar numeric ops	4.9%
Vector numeric ops	0.1%
Memory accesses	95.0%
Other	0.0

The per-core performance is memory-bound. Use a profiler to identify time-consuming loops and check their cache performance.

No time was spent in vectorized instructions. Check the compiler's vectorization advice to see why key loops could not be vectorized.

Actionable metrics for each application run

or instrumentation

verhead

application workload

Compare architectures, upgrades and more

allinea

Quantify gains immediately

allinea PERFORMANCE REPORTS

CPU RUN

CPU

A breakdown of the 94.6% CPU time:

Scalar numeric ops	11.7%
Vector numeric ops	0.0%
Memory accesses	88.2%
Waiting for accelerators	0.0%

The per-core performance is memory-bound. Use a profiler to identify time-cons

No time is spent i vectorization advi

Energy

A breakdown of how the 3.6 Wh was used:

CPU	62.9%
System	37.1%
Mean node power	92.4 W
Peak node power	94 W

Significant energy is wasted during MPI communications. It may be more efficient to use fewer nodes with more data on each node.

Significant time is spent waiting for memory accesses. Reducing the CPU clock frequency could reduce overall energy usage.

Accelerators

A breakdown of how accelerators were used:

GPU utilization	0.0%
Global memory accesses	0.0%
Mean GPU memory usage	0.0%
Peak GPU memory usage	0.0%

GPUs are available but are not used. Identify suitable hot loops with a profiler and try offloading them to the accelerator.

The peak device memory usage is low. It may be more efficient to offload a larger portion of the dataset to each device.

GPU RUN

CPU

A breakdown of the 70.2% CPU time:

Scalar numeric ops	2.5%
Vector numeric ops	0.0%
Memory accesses	39.7%
Waiting for accelerators	61.7%

Most of the time is spent waiting for accelerators. Use asynchrono

The per-cor identify time

Energy

A breakdown of how the 2.84 Wh was used:

CPU	28.4%
System	71.6%
Mean node power	163 W
Peak node power	175.8 W

Energy usage appears to be optimal.

Accelerators

A breakdown of how accelerators were used:

GPU utilization	92.5%
Global memory accesses	40.4%
Mean GPU memory usage	9.6%
Peak GPU memory usage	15.2%

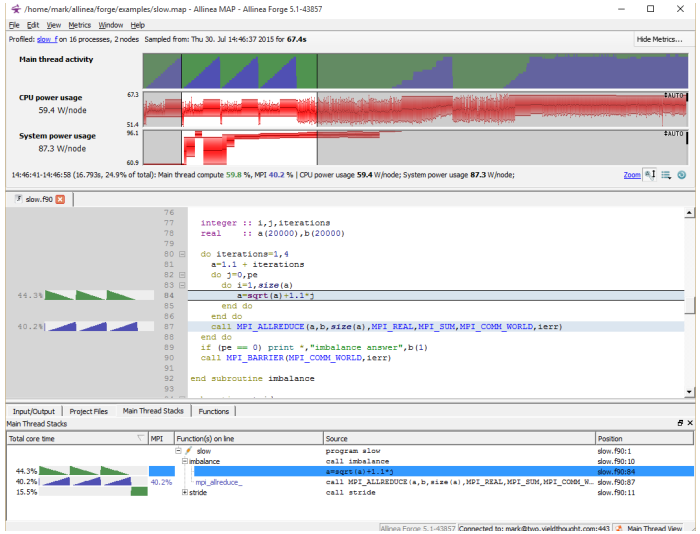
Significant time is spent in global memory accesses. Try modifying kernels to use shared memory instead and check for bad striding patterns.

The peak device memory usage is low. It may be more efficient to offload a larger portion of the dataset to each device.

allinea

Application-level energy profiling

allinea PERFORMANCE REPORTS

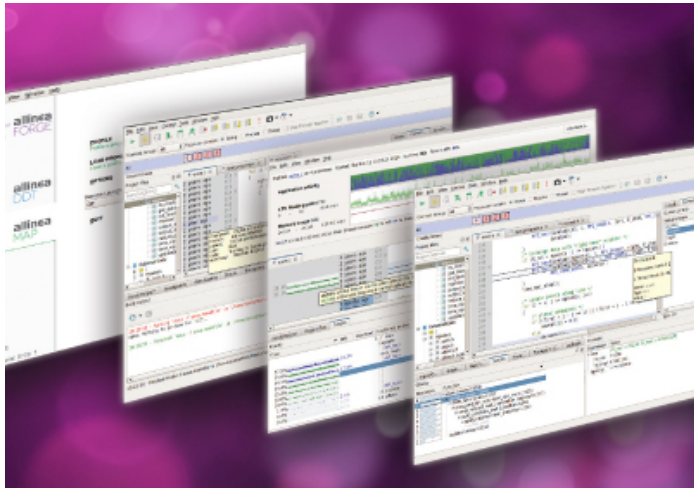


What does application-level profiling change?

- Trapped capacity at the application level
- Base operational requirements on key application behavior
- Job size and node placement
- Predictive scheduling (power, temperature, I/O, …)
- Awareness + visibility in user and IT teams
- Application source code improvements

allinea

Allinea Software: Energy Efficiency Research



Active research

Key ongoing projects:

- Integration with SLURM to allow data- and energy-aware scheduling decisions
- Energy cost model for superoptimizing compiler
- Feed machine learning algorithm to choose optimal compilation flags

allinea

Total Software Energy Reporting and Optimization

Superoptimizing compiler

- MAP provides function-level performance data to the compiler
- EMBECOSM's stochastic superoptimizer targets most energy-costly functions
- Reruns under MAP to measure impact of changes

Machine Guided Energy Efficient Compilation

- Building on existing MAGEEC project for embedded systems
- Uses machine learning to predict compiler option combinations that boost energy efficiency
- Extending to take additional data from MAP in the input vector

allinea

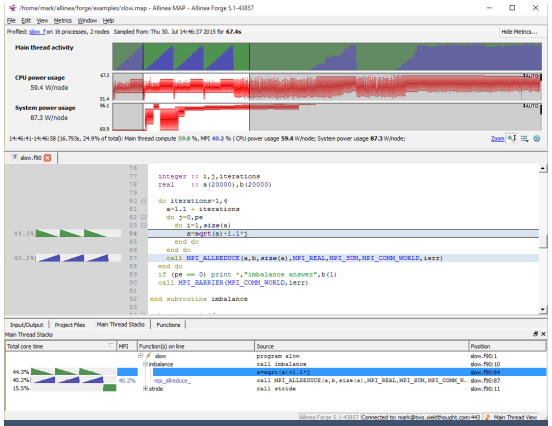
Horizon 2020: Next Generation I/O for Exascale

Work package overview	Target: 50% reduction in energy to solution when compared to today's systems
	Target: 20x improved I/O performance for real applications
	Performance Reports will feed energy and I/O measurements into the SLURM workload scheduler
Scheduler integration	Preload data onto nodes before job starts or schedule work close to existing data location
	Schedule compute-intensive jobs on cooler parts of the system
	Limit clock speed on jobs known to be memory-bound

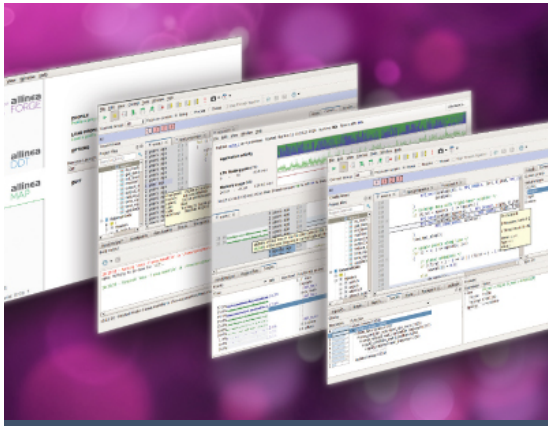
allinea

Monitoring and improving application energy efficiency

Allinea: high-performance software tools for HPC



Today: application-level energy profiling and trapped capacity reporting



Active research: energy-aware scheduling and superoptimizing compilers

allinea

allinea

High performance tools to debug, profile, and analyze your applications

Thank-you!
Q&A

Mark O'Connor, VP Product Management

allinea FORGE

allinea DDT

allinea MAP

allinea PERFORMANCE REPORTS