Software, Libraries, and Tools for Multi-Core Programming

Fall Creek Falls Conference
October 21, 2006

Richard F. Barrett

Scientific Computing Group and
Future Technologies Group
Oak Ridge National Laboratory
Oak Ridge, TN 37831

Acknowledgments

- NLCF and INCITE project teams.
- Scientific Computing@ORNL
- Future Technologies@ORNL.
- Sameer Shende, Rich Lee.

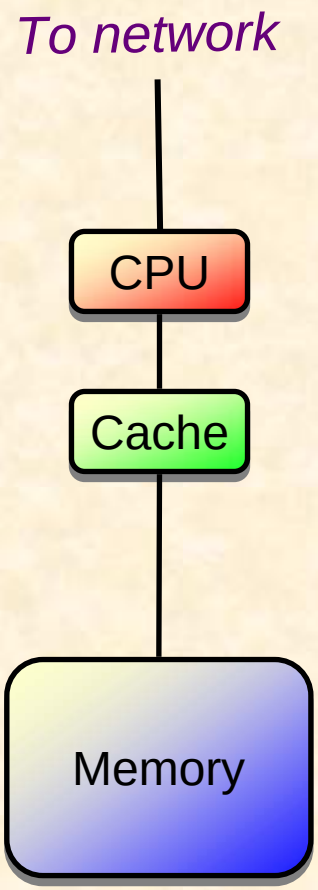
Single- to dual-core comparison

Jaguar

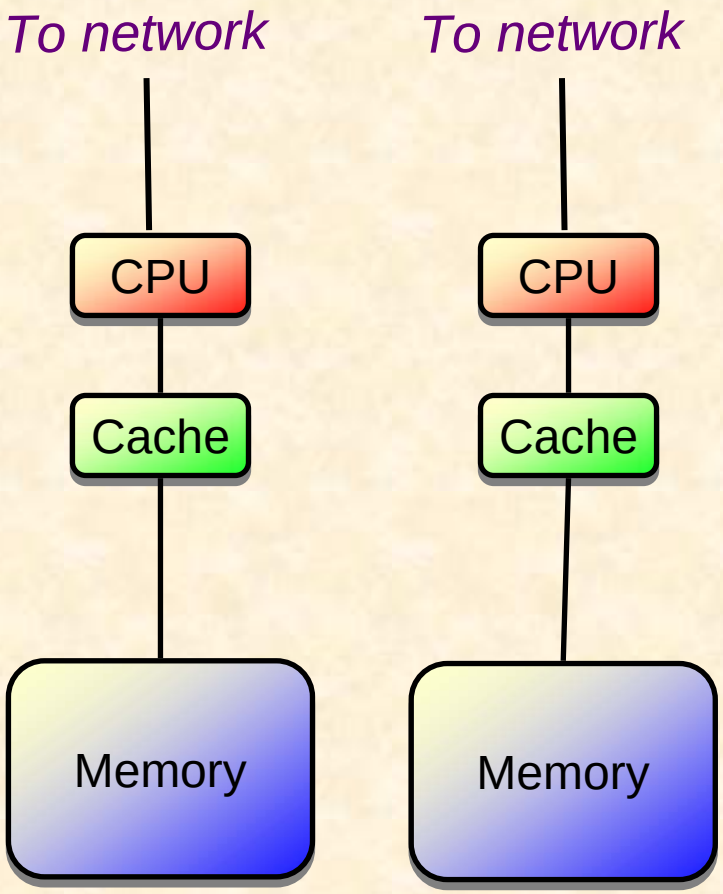
Code	*Problem*	*-SN*	*-VN*	*-SN$_t$/-VN$_t$*
AORSA	*4096 procs; 350x350 grid*	82.08 min	87.66 min	93.6
LAMMPS	*4096 procs; 290K atoms*	5756.5 psec/day	5834.4 psec/day	98.7
GTC	*Benchmark; 4800 procs*	30.60 min	32.08 min	95.4
POP	*64 pes 1 deg*	13.83 secs	16.87 secs	82.0
CAM	d-26 FVCAM3.1	3.68 simyrs/real day	3.29 simyrs/real day	89.4*
HYCOM	Benchmark; 385 procs	2188.2 sec	2709.6 sec	80.8

** -VN$_t$/-SN$_t$*

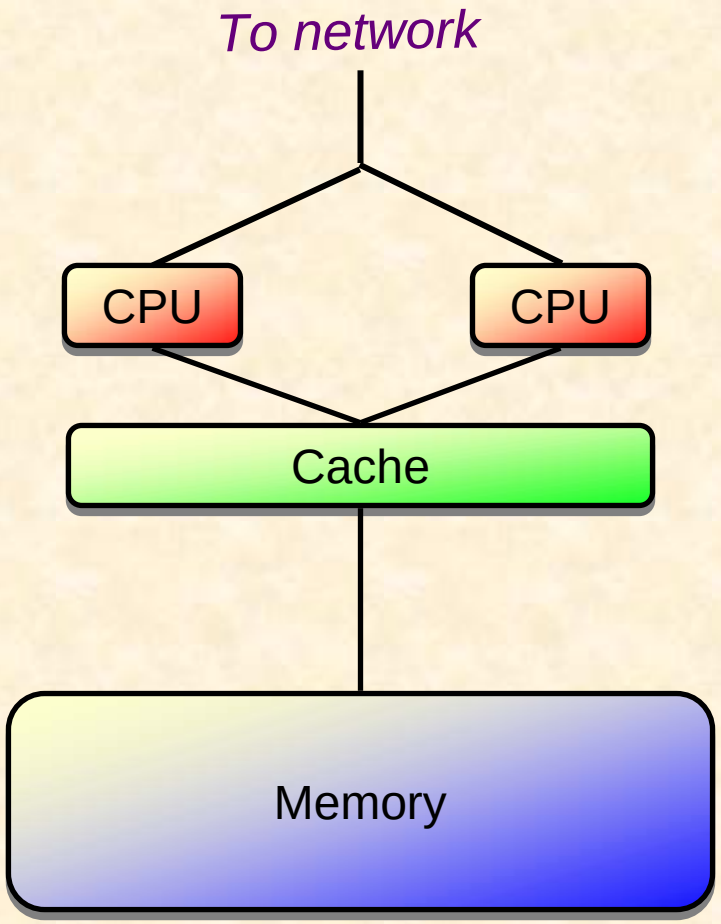
Evolution of the multi-core processor



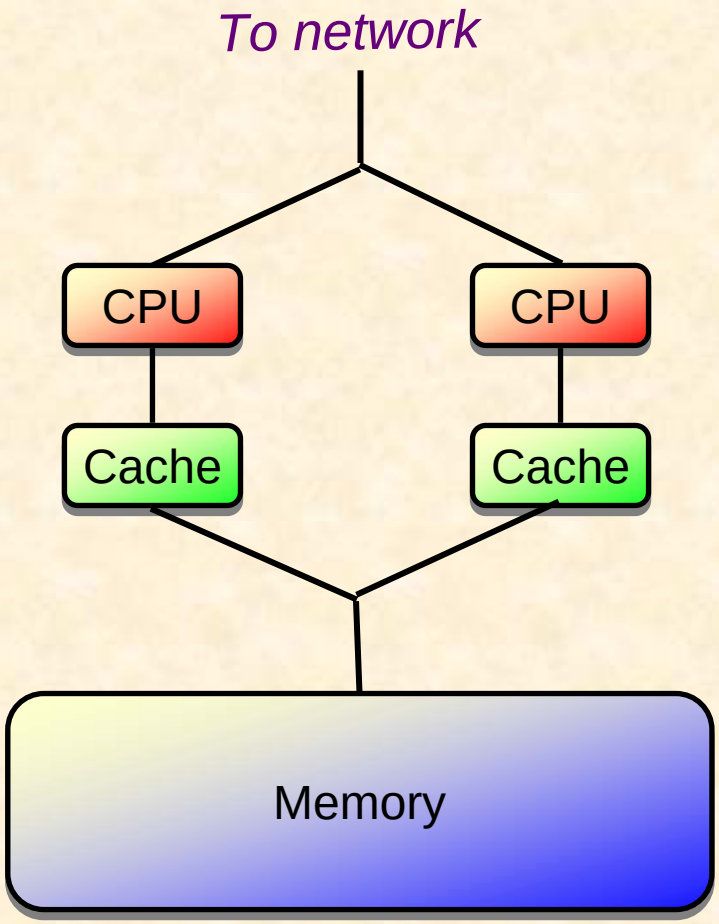
Evolution of the multi-core processor



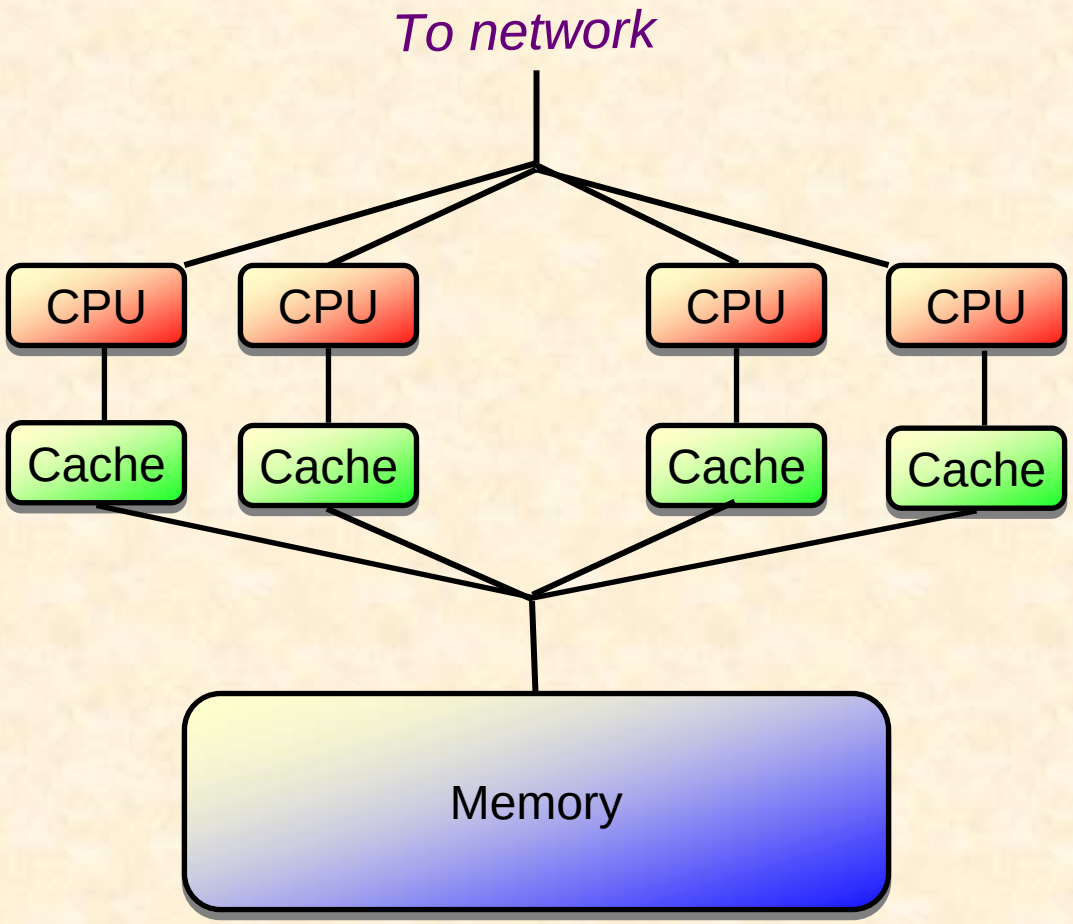
Evolution of the multi-core processor



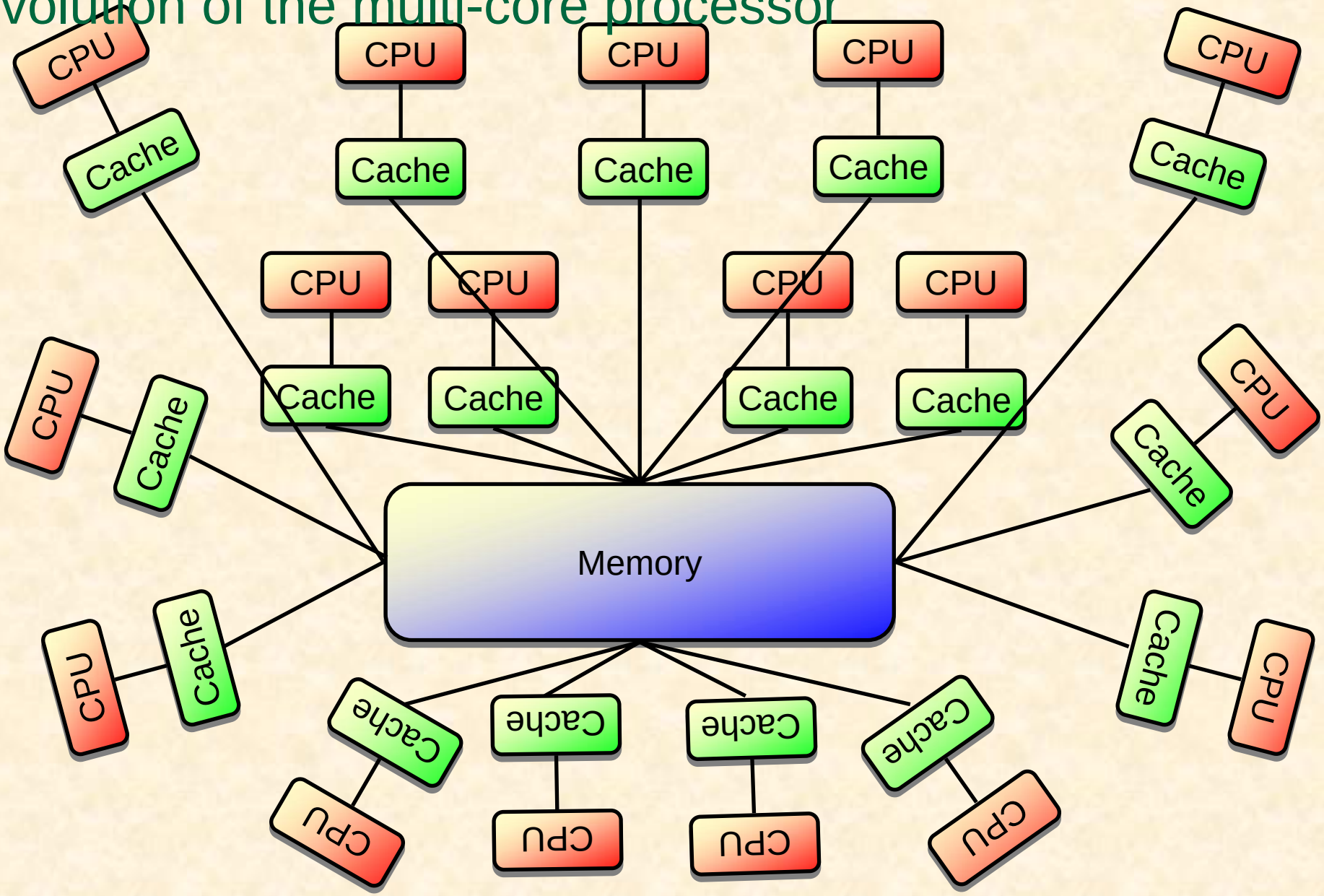
Evolution of the multi-core processor



Evolution of the multi-core processor



Evolution of the multi-core processor



Getting the most out of multi-core processors

"Expose more parallelism."

Any questions?

Getting the most out of multi-core processors

"Pass the buck."

- R Barrett

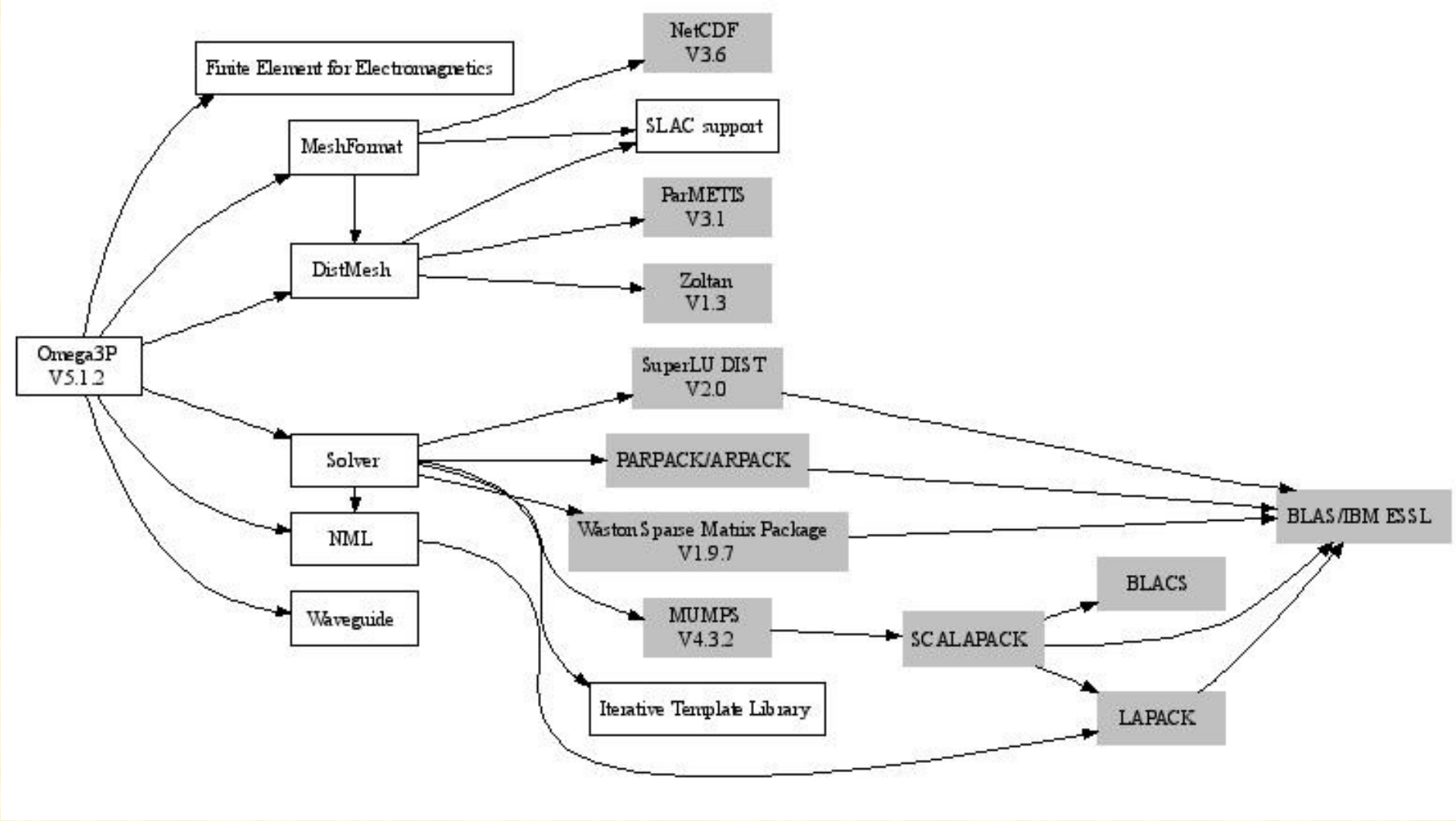
Looking forward:

- Compute Node Linux (CNL)
- Hybrid programming.
  - Intra-processor MPI within node.
- Process placement controls.

In the meantime…

- Any optimization efforts same as for any other machine.
  - Write cache-aware code, with lots of re-use.
  - Aggregate inter-process communication by a process.
  - See Mark Fahey's SC06 poster.
- http://nccs.gov/news/workshops/fy06usersmeeting/pdf/xt3-tutorial.pdf

What should your code look like?



Why should your code look like that?

- *Built in flexibility: Metis, Zoltan, etc:*
  - *Multi-level partitions: by PE, then by cores.*

- *Let someone else deal with the complexity*
  - *Multi-core aware?*
  - *Reduced precision:*
    - *ScaLAPACK, MUMPS, …*

Omega3P

Reduced precision in MUMPS

Problem Size	Number of eigen-pairs	Factorization (second)	Triangular Solver (Multiple Calls)(s)	EigenSolver (second)	Memory (GB)
614636	9	43.6	20	90.2	9.3
		40.6	117.4	185.9	4.8
1408976	11	127.5	61.9	236.8	57.3
		124.9	349.1	512.8	29.7
2364056	1	118.4	9.6	129.3	75.5
		113.4	51.5	170.6	41.2
3095880	9	244.2	37.3	293.8	155.3
		235.9	201.9	450.1	82.3
17283488	8	--	--	--	--
		1907.3	14091.9	16257.5	342

Reduced precision; regular dp.

	Math															Sys						I/O						
	BLAS	Cray Scilib	FFTPACK	FFTW	LAPACK	MUMPS	PBLAS	PEIGS	Parmetis	PARPACK	PetSC	Scalapack	SGI SCSL	SPRNG	ZOLTAN	ctime	tar	CP	Mkdir	Rm	Other	CCA-Tools	HDF5	PHDF5	MPI-IO	netCDF	pnetCDF	XML
LSMS	1				1											1							1					1
QMC	1				1									1														
CHIMERA					1																		1				1	
POP/CICE																										1		
SD3																	1	1	1	x								
GTC											1										1		1	1	1	1		1
MADNESS	1																											
AORSA			1				1					1														1		
LAMMPS				1																								
FLASH																							1				1	
Milc/Chroma																					1							
PFLOTRAN	1										1																	
QBOX	1			1	1							1																1
CAM		1											1								x					1		
CCSD	1																								1			
T3P						1			1						1						1					1		
VASP	1											1									x							
NEWTRNX					1					1												1	1					
NWChem	1		1					1				1																
OReTran	1				1																							
CASINO	1																				1							

Use tools to understand issues

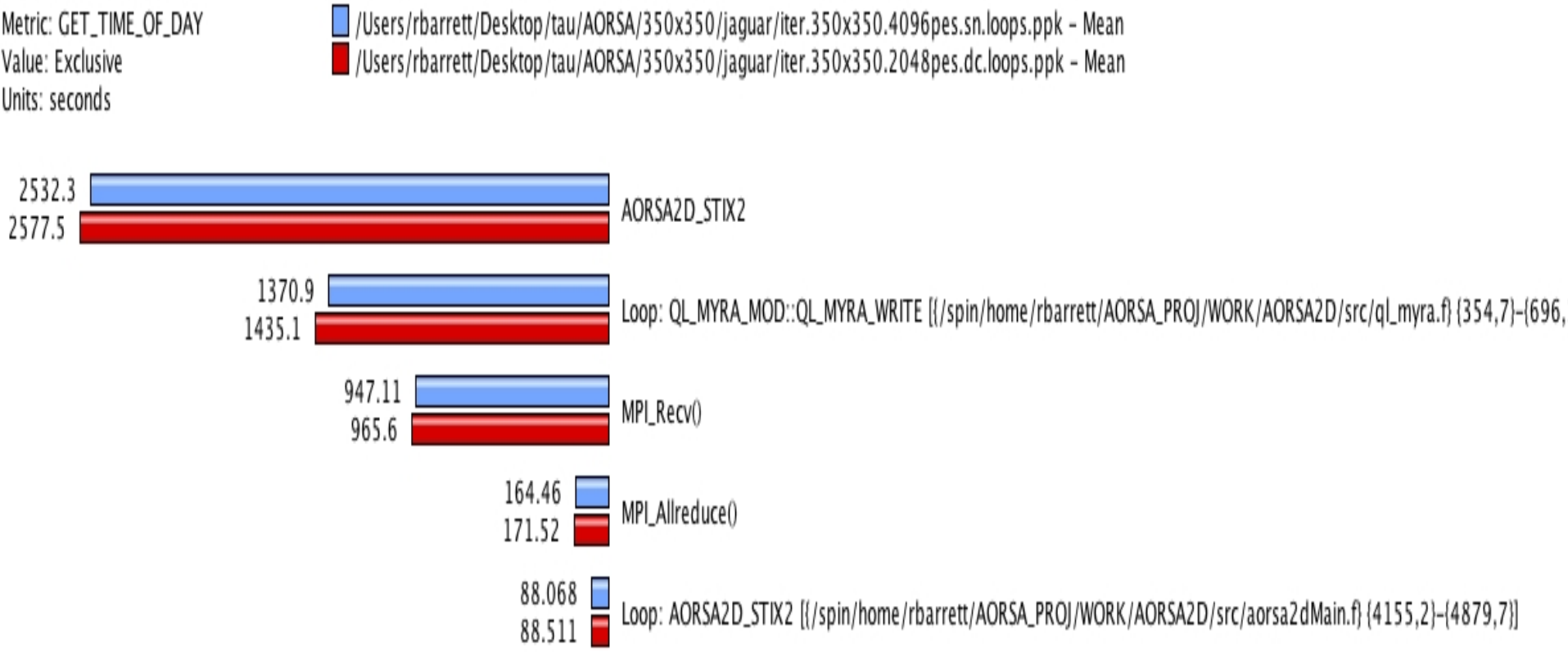
- Memory and network contention views?
  - Tools are *process* (*thread*) aware.
- Cause and effect view

Tool developers need hooks to information.

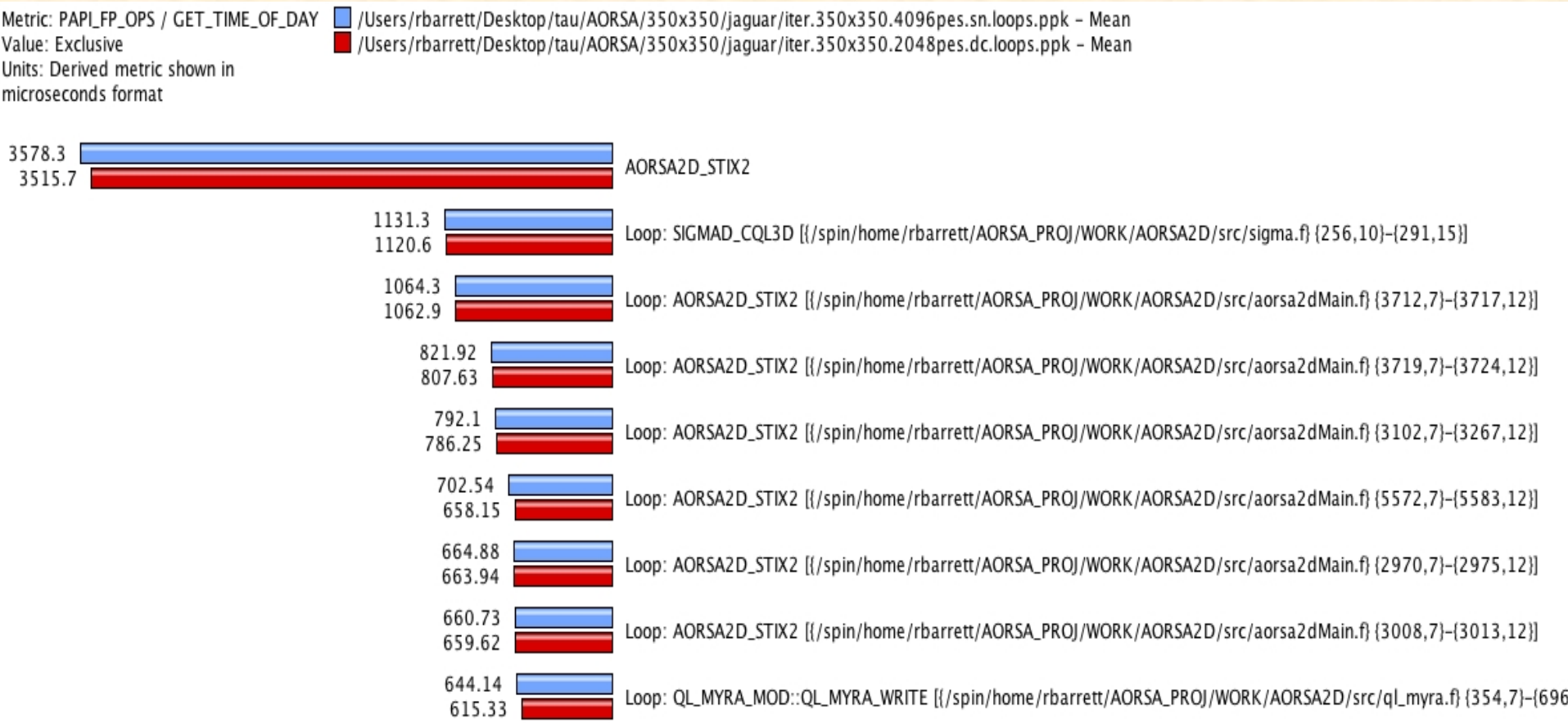
- MPI profiling layer, OPARI
- HPCS languages?

- Need OS support.

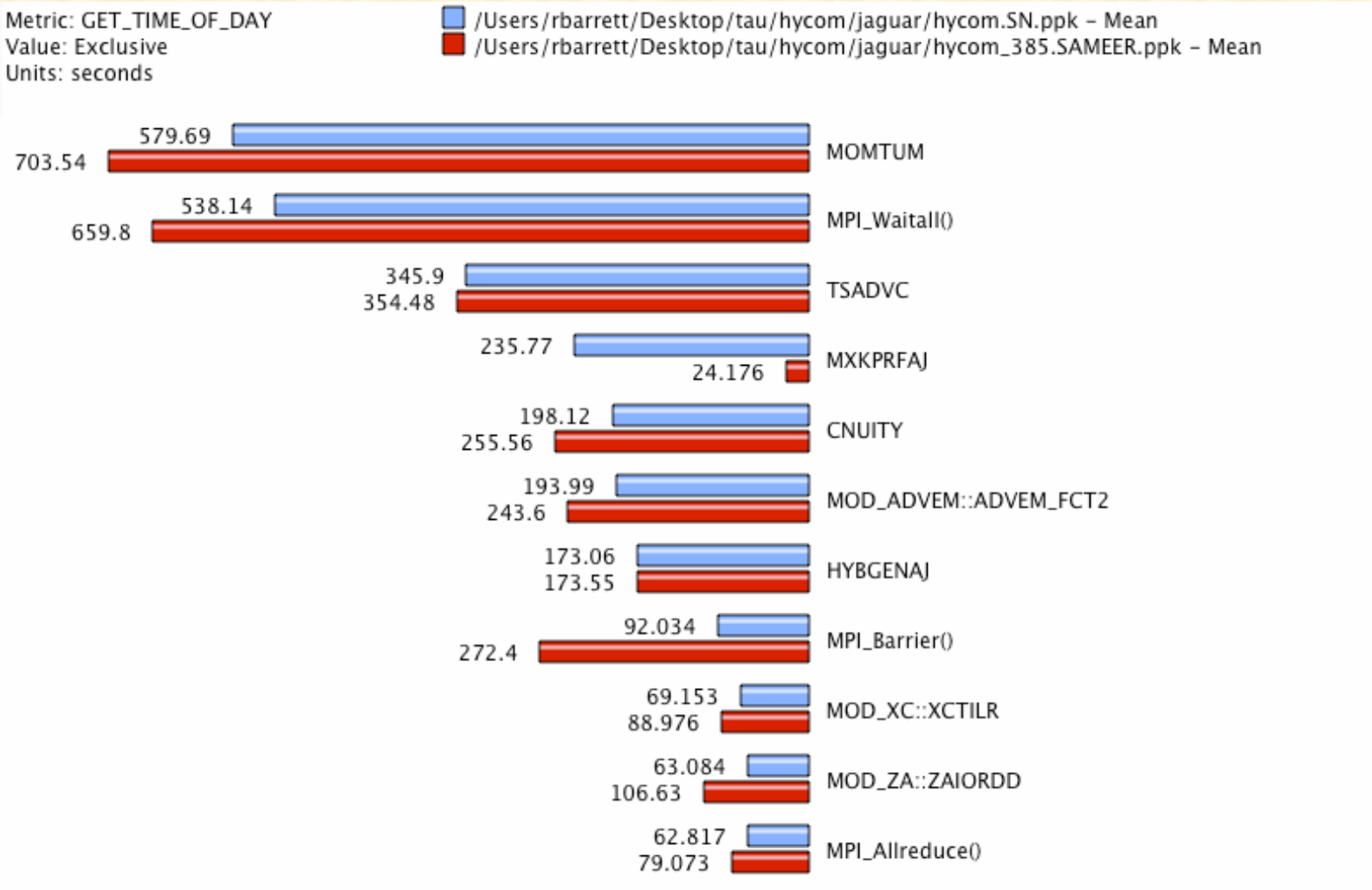
Case study: *AORSA*



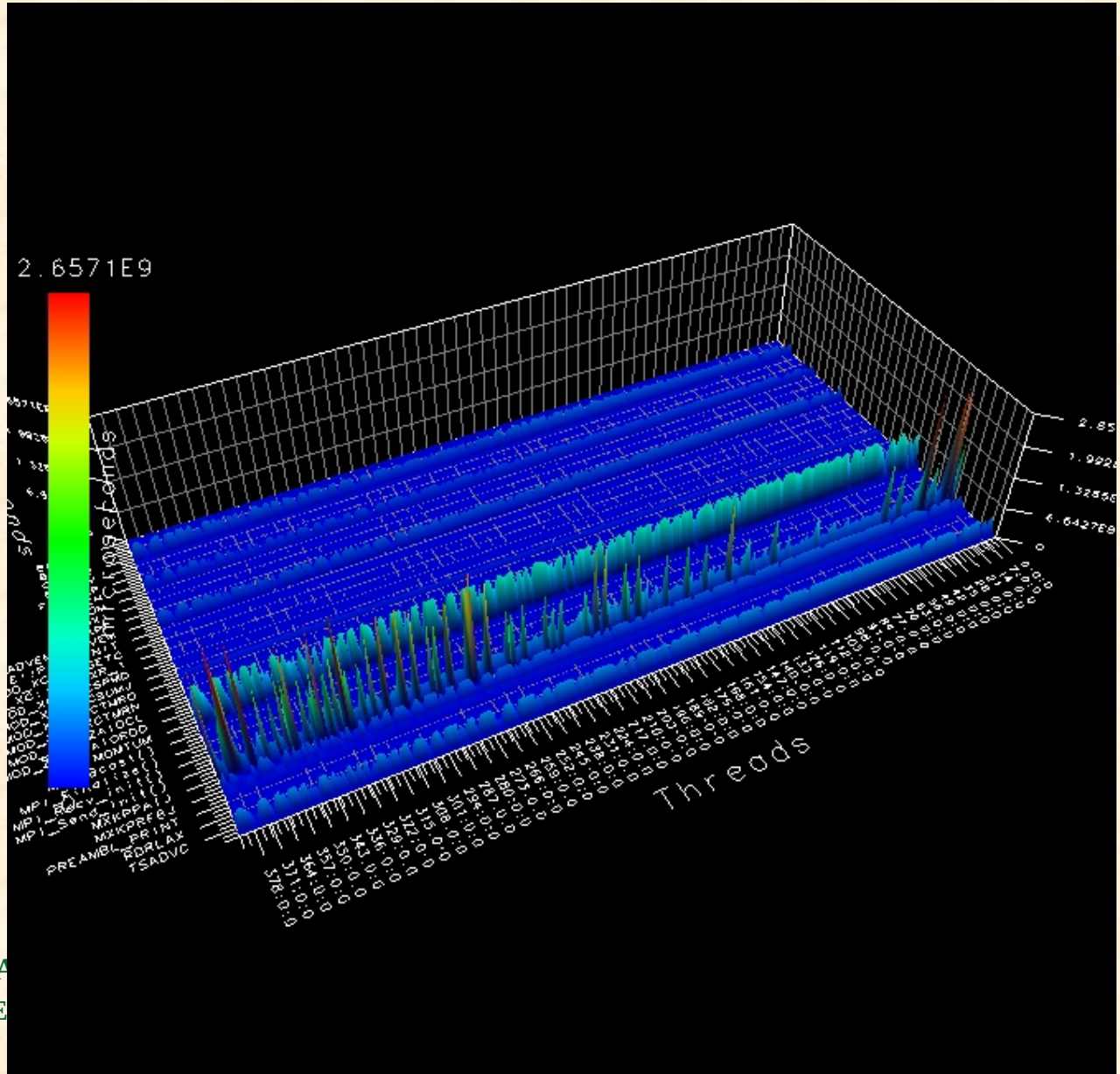
Case study: *AORSA*

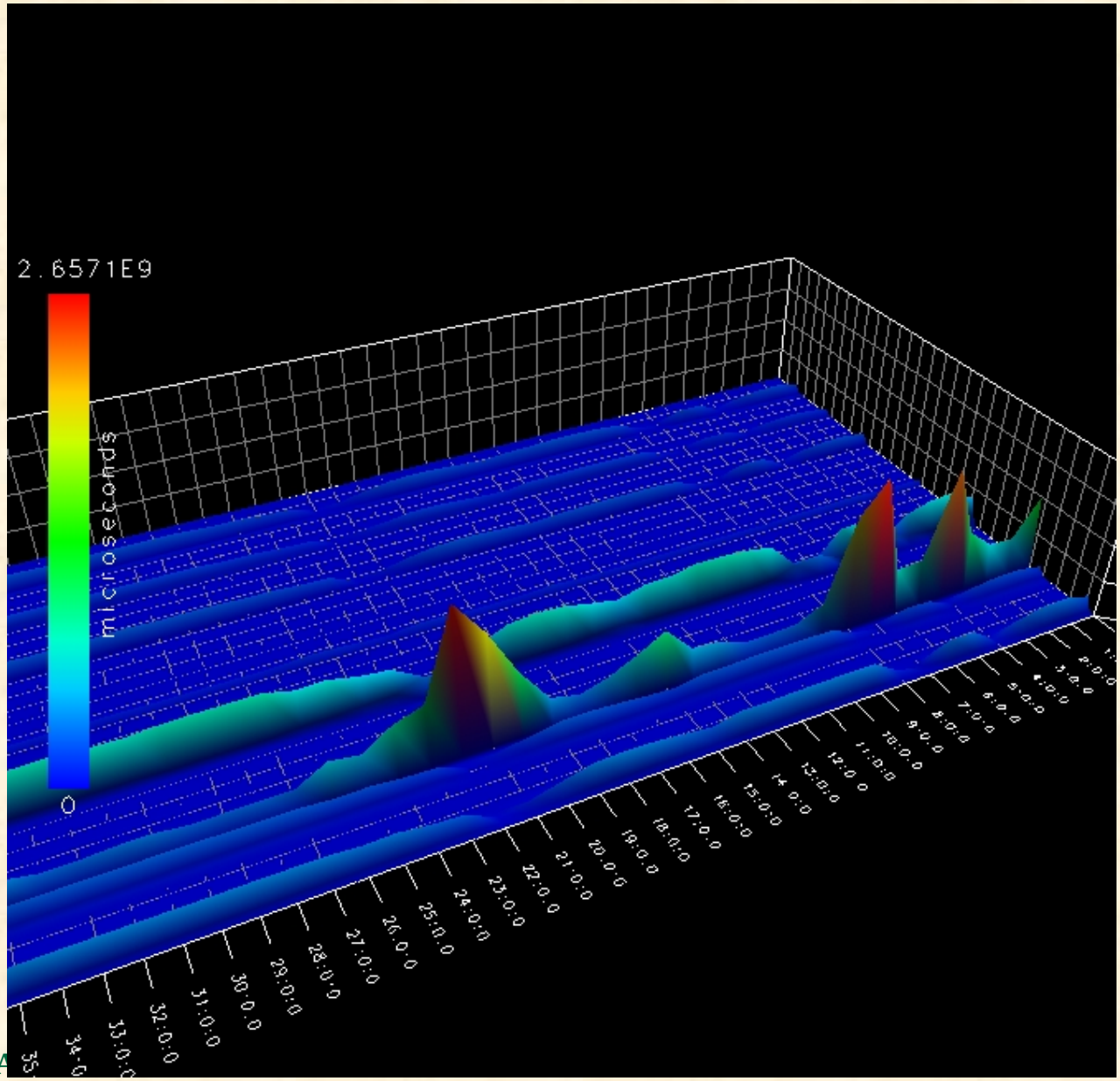


Case study: *HYCOM*



Case study: *HYCOM*




2.6571E9
microseconds
0

Sotfware

- **Languages:**
  - **UPC, CAF**
  - **HPCS**
    - **Tool developers need hooks to information.**
      - E.g. MPI profiling layer, OPARI
- **Message passing pre-processing:**
  - Anthony Danalis, Ki-Yong Kim, Lori Pollock, and Martin Swany, *"Transformations to Parallel Codes for Communication-Computation Overlap"*, Proceedings of IEEE/ACM Conference on High Performance Computing, Networking, Storage and Analysis (SC05), November, 2005.
- **Compiler responsibilities?**
  - Scheduling memory access, inter-process communication
  - Hyperthreading

What do we need for multi-core?

- Software with single processor performance.
- Libraries with single processor performance.
- Tools that scale.
- Tools that show us what's happening, and effects of our efforts.

Conclusion

"You can optimize some of the codes all of the time, and all of the codes some of the time, but you can not optimize all of the codes all of the time."

Requires understanding app goals, algorithms, code, architecture, languages, libraries, compilers... try things.