Georg Hager Edit this form

29 responses

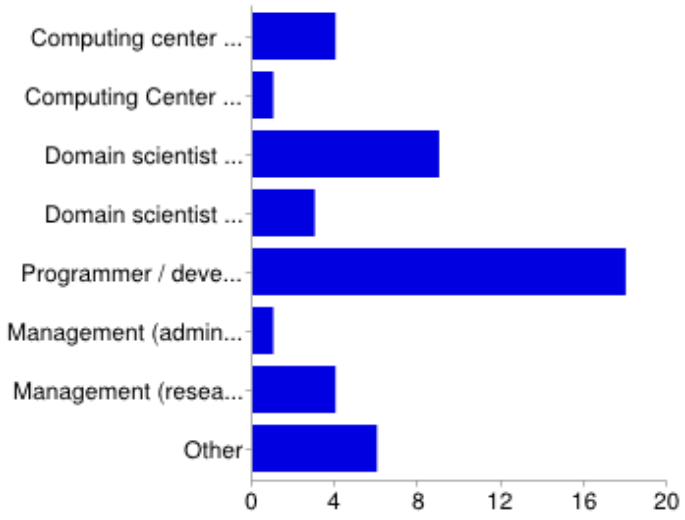
View all responses Publish analytics

Summary

Fill out online at at http://goo.gl/forms/hiXM5Feu3B

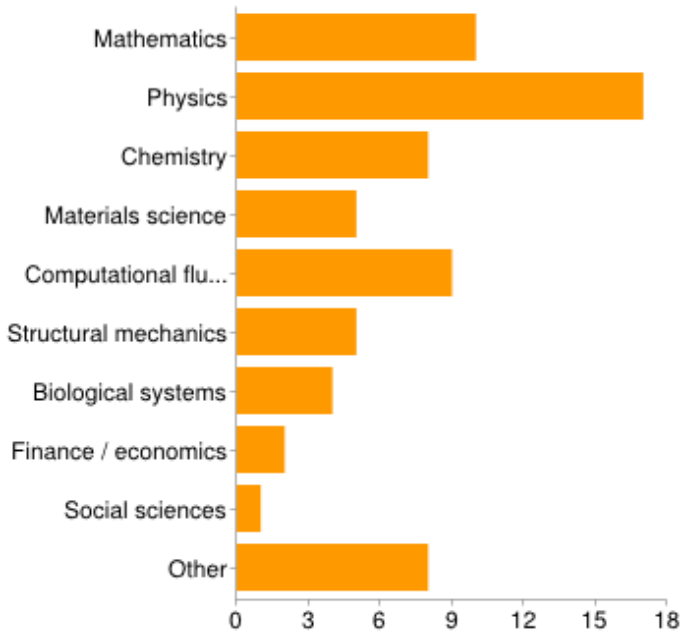
[Image]

Which area are you working in?



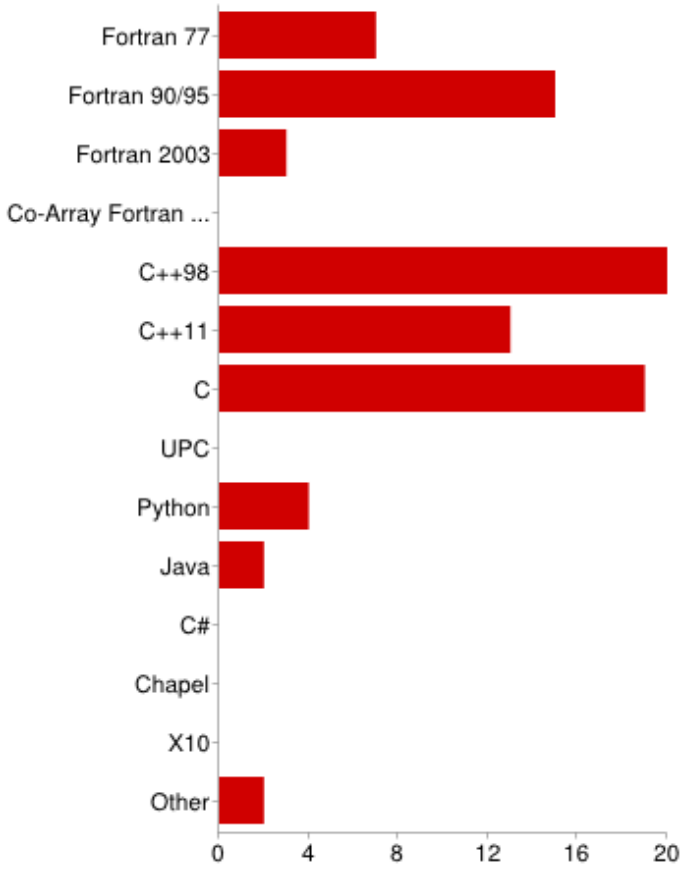
Computing center staff (consulting)	**4**	14%
Computing Center staff (sysadmin)	**1**	3%
Domain scientist (developer)	**9**	31%
Domain scientist (user)	**3**	10%
Programmer / developer	**18**	62%
Management (administrative)	**1**	3%
Management (research)	**4**	14%
Other	**6**	21%

Which areas of science do the codes come from that you are working on?



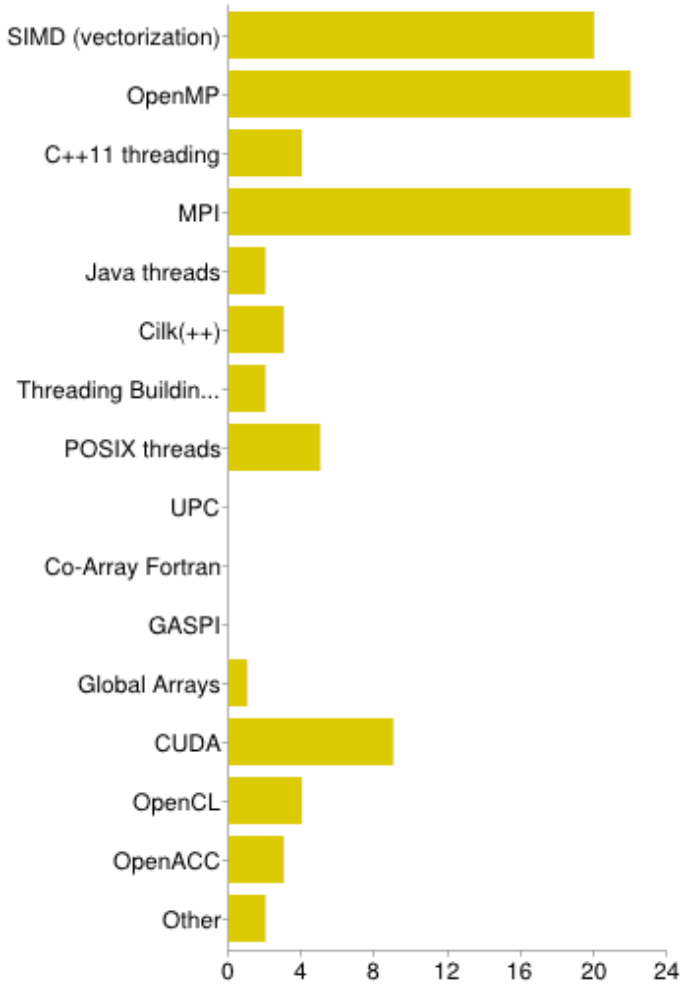
Mathematics	**10**	34%
Physics	**17**	59%
Chemistry	**8**	28%
Materials science	**5**	17%
Computational fluid dynamics	**9**	31%
Structural mechanics	**5**	17%
Biological systems	**4**	14%
Finance / economics	**2**	7%
Social sciences	**1**	3%
Other	**8**	28%

What is the programing language that you mostly deal with when optimizing code?



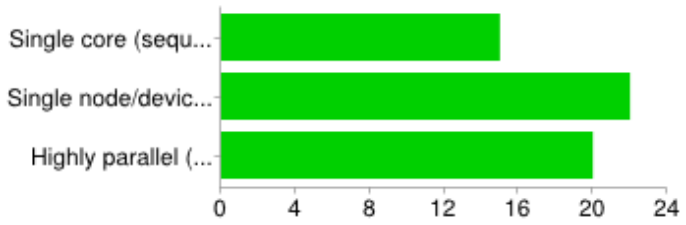
Fortran 77	**7**	24%
Fortran 90/95	**15**	52%
Fortran 2003	**3**	10%
Co-Array Fortran / Fortran 2008	**0**	0%
C++98	**20**	69%
C++11	**13**	45%
C	**19**	66%
UPC	**0**	0%
Python	**4**	14%
Java	**2**	7%
C#	**0**	0%
Chapel	**0**	0%
X10	**0**	0%
Other	**2**	7%

What is the parallel programming model that you mostly deal with when optimizing code?



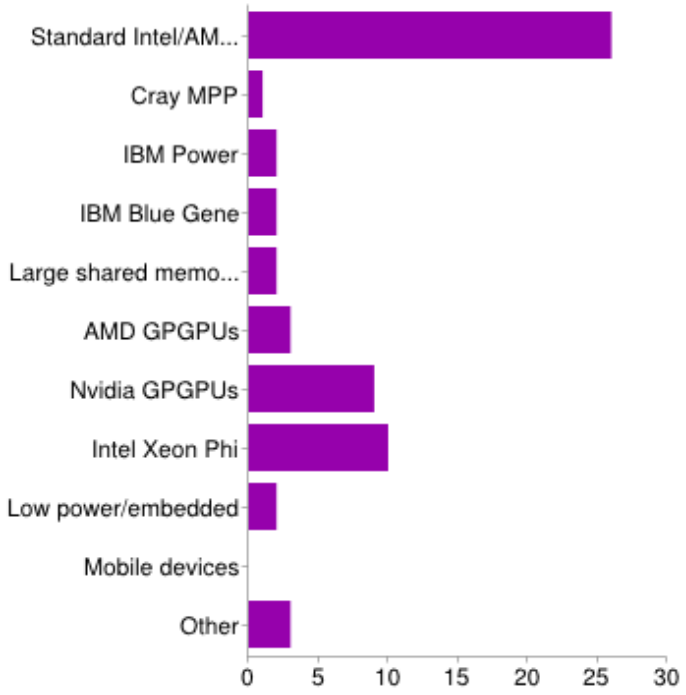
SIMD (vectorization)	**20**	69%
OpenMP	**22**	76%
C++11 threading	**4**	14%
MPI	**22**	76%
Java threads	**2**	7%
Cilk(++)	**3**	10%
Threading Building Blocks (TBB)	**2**	7%
POSIX threads	**5**	17%
UPC	**0**	0%
Co-Array Fortran	**0**	0%
GASPI	**0**	0%
Global Arrays	**1**	3%
CUDA	**9**	31%
OpenCL	**4**	14%
OpenACC	**3**	10%
Other	**2**	7%

Are your code optimizations centered on the single core, node, or highly parallel level?



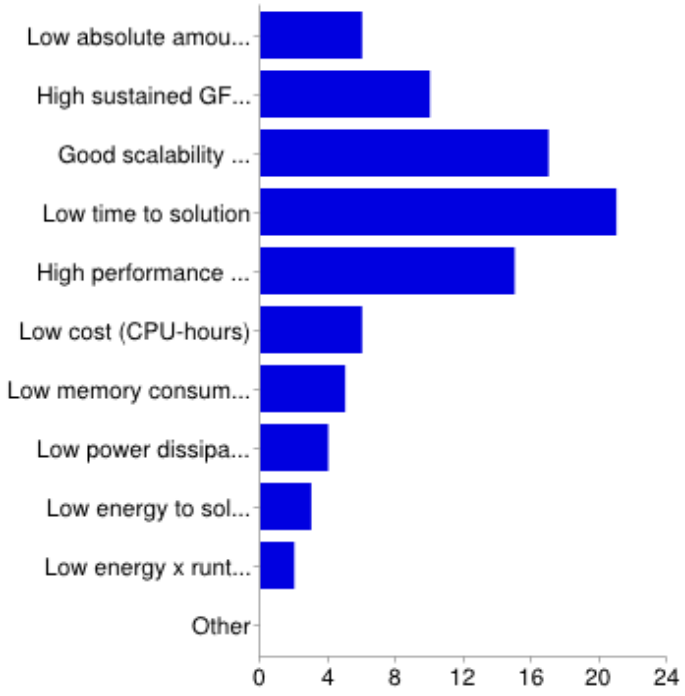
Single core (sequential)	**15**	52%
Single node/device (e.g., CPU socket, multiple CPU sockets, GPGPU, other accelerator)	**22**	76%
Highly parallel (distributed memory)	**20**	69%

What is/are the target architecture(s) for which you optimize code?



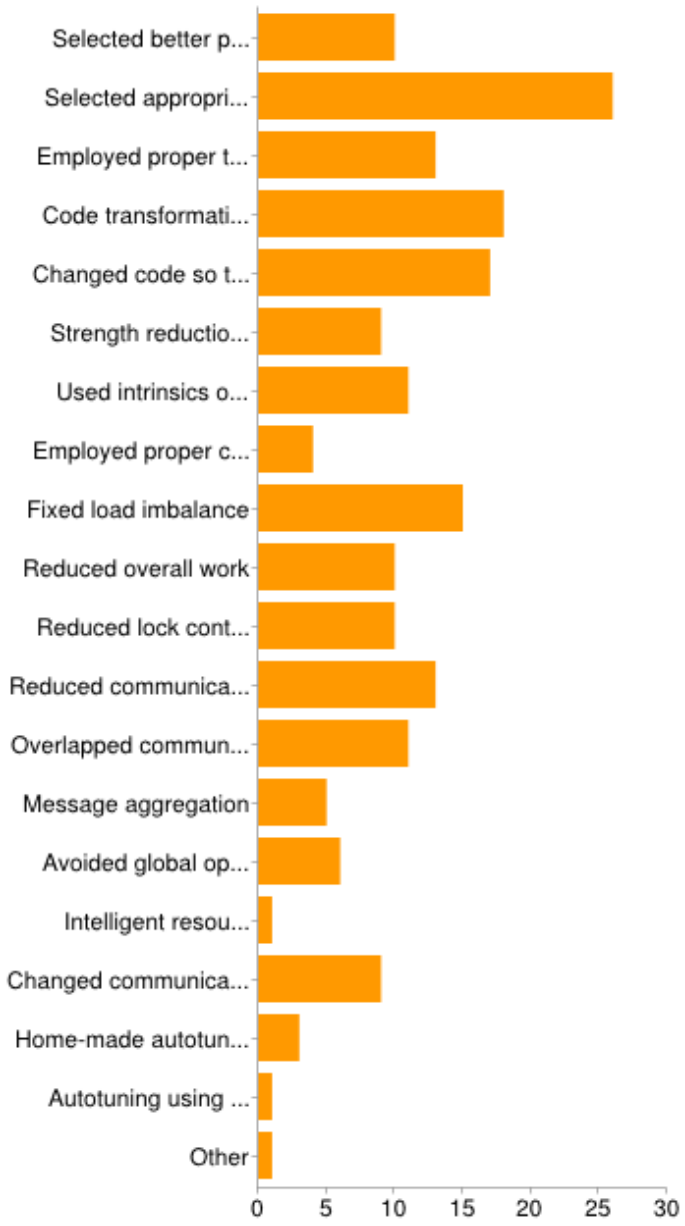
Standard Intel/AMD x86 chips/clusters	**26**	90%
Cray MPP	**1**	3%
IBM Power	**2**	7%
IBM Blue Gene	**2**	7%
Large shared memory systems	**2**	7%
AMD GPGPUs	**3**	10%
Nvidia GPGPUs	**9**	31%
Intel Xeon Phi	**10**	34%
Low power/embedded	**2**	7%
Mobile devices	**0**	0%
Other	**3**	10%

What are the typical target metrics you optimize for?



Low absolute amount of work	**6**	21%
High sustained GFlop/s rate	**10**	34%
Good scalability (large speedup)	**17**	59%
Low time to solution	**21**	72%
High performance (work/time)	**15**	52%
Low cost (CPU-hours)	**6**	21%
Low memory consumption	**5**	17%
Low power dissipation (i.e., low sustained Watts)	**4**	14%
Low energy to solution	**3**	10%
Low energy x runtime (or a variant thereof)	**2**	7%
Other	**0**	0%

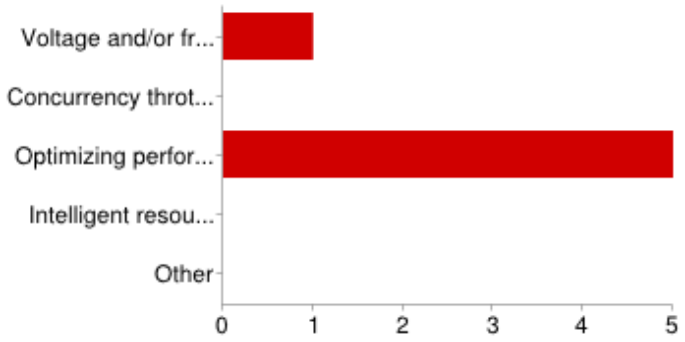
If you have ever optimized code performance, what approach(es) did you use so far to improve it?



Selected better programming language / programming model / library	**10**	34%
Selected appropriate compiler options	**26**	90%
Employed proper thread/process affinity	**13**	45%
Code transformations: blocking, unrolling, loop fusion, etc.	**18**	62%
Changed code so the compiler does a better job	**17**	59%
Strength reduction (avoid expensive operations)	**9**	31%
Used intrinsics or assembly language	**11**	38%
Employed proper ccNUMA page placement	**4**	14%
Fixed load imbalance	**15**	52%
Reduced overall work	**10**	34%
Reduced lock contention/serialization	**10**	34%
Reduced communication volume/frequency	**13**	45%
Overlapped communication with computation	**11**	38%
Message aggregation	**5**	17%
Avoided global operations	**6**	21%
Intelligent resource management (e.g., overlapping codes with different requirements to the hardware)	**1**	3%
Changed communication pattern	**9**	31%
Home-made autotuning (e.g., using scripts)	**3**	10%

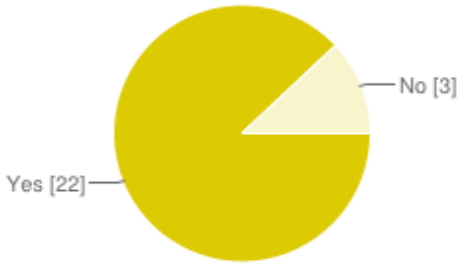
Autotuning using an available tool	**1**	3%
Other	**1**	3%

If you have ever optimized for energy-related metrics, what approach(es) did you use so far?



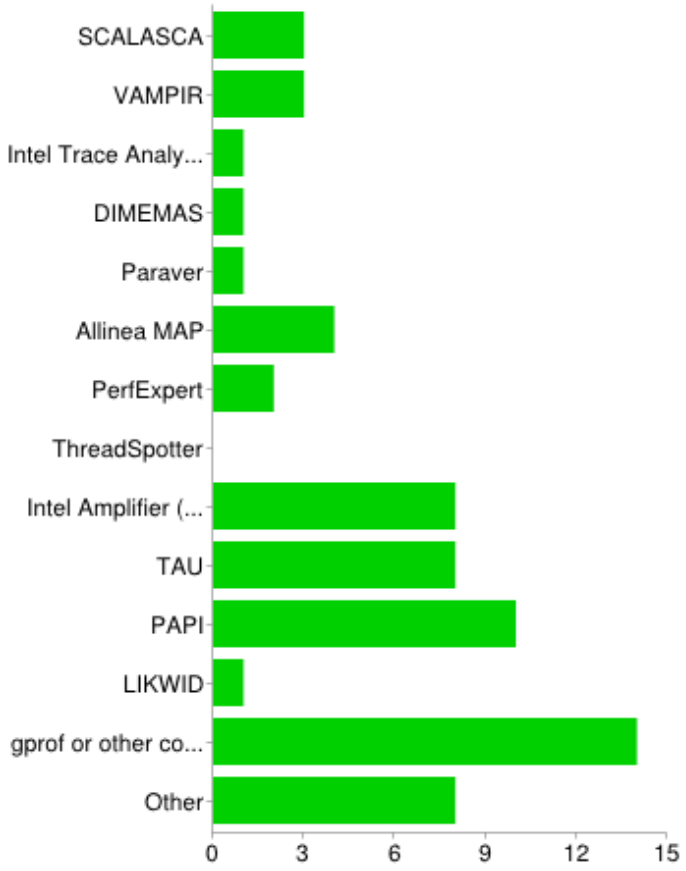
Voltage and/or frequency scaling	**1**	3%
Concurrency throttling (using fewer cores than are available)	**0**	0%
Optimizing performance	**5**	17%
Intelligent resource management (e.g., co-scheduling of different workloads)	**0**	0%
Other	**0**	0%

Have you ever used performance tools for optimizing applicaton code?



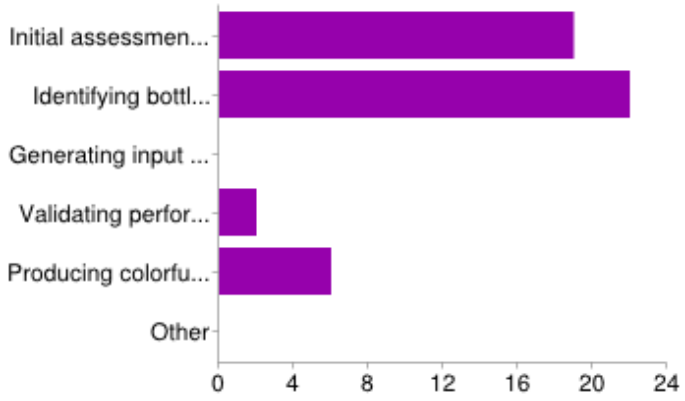
Yes	**22**	76%
No	**3**	10%

If you have used performance tools, which ones?



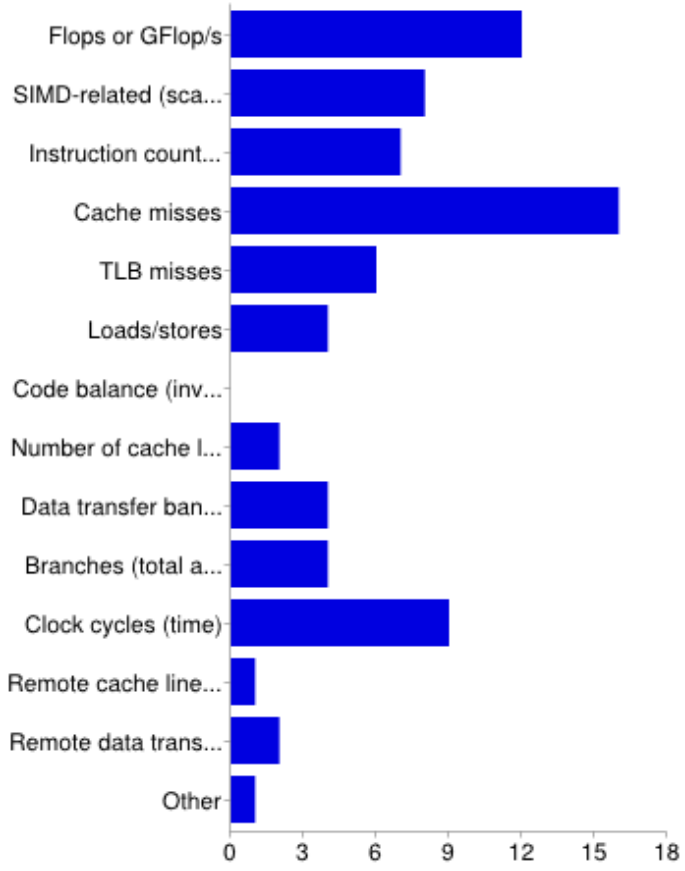
SCALASCA	**3**	10%
VAMPIR	**3**	10%
Intel Trace Analyzer/Collector	**1**	3%
DIMEMAS	**1**	3%
Paraver	**1**	3%
Allinea MAP	**4**	14%
PerfExpert	**2**	7%
ThreadSpotter	**0**	0%
Intel Amplifier (VTune)	**8**	28%
TAU	**8**	28%
PAPI	**10**	34%
LIKWID	**1**	3%
gprof or other compiler-based tool	**14**	48%
Other	**8**	28%

If you use tools, what do you use them for?



Initial assessment/profiling	**19**	66%
Identifying bottlenecks	**22**	76%
Generating input for statistical modeling	**0**	0%
Validating performance models	**2**	7%
Producing colorful graphs	**6**	21%
Other	**0**	0%

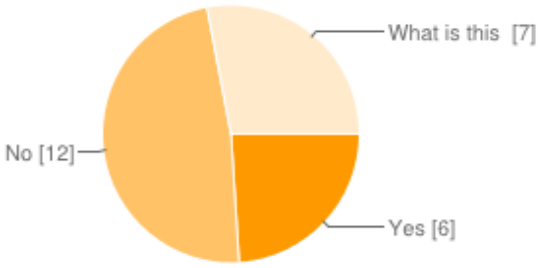
If you use hardware performance counter measurements, what events/metrics do you look at?



Flops or GFlop/s	**12**	41%
SIMD-related (scalar vs. packed instructions)	**8**	28%
Instruction count / IPC	**7**	24%
Cache misses	**16**	55%
TLB misses	**6**	21%

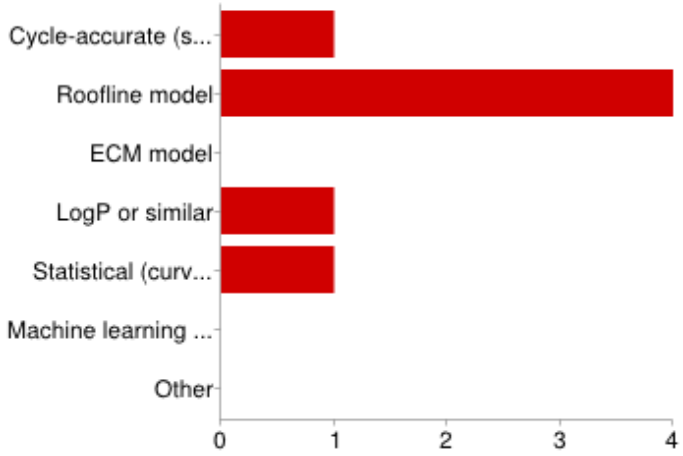
Loads/stores	**4**	14%
Code balance (inverse intensity)	**0**	0%
Number of cache lines (data volume)	**2**	7%
Data transfer bandwidths	**4**	14%
Branches (total and mispredicted)	**4**	14%
Clock cycles (time)	**9**	31%
Remote cache line evicts	**1**	3%
Remote data transfers (cross-NUMA domains)	**2**	7%
Other	**1**	3%

Do you use performance models in your optimization efforts?



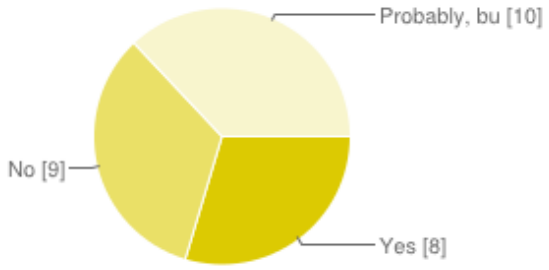
Yes	**6**	21%
No	**12**	41%
What is this "performance model" stuff anyway?	**7**	24%

If you use performance models, which ones?



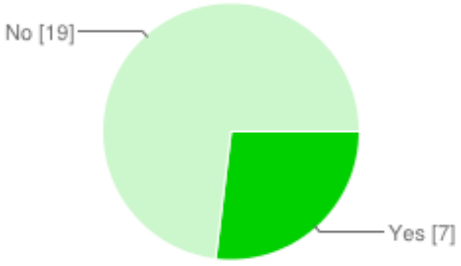
Cycle-accurate (simulation)	**1**	3%
Roofline model	**4**	14%
ECM model	**0**	0%
LogP or similar	**1**	3%
Statistical (curve fitting, extrapolation)	**1**	3%
Machine learning based	**0**	0%
Other	**0**	0%

Do you think you would have interesting optimization case studies to share at a dedicated workshop?



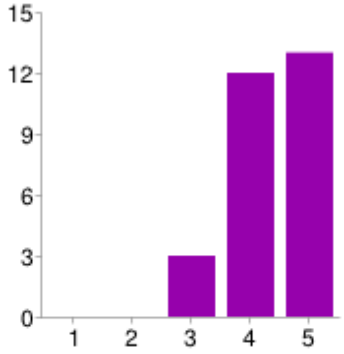
Yes	**8**	28%
No	**9**	31%
Probably, but would require substantial effort	**10**	34%

Have you ever had a paper rejected because it was "just" an optimization case study with "no novelty"?



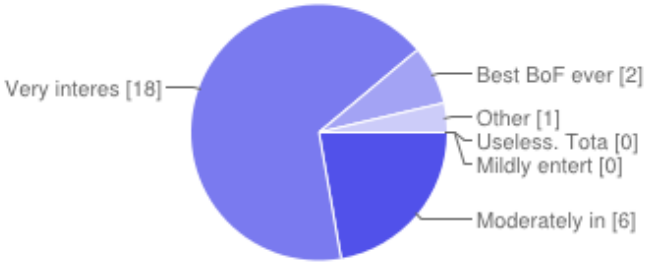
Yes	**7**	24%
No	**19**	66%

How would you rate the usefulness of an "Optimization community" for your daily work?



1	**0**	0%
2	**0**	0%
3	**3**	10%
4	**12**	41%
5	**13**	45%

Your overall opinion about this BoF?



Useless. Total waste of time.	**0**	0%
Mildly entertaining.	**0**	0%
Moderately interesting, learned something new.	**6**	21%
Very interesting. More of this!	**18**	62%
Best BoF ever!	**2**	7%
Other	**1**	3%

Anything you would like to share? Suggestions, comments about this BoF?

I really like the research that comes out of he RRZE et al. group. One small criticism of the ECM model: the assumption that there is no overlap of traffic between the levels of the cache always gives me pause. I wish you could get an official onfirmation/refutation of this from Intel.

Congratulations, and thanks for the amazing BoF

Enjoyed the session, but it wasn't really a BoF. Instead it was just a series of short presentations. If billing as a BoF again next year, should incorporate more time for discussion and audience participation.

needs more complex examples

a tidbit of more general optimization strategies for various types of code.

Would like to see more low-level optimization techniques that are closer to the metal

Liked the fact it wasn't all about MPI

Number of daily responses