Performance Tuning:
How to Prepare to be Surprised, and then…
How to Hunt for the Surprises

Presented to
ATPESC 2017 Participants

James Reinders
HPC Consultant, Retired from Intel in 2016 (after 10,001 days)

Q Center, St. Charles, IL (USA)
Date 08/08/2017

EXASCALE COMPUTING PROJECT

National Nuclear Security Administration

U.S. DEPARTMENT OF ENERGY | Office of Science

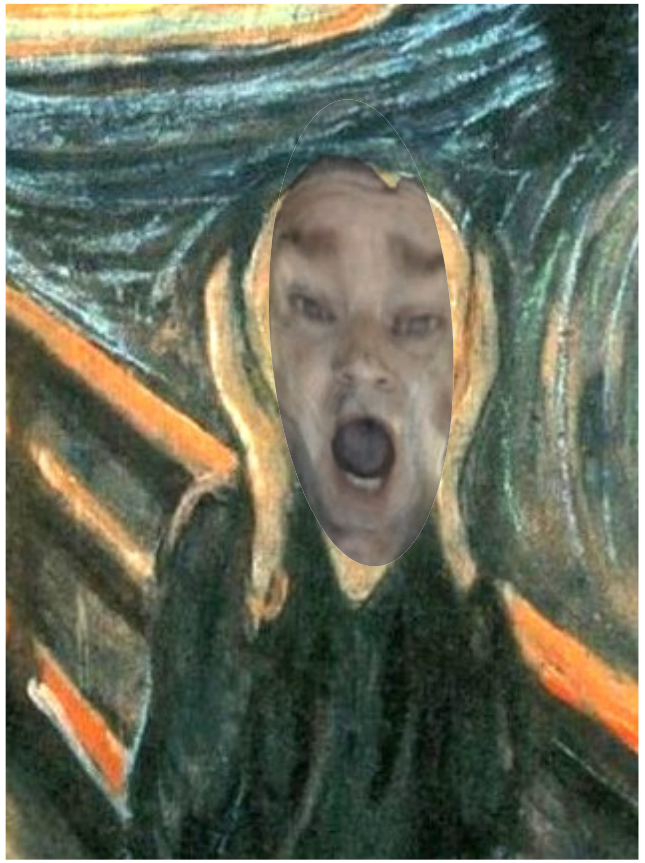
Argonne NATIONAL LABORATORY

Performance Tuning: How to Prepare to be Surprised, and then… How to Hunt for the Surprises

APPEARING

SOON

SUMMARY OF MY ENTIRE TALK



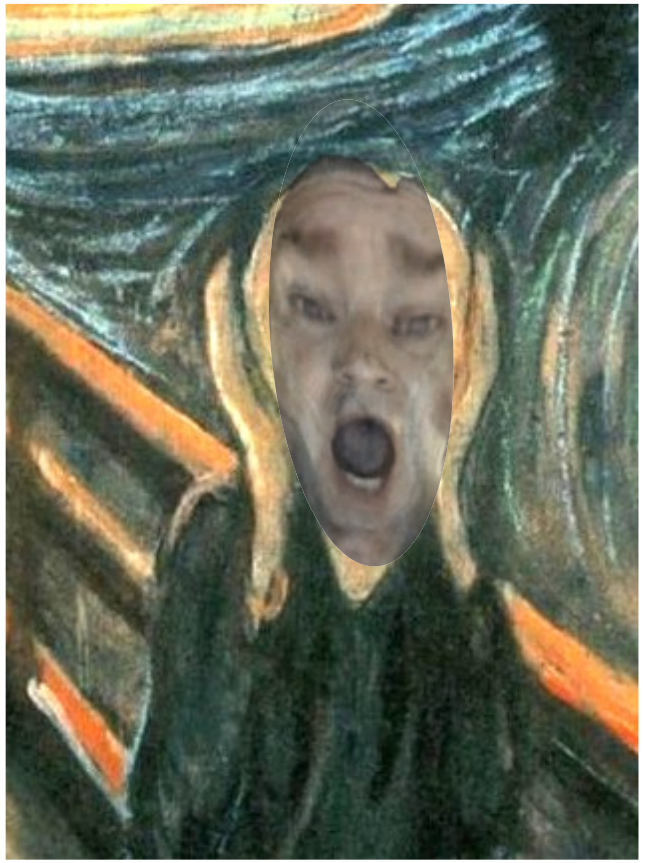
Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Performance Tuning: How to Prepare to be Surprised, and then… How to Hunt for the Surprises

- **Have Expectations**
  - **make sure you *can* be surprised**
- **Validate your expectations**
  - **follow-up on discrepancies**
- **Follow a methodology**
  - **Use tools but don't let them wag you**

SUMMARY OF MY ENTIRE TALK



Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Performance Tuning: How to Prepare to be Surprised, and then… How to Hunt for the Surprises

- Have Expectations
  - make sure you *can* be surprised
- **Validate your expectations**
  - **follow-up on discrepancies**
- Follow a methodology
  - Use tools but don't let them wag you

SUMMARY OF MY ENTIRE TALK

Доверяй, но проверяй

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Performance Tuning: How to Prepare to be Surprised, and then… How to Hunt for the Surprises

- Have Expectations
  - make sure you *can* be surprised
- Validate your expectations
  - follow-up on discrepancies
- **Follow a methodology**
  - **Use tools but don't let them wag you**

SUMMARY OF MY ENTIRE TALK

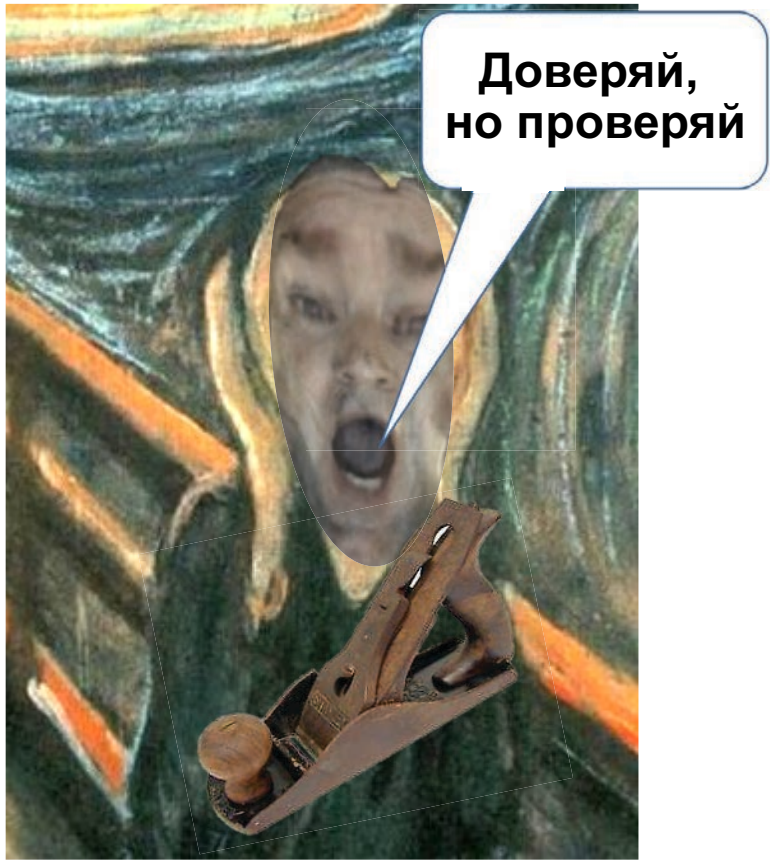
Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Performance Tuning: How to Prepare to be Surprised, and then… How to Hunt for the Surprises

- **Have Expectations**
  - make sure you *can* be surprised
- **Validate your expectations**
  - follow-up on discrepancies
- **Follow a methodology**
  - Use tools but don't let them wag you

SUMMARY OF MY ENTIRE TALK



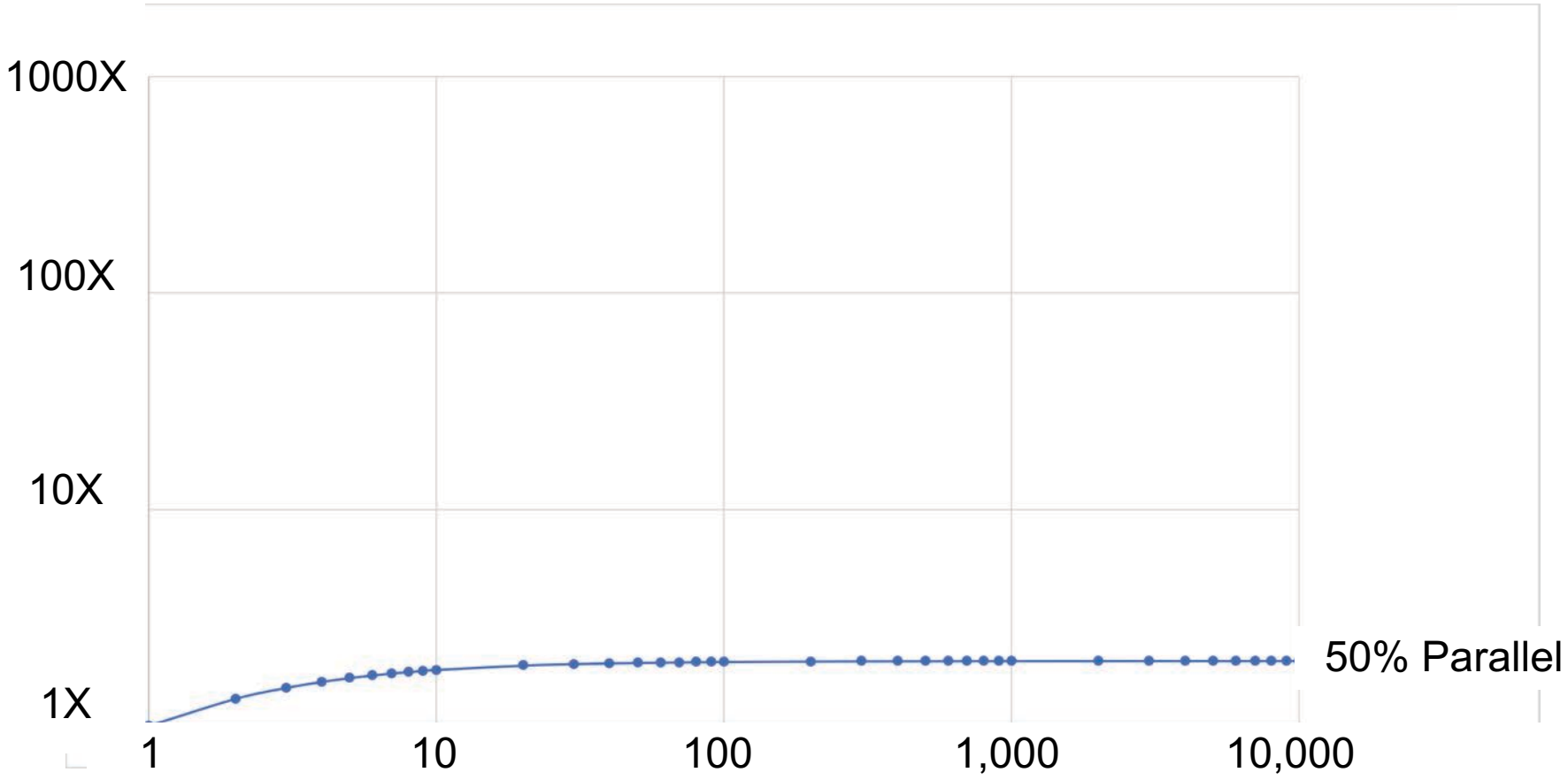
Argonne NATIONAL LABORATORY

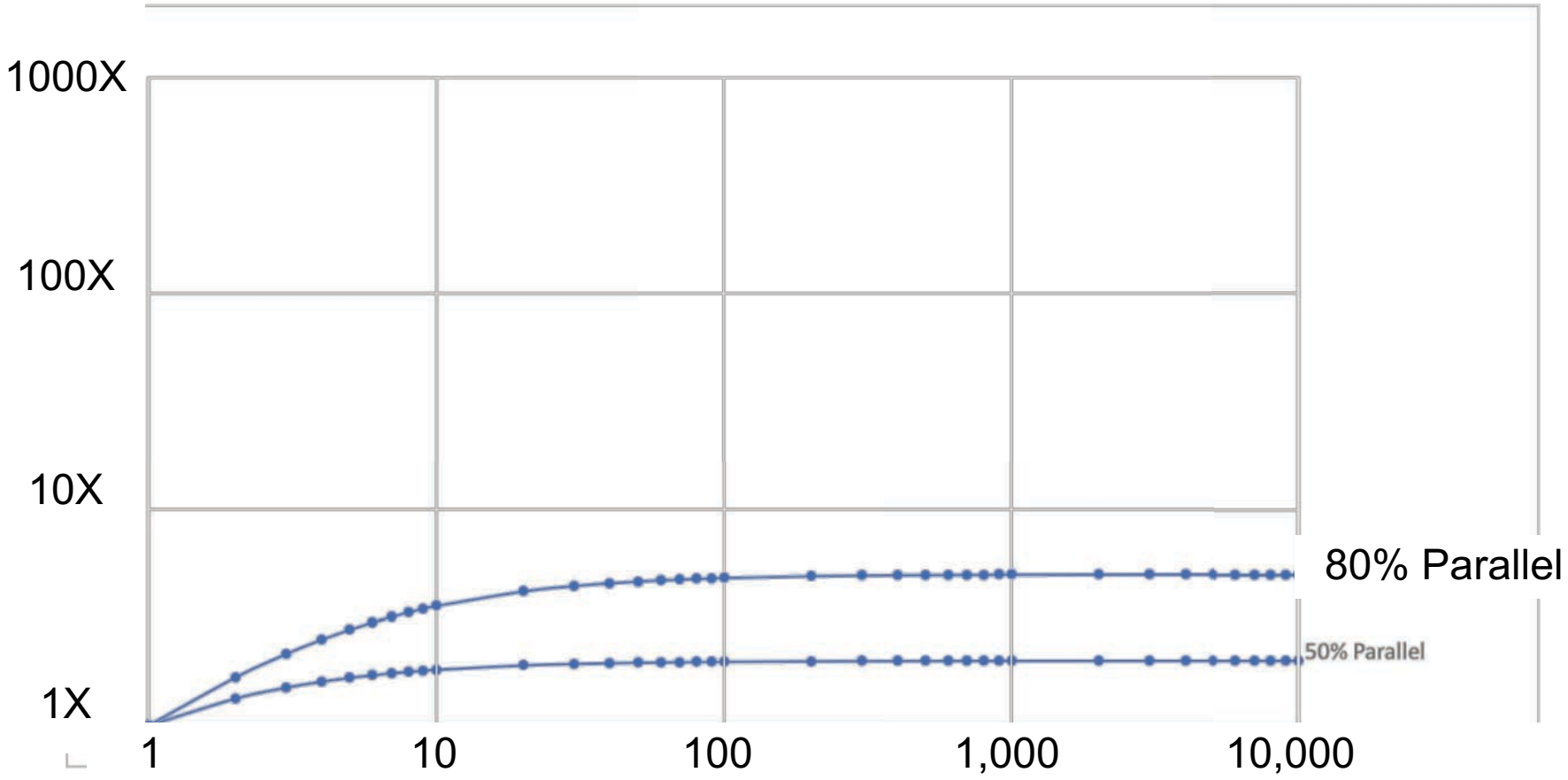
ECP EXASCALE COMPUTING PROJECT

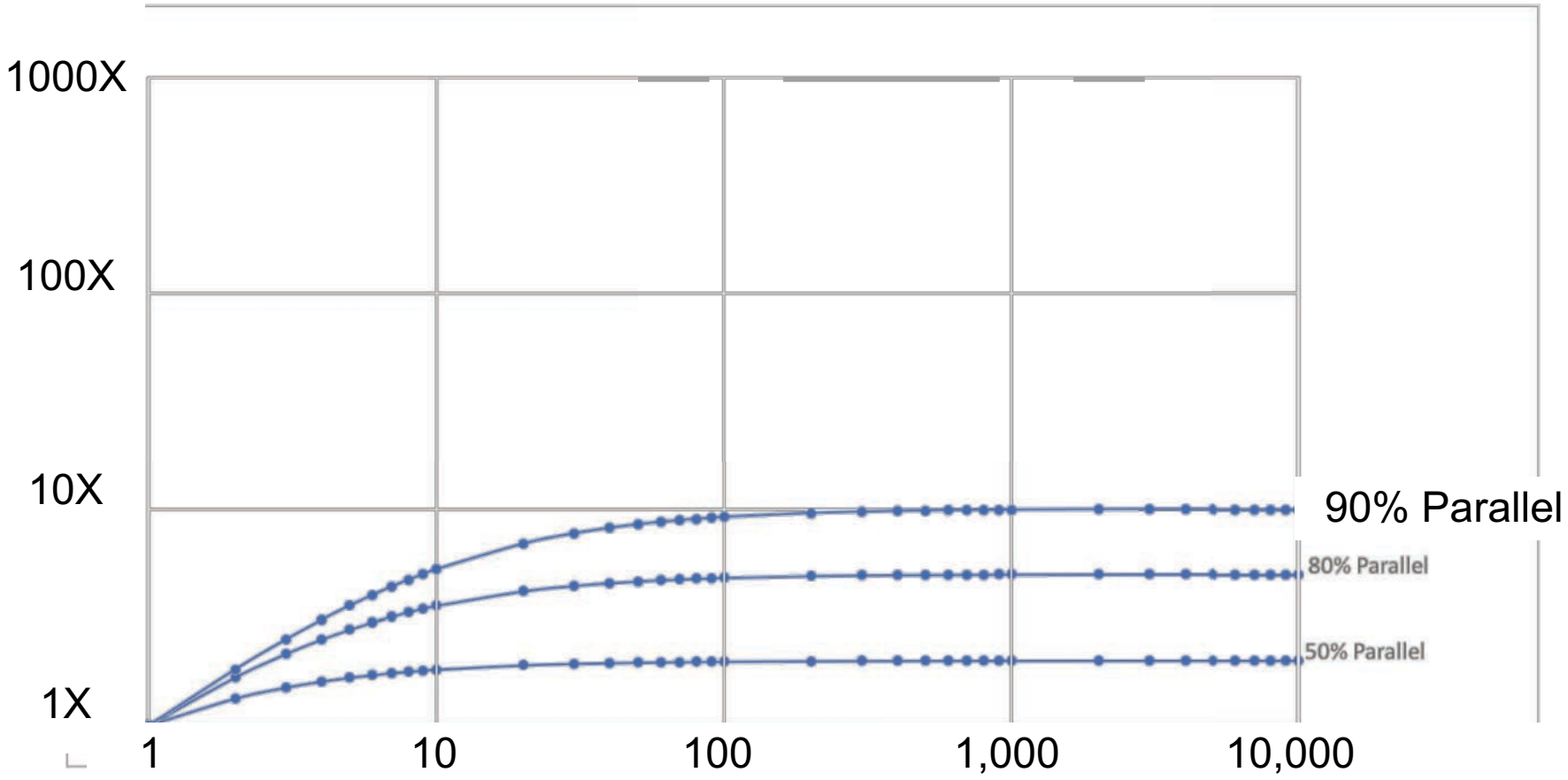
What is a *cache*?

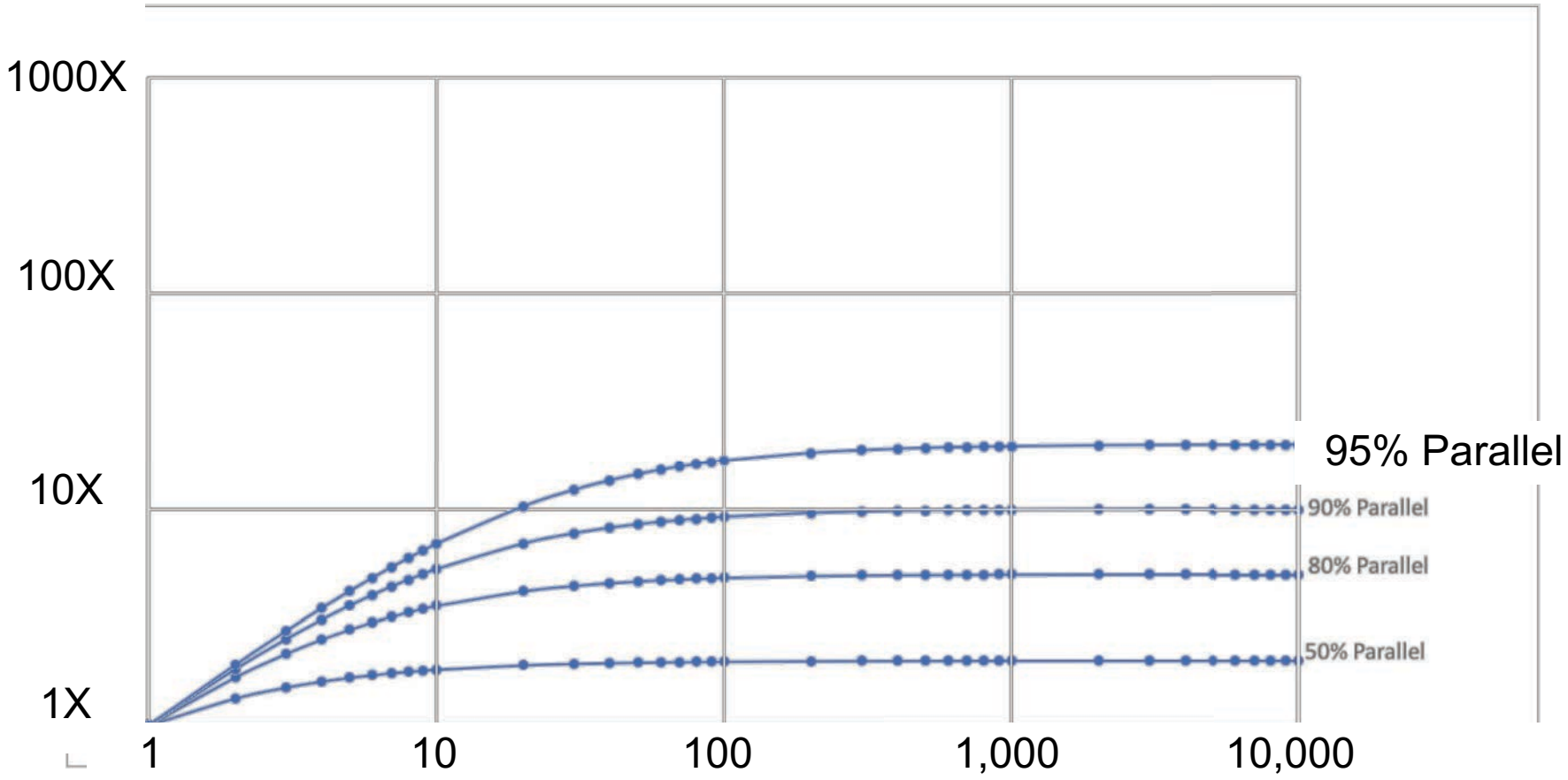
Argonne NATIONAL LABORATORY

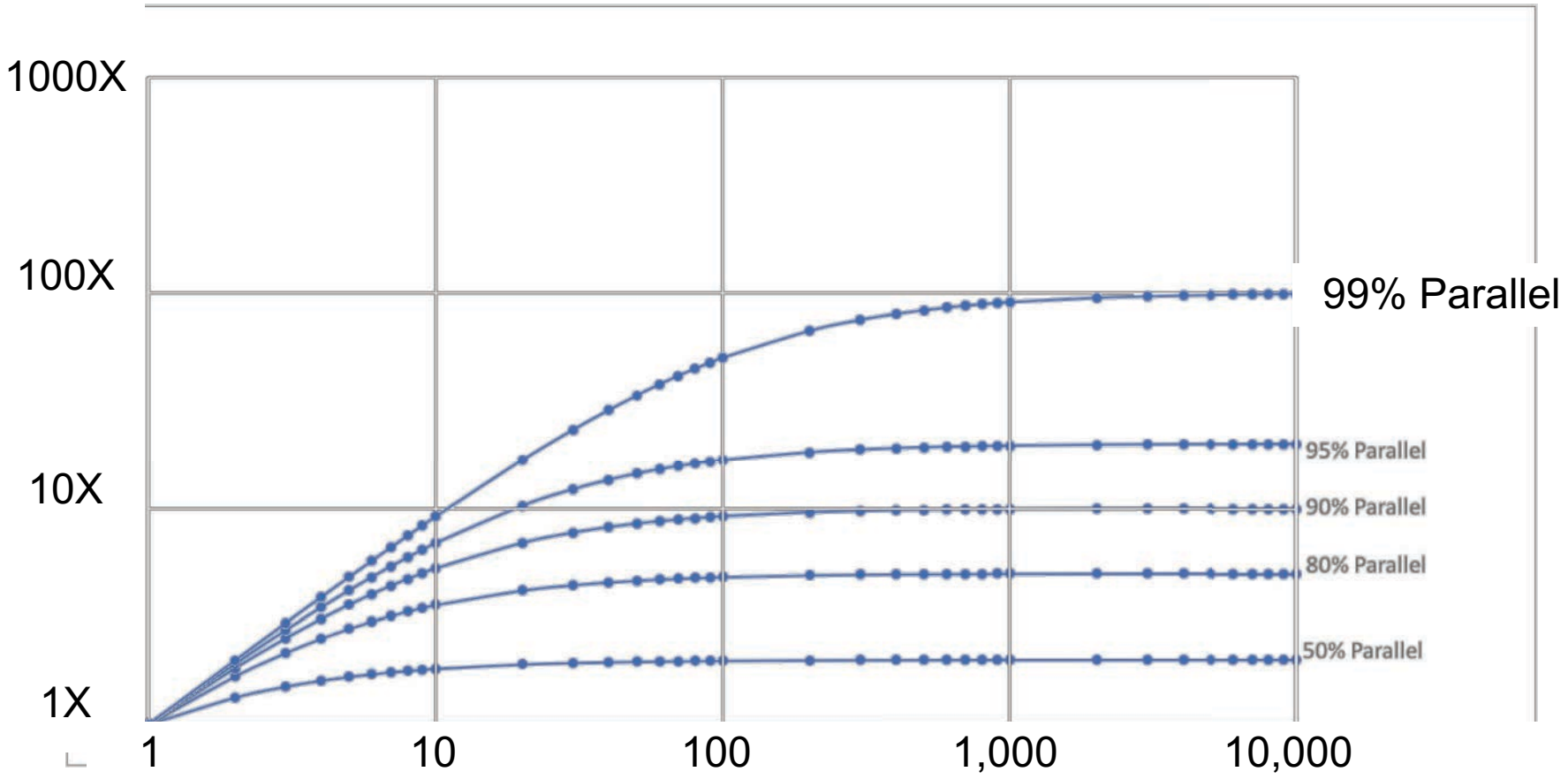
ECP EXASCALE COMPUTING PROJECT

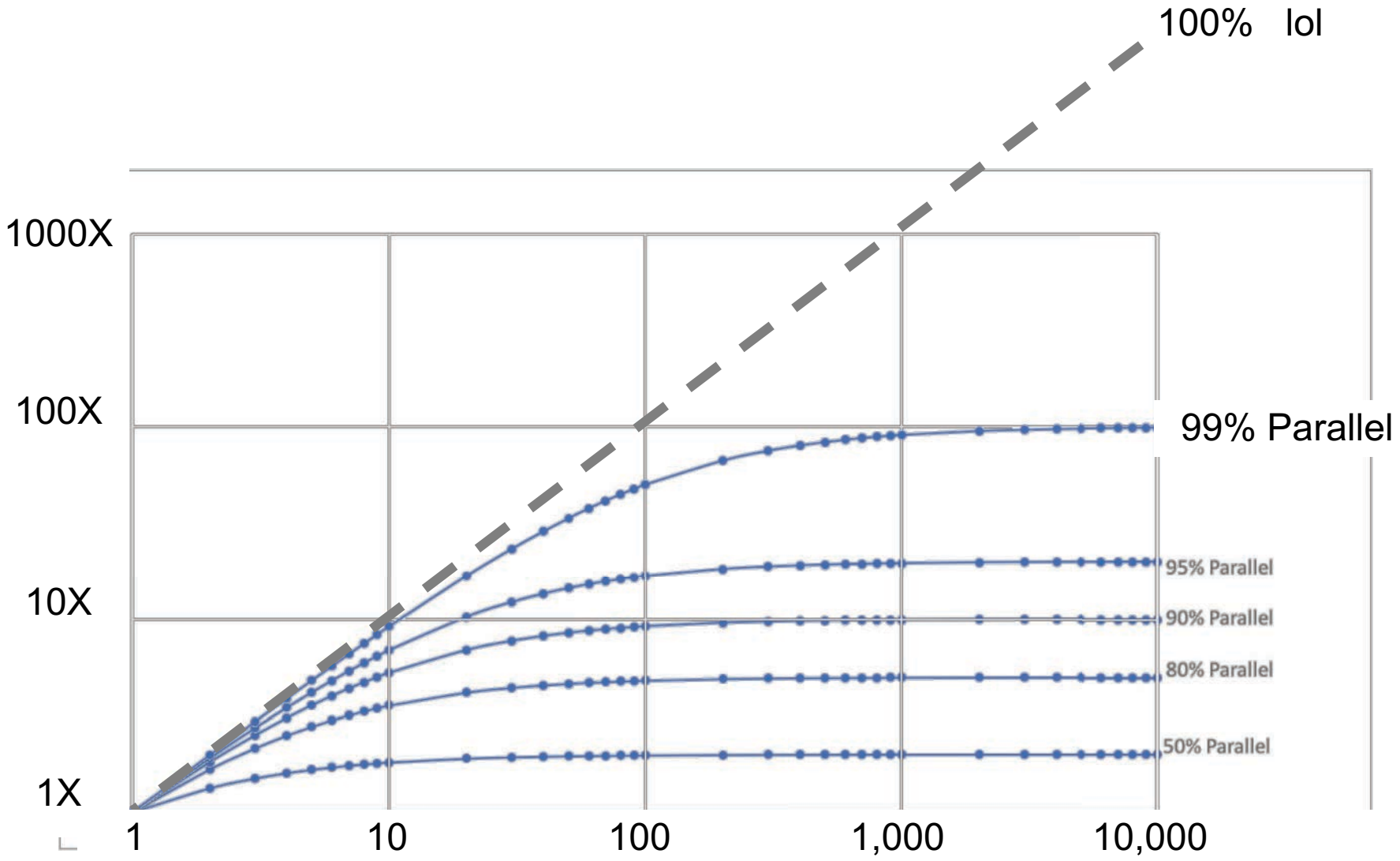

1000X
100X
10X
1X
1
10
100
1,000
10,000
50% Parallel

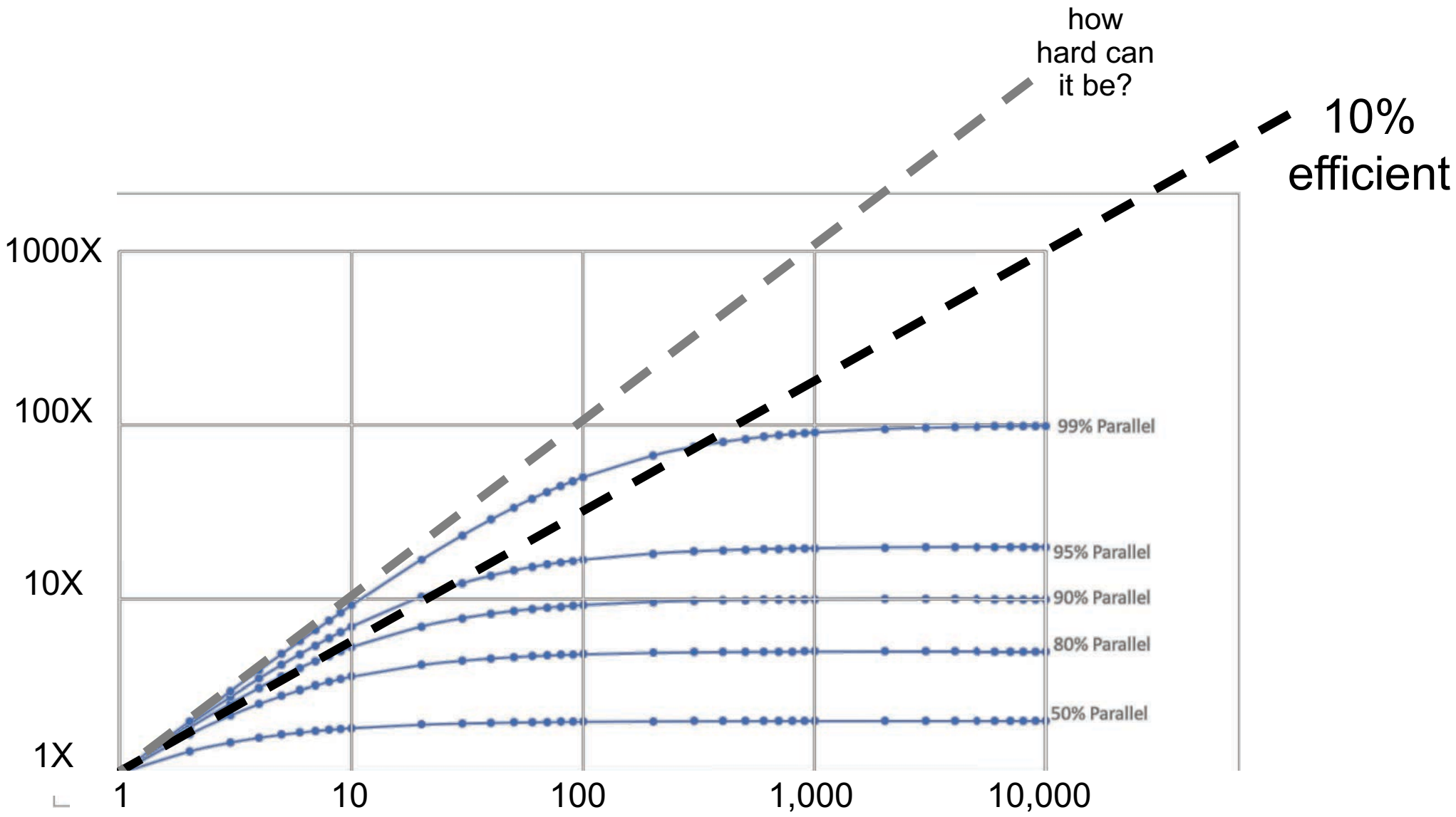

1000X
100X
10X
1X
1
10
100
1,000
10,000
80% Parallel
50% Parallel


1000X
100X
10X
1X
1
10
100
1,000
10,000
90% Parallel
80% Parallel
50% Parallel


1000X
100X
10X
1X
1
10
100
1,000
10,000
95% Parallel
90% Parallel
80% Parallel
50% Parallel

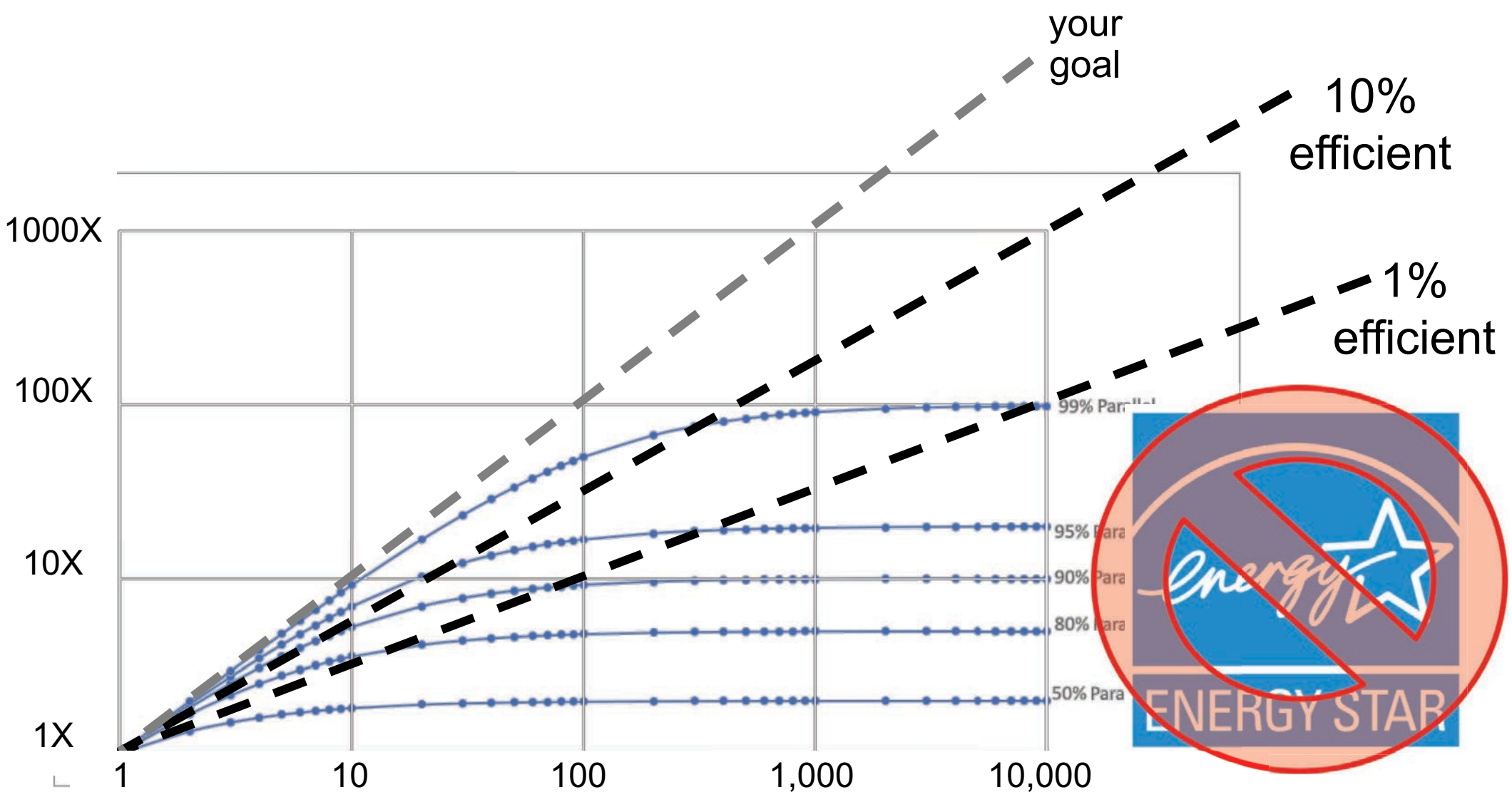

1000X
100X
10X
1X
1
10
100
1,000
10,000
99% Parallel
95% Parallel
90% Parallel
80% Parallel
50% Parallel


100% lol
1000X
100X
10X
1X
99% Parallel
95% Parallel
90% Parallel
80% Parallel
50% Parallel
1
10
100
1,000
10,000



Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT



Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

I have a 10,000 core
machine – but I insist on
50% efficiency.

How many cores
should I allow
you to use?

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

I have a 10,000 core machine – but I insist on 50% efficiency.

How many cores should I allow you to use?

	max that is 50% efficient
0.5	3
0.6	3
0.75	5
0.9	11
0.91	12
0.92	13
0.93	15
0.94	17
0.95	21
0.96	26
0.97	34
0.98	51
0.99	101
0.999	1,001
0.9999	10,000

at 95% parallel we can only keep 21 cores busy and be 50% efficient

(at 22 cores, we would slip to 48.8%)

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

I have a 10,000 core machine – but I insist on 50% efficiency.

How many cores should I allow you to use?

	max that is 10% efficient	max that is 33% efficient	max that is 50% efficient
0.5	19	5	3
0.6	23	6	3
0.75	37	9	5
0.9	91	21	11
0.91	101	23	12
0.92	113	26	13
0.93	129	30	15
0.94	151	34	17
0.95	181	41	21
0.96	226	51	26
0.97	301	68	34
0.98	451	102	51
0.99	900	204	101
0.999	9,001	2,031	1,001
0.9999	90,000	20,304	10,000

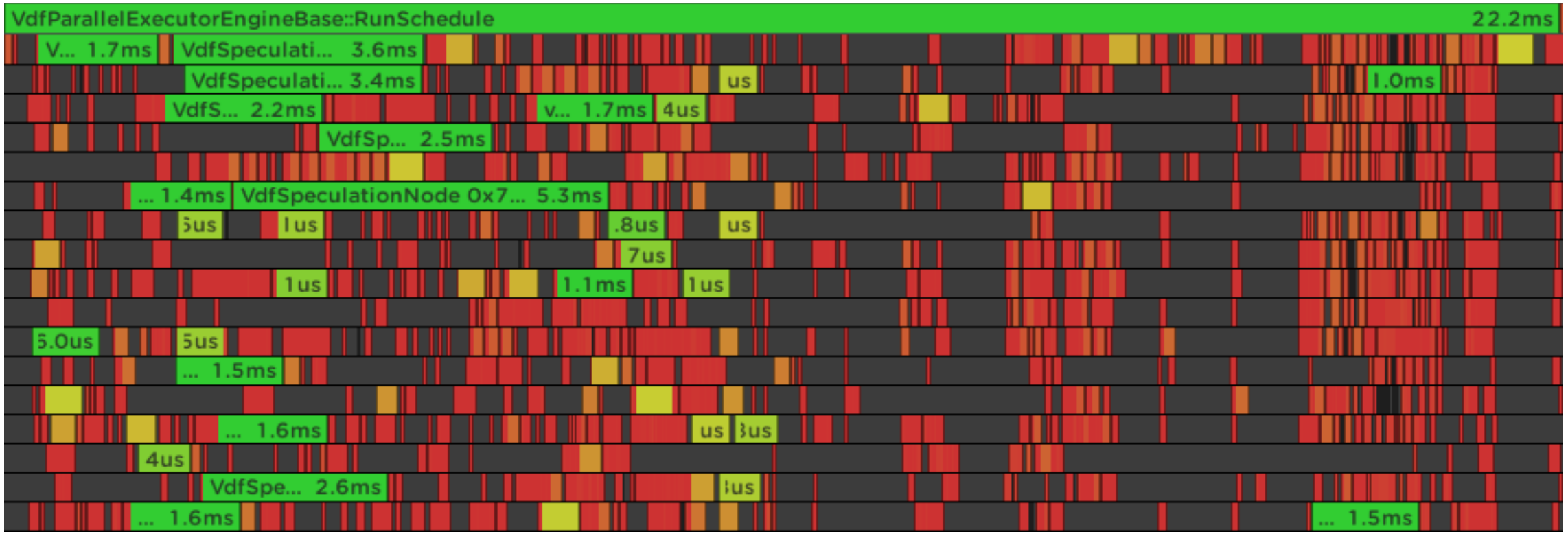
Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

REAL WORLD EXAMPLE

Glacier National Park, photo by James Reinders, August 4, 2017

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

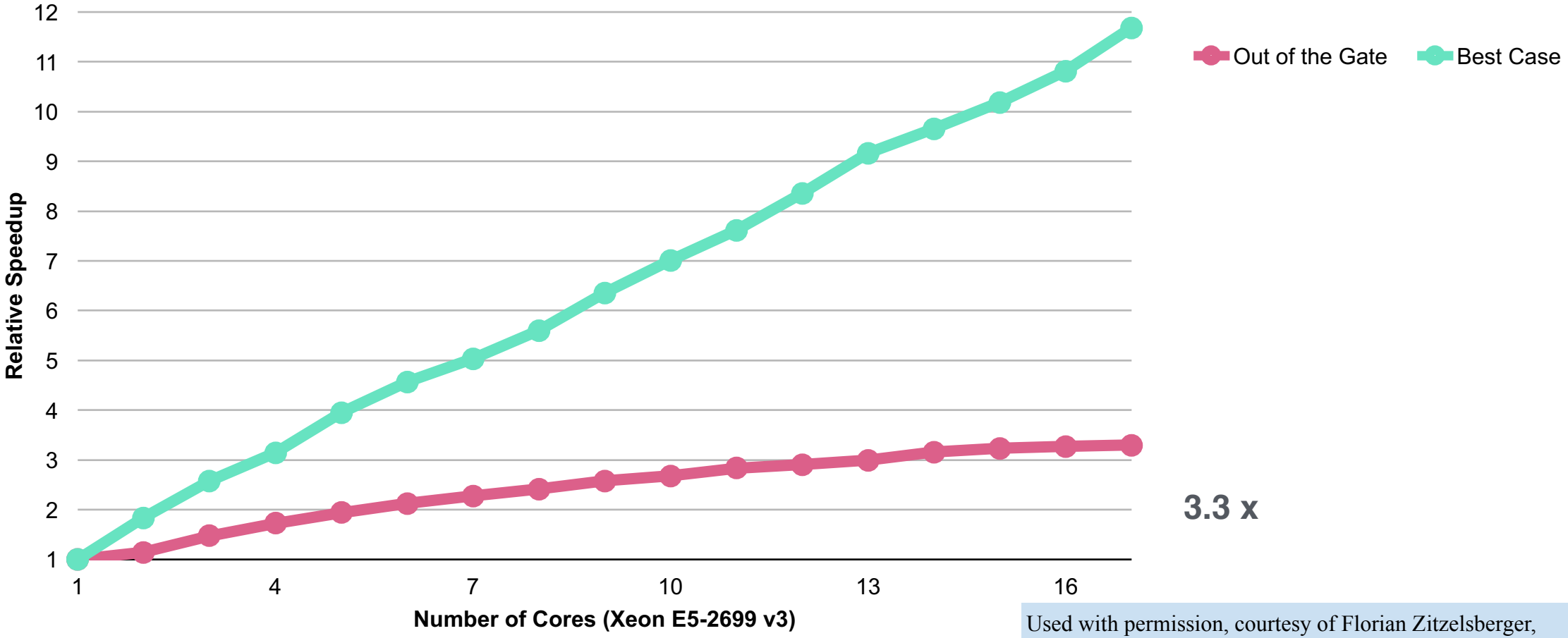
Single Frame



Underutilized?

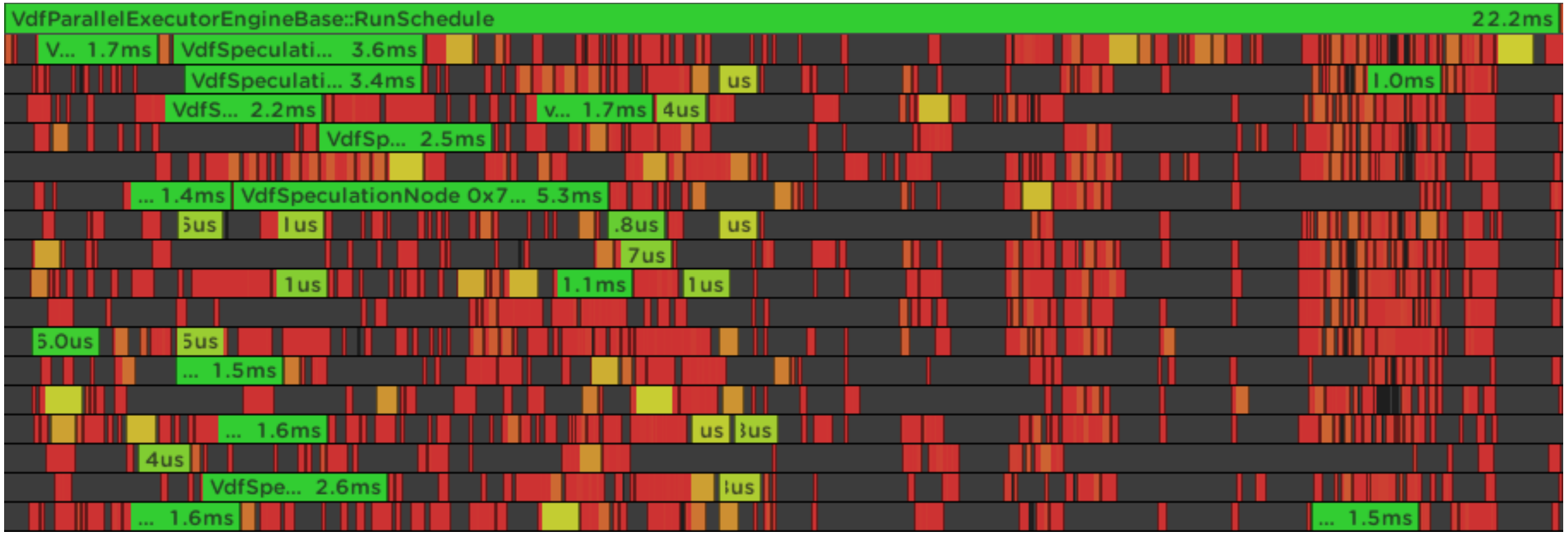
Adapted with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

Results



Used with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

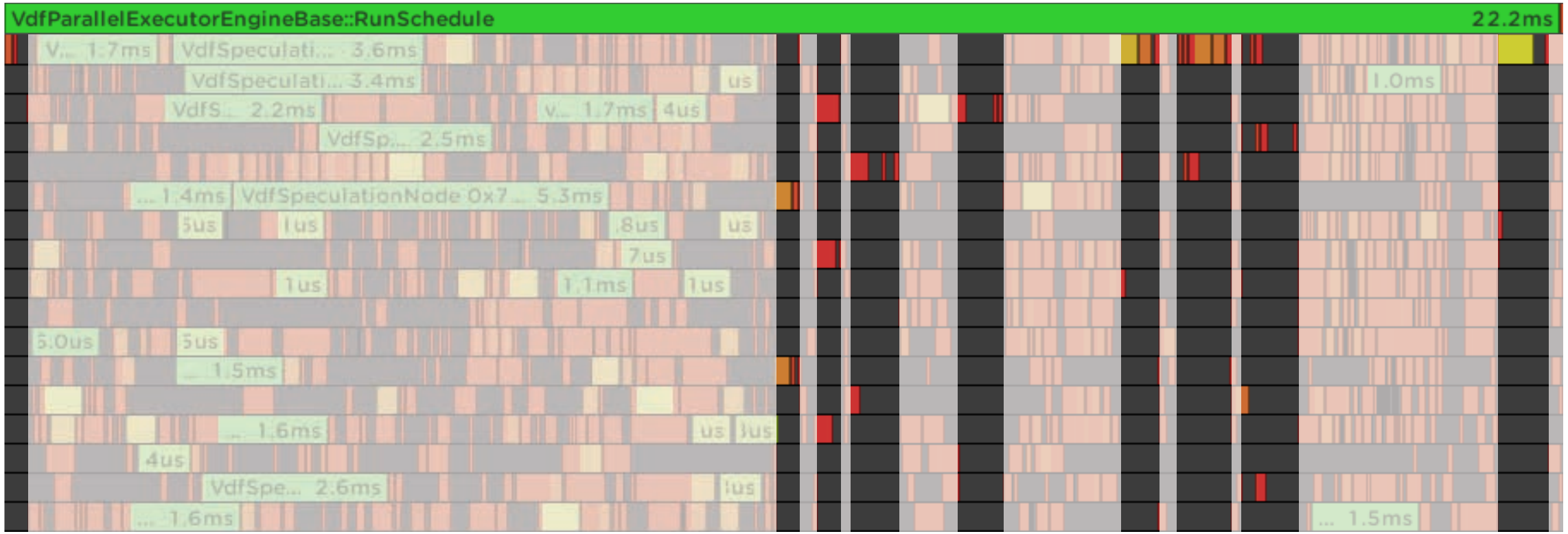
Single Frame



Underutilized?

Adapted with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

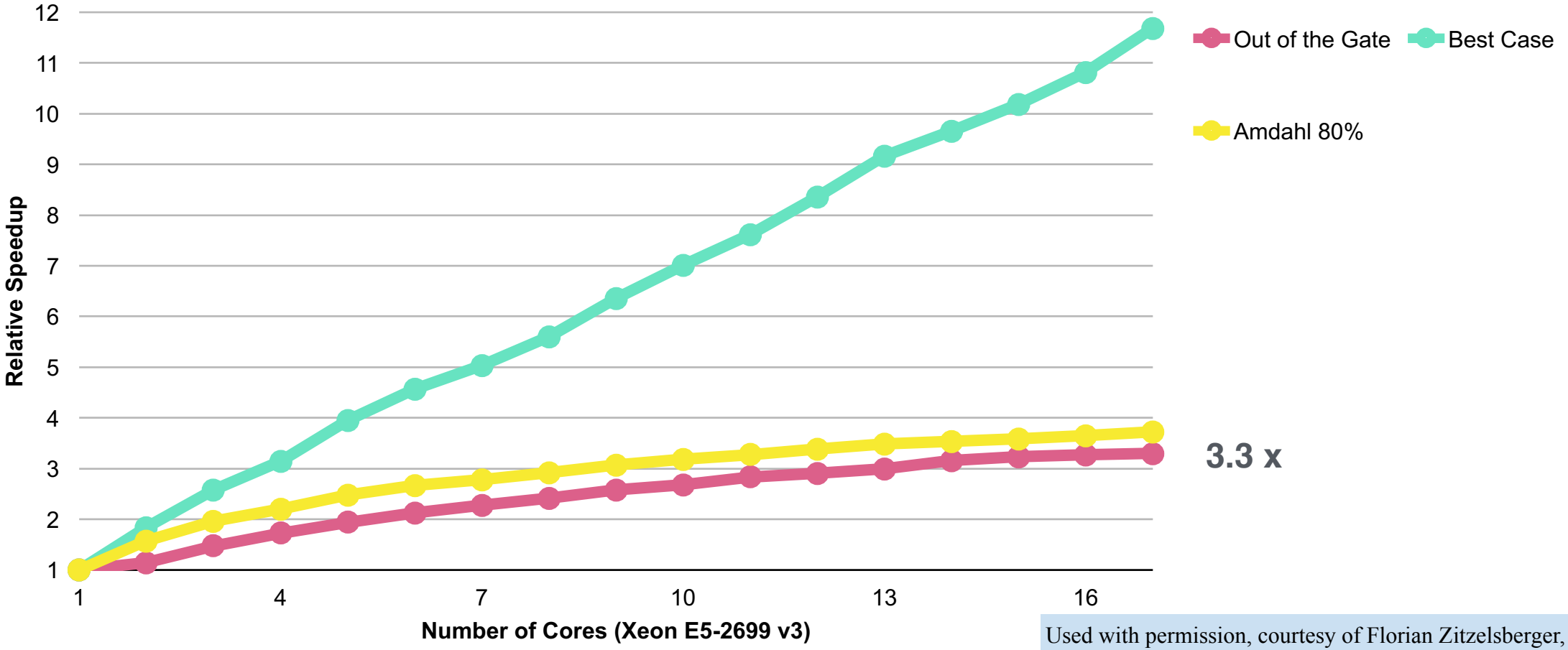
Single Frame



~ 20% Underutilized

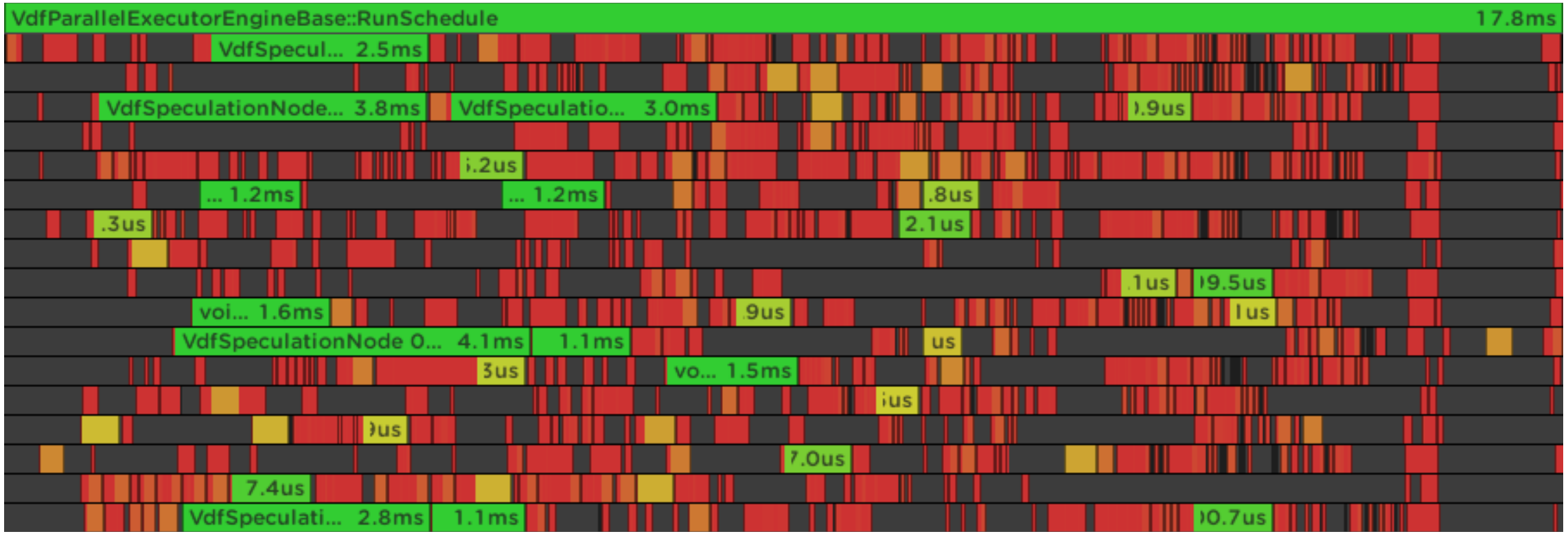
Used with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

Underutilization



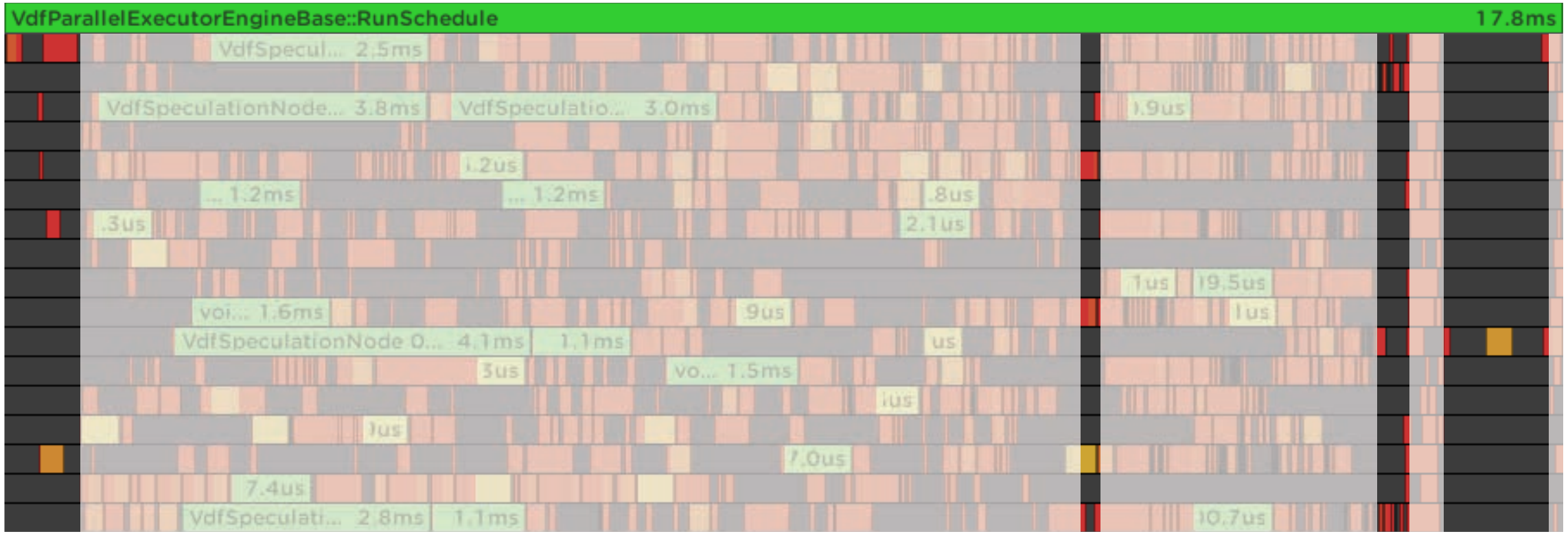
Used with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

Optimized Rig



Adapted with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

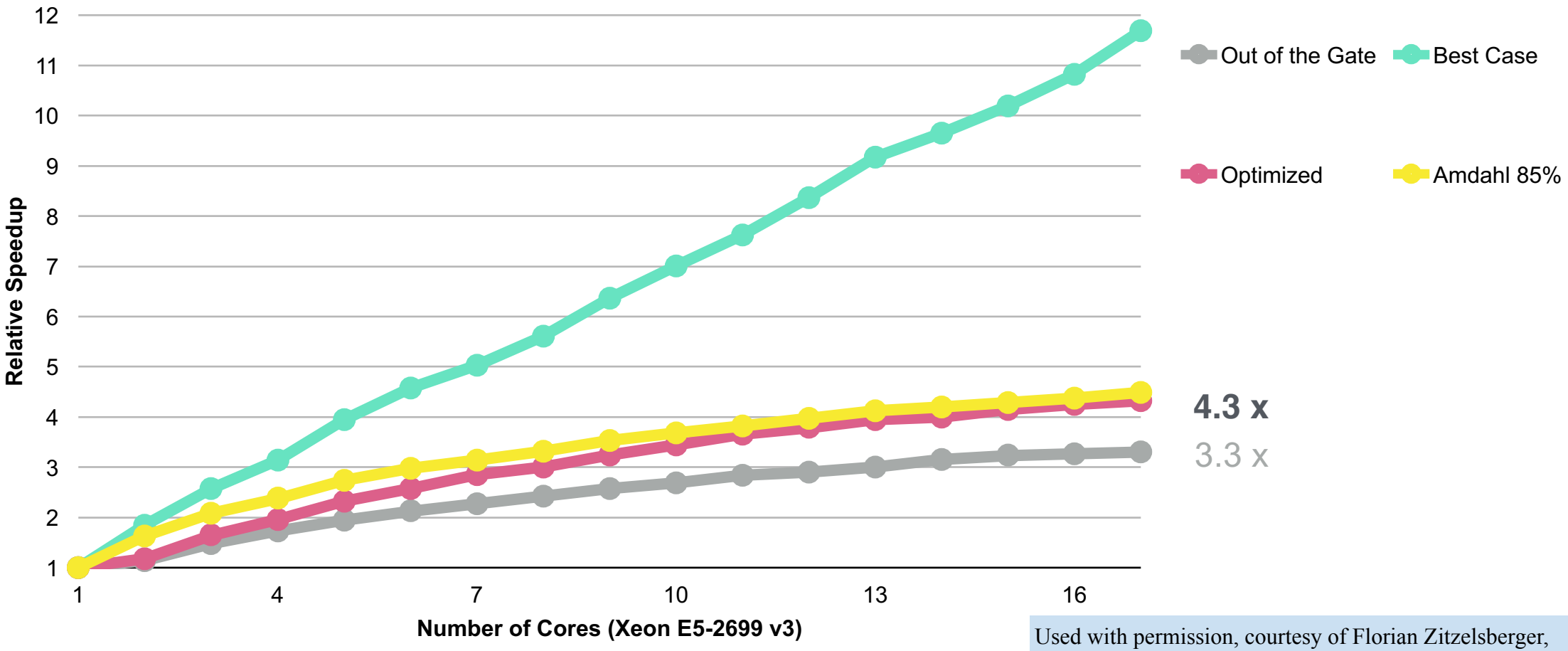
Optimized Rig



~ 15% Underutilized

Used with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

Optimized Rig



Used with permission, courtesy of Florian Zitzelsberger, Pixar Animation Studios, from "Building a Scalable Evaluation Engine for Presto" segment of "Multithreading for Visual Effects" talk at SIGGRAPH2017.

KEEP CALM AND LOOK FOR WHAT YOU DO NOT SEE

It is hard to “see” if you do not look.

It is hard to “see” if you do not look.

We could guess,
after all – we are smart enough
to *believe* we know what is happening.

Look for:
?

Look for:

- Confirmation

Look for:

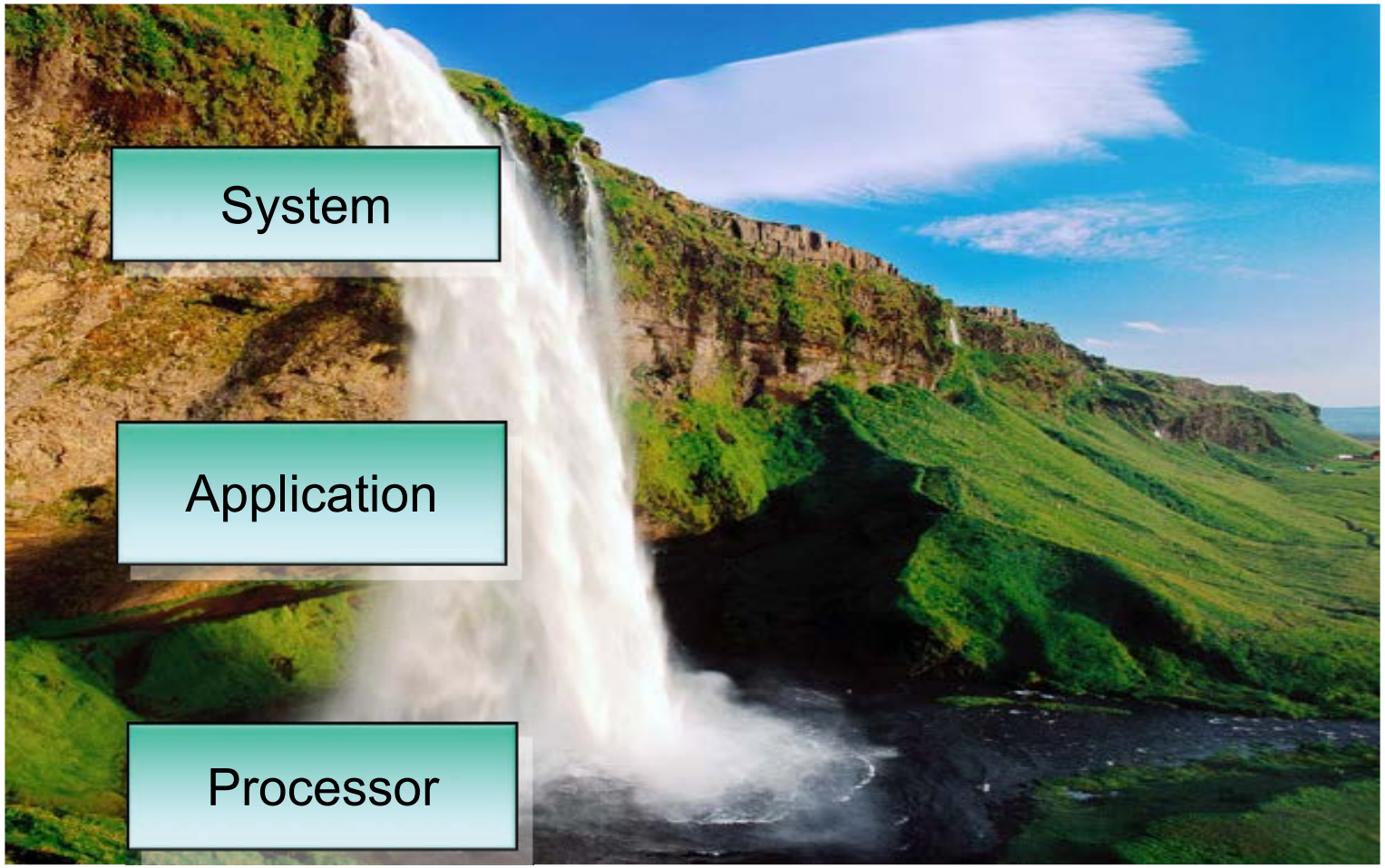
- Confirmation
- Surprises

Look for:

- Confirmation
- Surprises

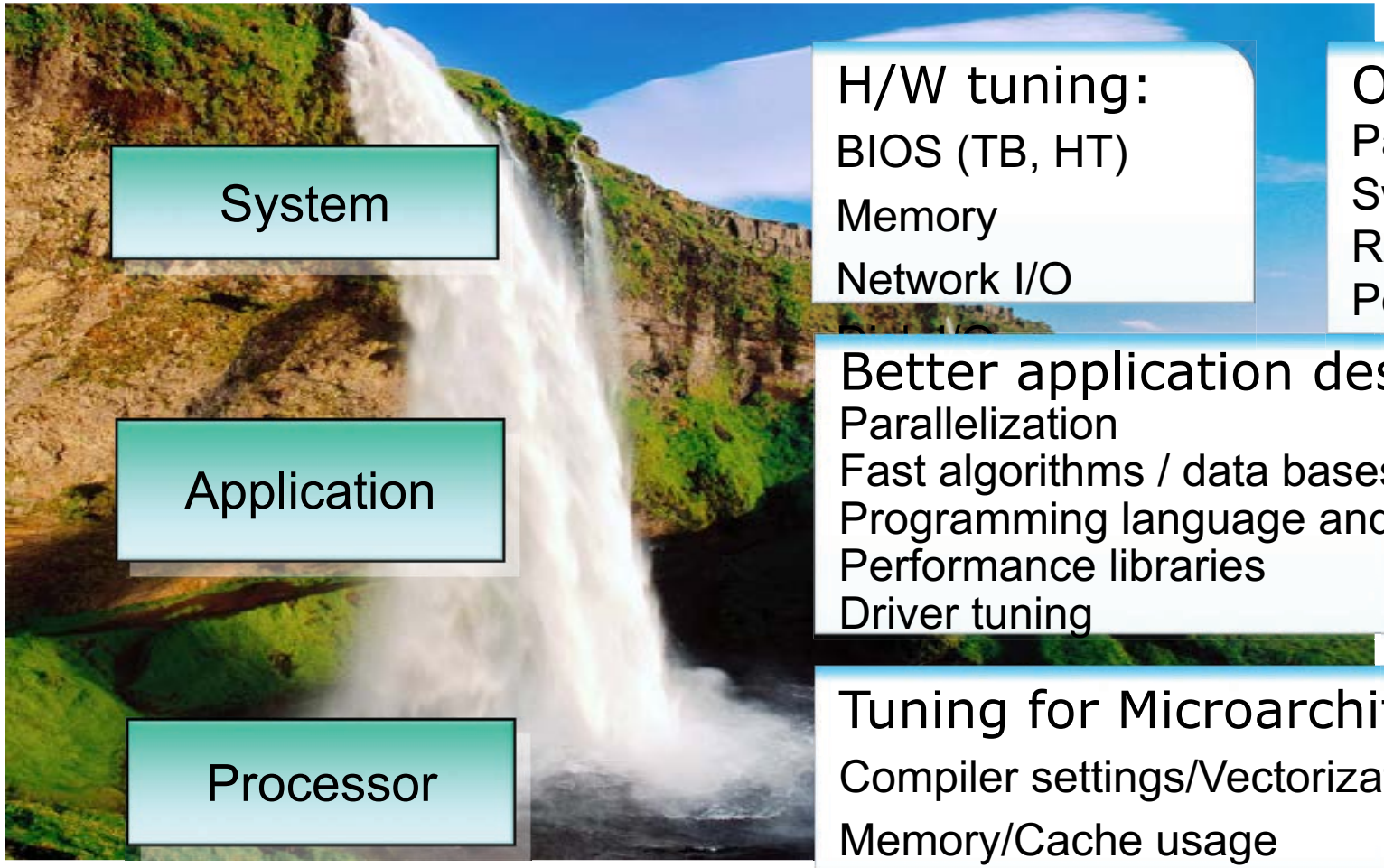
Your EXPERTISE will grow as you investigate.

Optimization: A Top-down Approach



https://software.intel.com/en-us/articles/de-mystifying-software-performance-optimization

Optimization: A Top-down Approach



System

Application

Processor

H/W tuning:
- BIOS (TB, HT)
- Memory
- Network I/O

OS tuning:
- Page size
- Swap file
- RAM Disk
- Power settings

Better application design:
- Parallelization
- Fast algorithms / data bases
- Programming language and RT libs
- Performance libraries
- Driver tuning

Tuning for Microarchitecture:
- Compiler settings/Vectorization
- Memory/Cache usage
- CPU pitfalls

OS, System Expertise
SW/uArch

https://software.intel.com/en-us/articles/de-mystifying-software-performance-optimization

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Optimization: A Top-down Approach

System

Application

Processor

H/W tuning:
BIOS (TB, HT)
Memory
Network I/O

OS tuning:
Page size
Swap file
RAM Disk
Power settings

Better application design:
Parallelization
Fast algorithms / data bases
Programming language and RT libs
Performance libraries
Driver tuning

Tuning for Microarchitecture:
Compiler settings/Vectorization
Memory/Cache usage
CPU pitfalls

OS, System
SW/uArch
Expertise

https://software.intel.com/en-us/articles/de-mystifying-software-performance-optimization

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Application Tuning

Who: Software Developers, Performance Engineers, Domain Experts

How:

- Workload selection
  - Repeatable results
  - Steady stat
- Define Metrics and Collect Baseline
  - Wall-clock time, FLOPS, FPS
  - <insert your metric here>
- Identify Hotspots
  - Focus effort where it counts
  - Use Tools
- Determine inefficiencies
  - Is there parallelism?
  - Are you memory bound?
  - Will better algorithms or programming languages help?

This step often requires some knowledge of the application and its algorithms

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Application Tuning
Find Hotspots

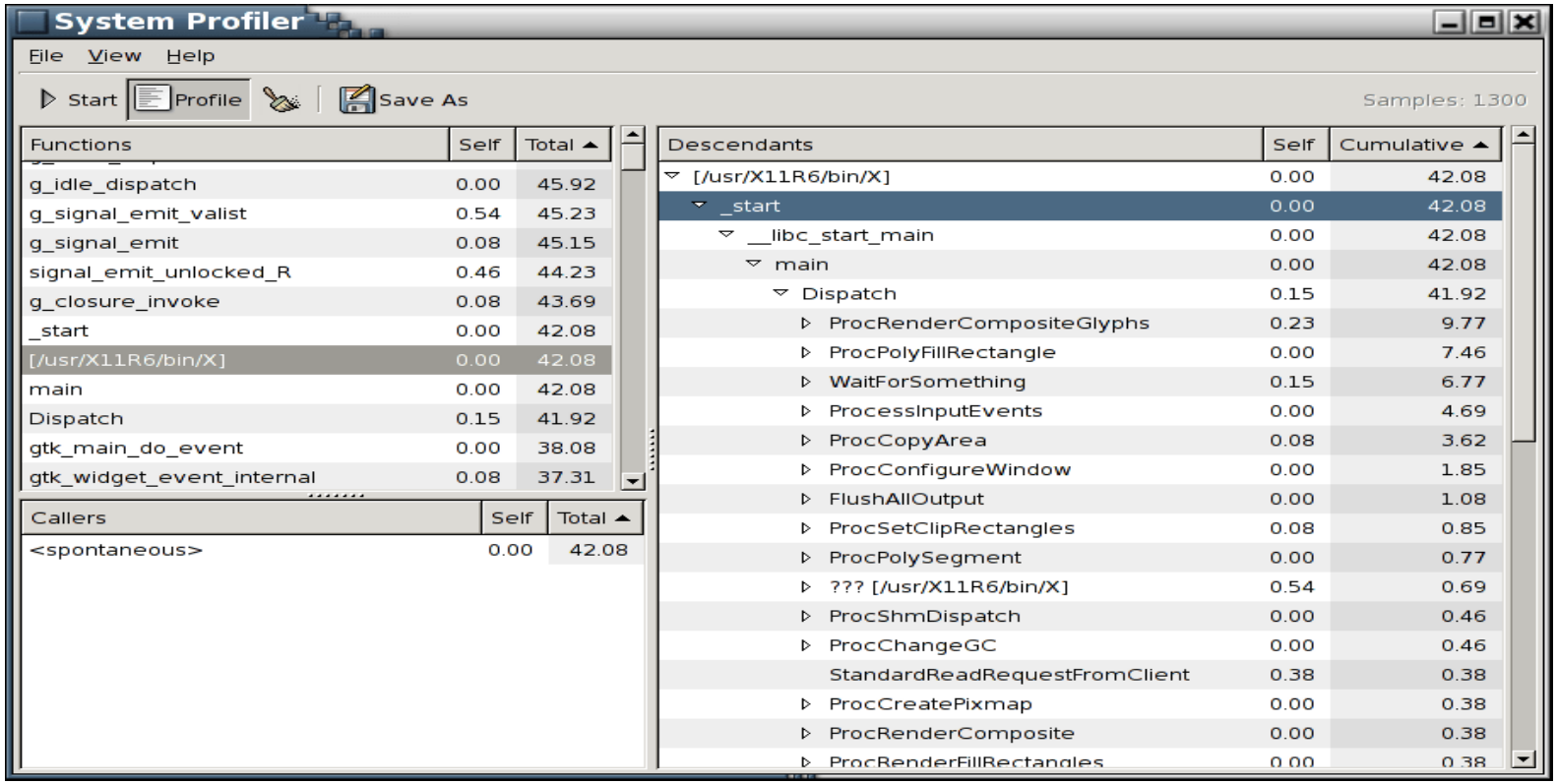
- This could be at the module, function, or source code level
- Determine your own granularity

```
$ opreport --exclude-dependent --demangle=smart --symbols `which lyx`
CPU: PIII, speed 863.195 MHz (estimated)
Counted CPU_CLK_UNHALTED events (clocks processor is not halted) with a unit mask of 0x00 (No unit mask)
vma       samples  %            symbol name
081ec974 5016      8.5096       _Rb_tree<unsigned short, pair<unsigned short const, int>,  unsigned short
0810c4ec 3323      5.6375       Paragraph::getFontSettings(BufferParams const&, int) const
081319d8 3220      5.4627       LyXText::getFont(Buffer const*, Paragraph*, int) const
080e45d8 3011      5.1082       LyXFont::realize(LyXFont const&)
080e3d78 2623      4.4499       LyXFont::LyXFont()
081255a4 1823      3.0927       LyXText::singleWidth(BufferView*, Paragraph*, int, char) const
080e3cf0 1804      3.0605       operator==(LyXFont::FontBits const&, LyXFont::FontBits const&)
081128e0 1729      2.9332       Paragraph::Pimpl::getChar(int) const
081ed020 1380      2.3412       font_metrics::width(char const*, unsigned, LyXFont const&)
08110d60 1310      2.2224       Paragraph::getChar(int) const
081ebc94 1227      2.0816       qfont_loader::getfontinfo(LyXFont const&)
...
```

oprofile: http://oprofile.sourceforge.net/

Application Tuning
Find Hotspots

- This could be at the module, function, or source code level
- Determine your own granularity



sysprof: http://sysprof.com

Application Tuning
Find Hotspots

- This could be at the module, function, or source code level
- Determine your own granularity

Grouping: Function / Call Stack

Function / Call Stack	CPU Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Wait Time by Utilization (Idle, Poor, Ok, Ideal)	Overhead and Spin Time	Thread Oversubscription	Module	Start Address	Fun
grid_intersect	5.360s		0s	4.527s	3_tachyon_omp.exe	0x40c7f0	grid_intersect
intersect_objects	5.084s		0s	4.291s	3_tachyon_omp.exe	0x402130	intersect_objects(struct ray *)
grid_intersect ← intersect_objects	0.276s		0s	0.237s	3_tachyon_omp.exe	0x40c7f0	grid_intersect
sphere_intersect	3.542s		0s	2.914s	3_tachyon_omp.exe	0x40aca0	sphere_intersect
grid_intersect	3.542s		0s	2.914s	3_tachyon_omp.exe	0x40c7f0	grid_intersect
SwitchToThread	0.986s		0.986s	0.901s	KERNELBASE.dll	0x10047e49	SwitchToThread
video::next_frame ← thread_trace ← __kmp_invo	0.811s		0.811s	0.727s	3_tachyon_omp.exe	0x402770	video::next_frame(void)
video::main_loop ← main ← WinMain ← _tmain	0.175s		0.175s	0.175s	3_tachyon_omp.exe	0x402990	video::main_loop(void)
__kmp_launch_thread	0.874s	2.104s	0.874s	0.008s	libiomp5md.dll	0x1004b0d0	__kmp_launch_thread
grid_bounds_intersect	0.297s		0s	0.215s	3_tachyon_omp.exe	0x40c4f0	grid_bounds_intersect
shader	0.106s		0s	0.066s	3_tachyon_omp.exe	0x406b50	shader(struct ray *)
GdipDrawImagePointRectI	0.098s		0s	0.098s	gdiplus.dll	0x10060336	GdipDrawImagePointRectI
pos2grid	0.090s		0s	0.074s	3_tachyon_omp.exe	0x40c410	pos2grid
Raypnt	0.073s		0s	0.073s	3_tachyon_omp.exe	0x406610	Raypnt(struct ray *,double)
tri_intersect	0.057s		0s	0.048s	3_tachyon_omp.exe	0x40b340	tri_intersect
camray	0.048s		0s	0.038s	3_tachyon_omp.exe	0x401c70	camray(struct scenedef *,int,in
Selected 1 row(s):	5.360s		0s	4.527s			

Intel® VTune™ Amplifier XE: http://intel.ly/vtune-amplifier-xe

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Application Tuning
Find Hotspots

- This could be at the module, function, or source code level
- Determine your own granularity

Source | Assembly | Assembly grouping: Address

So. Li.	Source	CPU Time: Total by Utilization (Idle, Poor, Ok, Ideal, Over)
558	cur = cur->next;	0.020s
559	}	
560	curvox.x += step.x;	
561	if (ry->maxdist < tmax.x || curvox.x == out.x)	0.011s
562	break;	
563	voxindex += step.x;	
564	tmax.x += tdelta.x;	
565	curpos = nXp;	
566	nXp.x += pdeltaX.x;	
567	nXp.y += pdeltaX.y;	
568	nXp.z += pdeltaX.z;	
569	}	
570	else if (tmax.z < tmax.y) {	0.019s
571	cur = g->cells[voxindex];	0.232s
572	while (cur != NULL) {	0.030s
573	if (ry->mbox[cur->obj->id] != ry->serial) {	2.020s
574	ry->mbox[cur->obj->id] = ry->serial;	1.212s
575	cur->obj->methods->intersect(cur->obj, ry);	0.577s
576	}	
577	cur = cur->next;	0.695s
578	}	
579	curvox.z += step.z;	0.014s
580	if (ry->maxdist < tmax.z || curvox.z == out.z)	0.036s
581	break;	
582	voxindex += step.z*g->xsize*g->ysize;	
	Selected 1 row(s):	2.020s

Address	Sour... Line	Assembly	CPU Time: Total by Utilization (Idle, Poor, Ok, Ideal, Over)
0x40cd21	573	cmp dword ptr [eax+edx*4], ecx	1.198s
0x40cd24	573	jz 0x40cd3d <Block 49>	0.011s
0x40cd26		**Block 47:**	
0x40cd26	574	mov edx, dword ptr [esi+0x4]	1.018s
0x40cd29	574	mov edx, dword ptr [edx]	0.137s
0x40cd2b	574	mov dword ptr [eax+edx*4], ecx	0.058s
0x40cd2e	575	mov eax, dword ptr [esi+0x4]	0.103s
0x40cd31	575	mov ecx, dword ptr [eax+0x8]	0.037s
0x40cd34	575	mov edx, dword ptr [ecx]	0.047s
0x40cd36	575	push edi	0.032s
0x40cd37	575	push eax	0.072s
0x40cd38	575	call edx	0.275s
0x40cd3a		**Block 48:**	
0x40cd3a	575	add esp, 0x8	0.020s
0x40cd3d		**Block 49:**	
0x40cd3d	577	mov esi, dword ptr [esi]	0.551s
0x40cd3f	577	test esi, esi	0.145s
0x40cd41	577	jnz 0x40cd16 <Block 46>	
0x40cd43		**Block 50:**	
0x40cd43	572	movsd xmm0, qword ptr [esp+0x78]	0.030s
0x40cd49	572	mov eax, dword ptr [esp+0x48]	0.000s
0x40cd4d		**Block 51:**	
0x40cd4d	579	mov ecx, dword ptr [esp+0x58]	
0x40cd51	579	add dword ptr [esp+0x40], ecx	0.014s
0x40cd55	580	comisd xmm0, qword ptr [edi+0x48]	
		Highlighted 6 row(s):	

Intel® VTune™ Amplifier XE: http://intel.ly/vtune-amplifier-xe

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Application Tuning
Find Hotspots

- This could be at the module, function, or source code level
- Determine your own granularity

Source | Assembly | Assembly grouping: Address

So. Li.	Source	CPU Time: Total by Utilization (Idle, Poor, Ok, Ideal, Over)
558	cur = cur->next;	0.020s
559	}	
560	curvox.x += step.x;	
561	if (ry->maxdist < tmax.x || curvox.x == out.x)	0.011s
562	break;	
563	voxindex += step.x;	
564	tmax.x += tdelta.x;	
565	curpos = nXp;	
566	nXp.x += pdeltaX.x;	
567	nXp.y += pdeltaX.y;	
568	nXp.z += pdeltaX.z;	
569	}	
570	else if (tmax.z < tmax.y) {	0.019s
571	cur = g->cells[voxindex];	0.232s
572	while (cur != NULL) {	0.030s
573	if (ry->mbox[cur->obj->id] != ry->serial) {	2.020s
574	ry->mbox[cur->obj->id] = ry->serial;	1.212s
575	cur->obj->methods->intersect(cur->obj, ry);	0.577s
576	}	
577	cur = cur->next;	0.695s
578	}	
579	curvox.z += step.z;	0.014s
580	if (ry->maxdist < tmax.z || curvox.z == out.z)	0.036s
581	break;	
582	voxindex += step.z*g->xsize*g->ysize;	
	Selected 1 row(s):	2.020s

Address	Sour... Line	Assembly	CPU Time: Total by Utilization (Idle, Poor, Ok, Ideal, Over)
0x40cd21	573	cmp dword ptr [eax+edx*4], ecx	1.198s
0x40cd24	573	jz 0x40cd3d <Block 49>	0.011s
0x40cd26		**Block 47:**	
0x40cd26	574	mov edx, dword ptr [esi+0x4]	1.018s
0x40cd29	574	mov edx, dword ptr [edx]	0.137s
0x40cd2b	574	mov dword ptr [eax+edx*4], ecx	0.058s
0x40cd2e	575	mov eax, dword ptr [esi+0x4]	0.103s
0x40cd31	575	mov ecx, dword ptr [eax+0x8]	0.037s
0x40cd34	575	mov edx, dword ptr [ecx]	0.047s
0x40cd36	575	push edi	0.032s
0x40cd37	575	push eax	0.072s
0x40cd38	575	call edx	0.275s
0x40cd3a		**Block 48:**	
0x40cd3a	575	add esp, 0x8	0.020s
0x40cd3d		**Block 49:**	
0x40cd3d	577	mov esi, dword ptr [esi]	0.551s
0x40cd3f	577	test esi, esi	0.145s
0x40cd41	577	jnz 0x40cd16 <Block 46>	
0x40cd43		**Block 50:**	
0x40cd43	572	movsd xmm0, qword ptr [esp+0x78]	0.030s
0x40cd49	572	mov eax, dword ptr [esp+0x48]	0.000s
0x40cd4d		**Block 51:**	
0x40cd4d	579	mov ecx, dword ptr [esp+0x58]	
0x40cd51	579	add dword ptr [esp+0x40], ecx	0.014s
0x40cd55	580	comisd xmm0, qword ptr [edi+0x48]	
		Highlighted 6 row(s):	

This may reinforce your understanding of the application but often reveals surprises

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Application Tuning
Resource Utilization

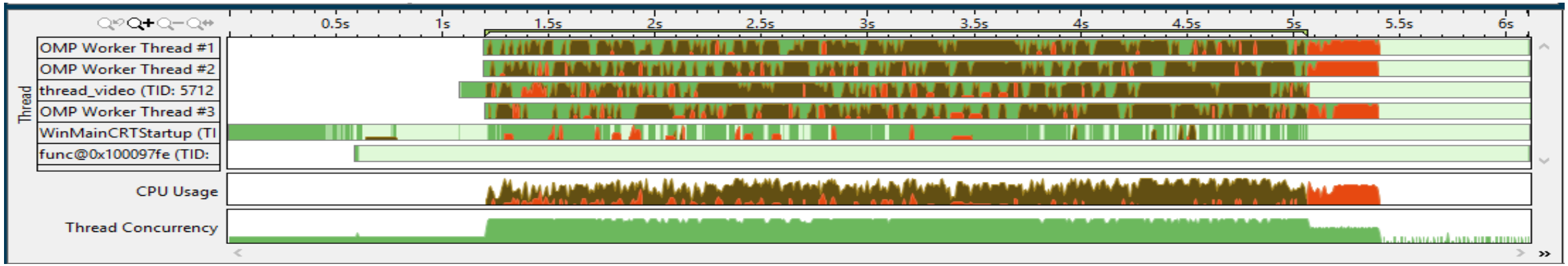
- Is the application parallel?
- Multi-thread vs. Multi-process
- Memory Bound?

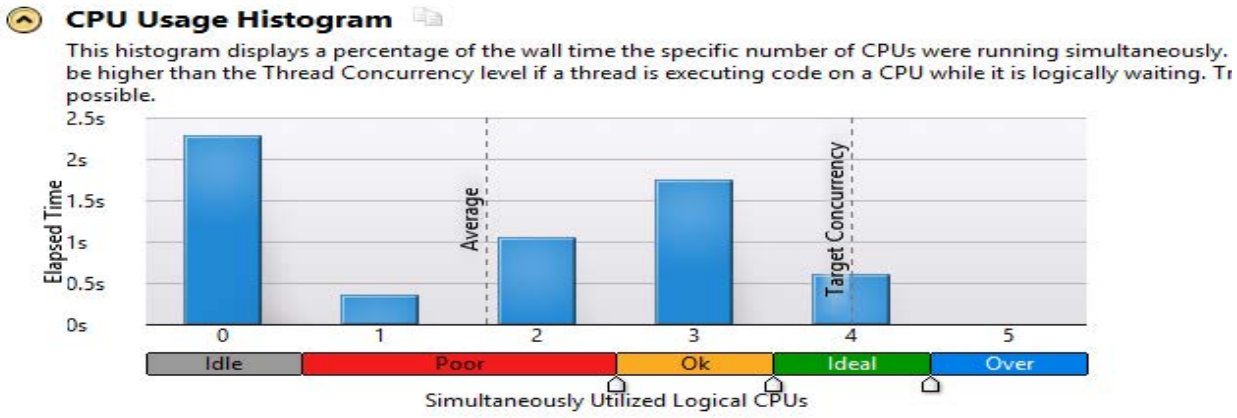
```
last pid: 86494;  load averages:  0.83,  0.65,  0.69   up 67+22:48:43  14:44:15
227 processes: 1 running, 224 sleeping, 2 zombie
CPU: 20.2% user,  0.0% nice,  6.5% system,  0.2% interrupt, 73.1% idle
Mem: 1657M Active, 1868M Inact, 273M Wired, 190M Cache, 112M Buf, 11M Free
Swap: 4500M Total, 249M Used, 4251M Free, 5% Inuse

  PID USERNAME  THR PRI NICE   SIZE    RES STATE  C   TIME   WCPU COMMAND
86460 www         1   4    0   150M 30204K accept 1   0:02 11.18% php-cgi
86458 www         1   4    0   150M 29912K accept 0   0:02  8.98% php-cgi
86463 pgsql       1   4    0   949M    99M sbwait 1   0:01  7.96% postgres
85885 www         1   4    0   150M 35204K accept 2   0:07  7.57% php-cgi
85274 www         1   4    0   149M 40868K sbwait 3   0:27  5.18% php-cgi
85267 www         1   4    0   151M 40044K sbwait 2   0:33  4.59% php-cgi
85884 www         1   4    0   150M 41584K accept 2   0:14  4.59% php-cgi
85887 pgsql       1   4    0   951M   128M sbwait 1   0:04  4.20% postgres
85886 pgsql       1   4    0   949M   161M sbwait 0   0:08  3.37% postgres
86459 pgsql       1   4    0   949M 75960K sbwait 2   0:01  3.37% postgres
85279 pgsql       1   4    0   950M   192M sbwait 2   0:14  2.39% postgres
85269 pgsql       1   4    0   950M   199M sbwait 1   0:19  2.20% postgres
85268 www         1   4    0   152M 44356K sbwait 2   0:32  1.17% php-cgi
85273 pgsql       1   4    0   950M   215M sbwait 0   0:19  1.17% postgres
97082 pgsql       1  44    0 26020K  6832K select 0  46:55  0.00% postgres
  892 root        1   4    0  3160K     8K -      2  13:33  0.00% nfsd
 1796 root        1  44    0 19780K 13660K select 3  12:43  0.00% Xvfb
```

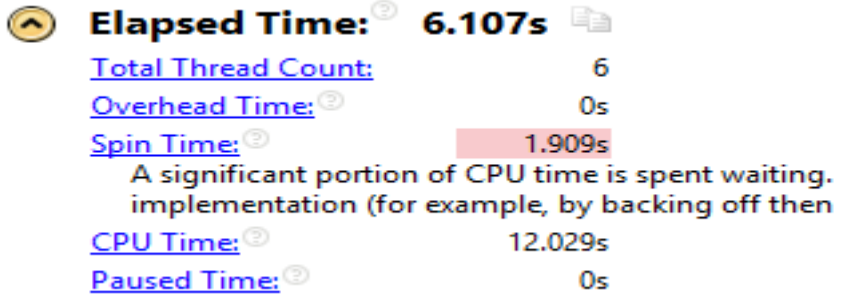

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Application Tuning
Resource Utilization

- Is the application parallel?

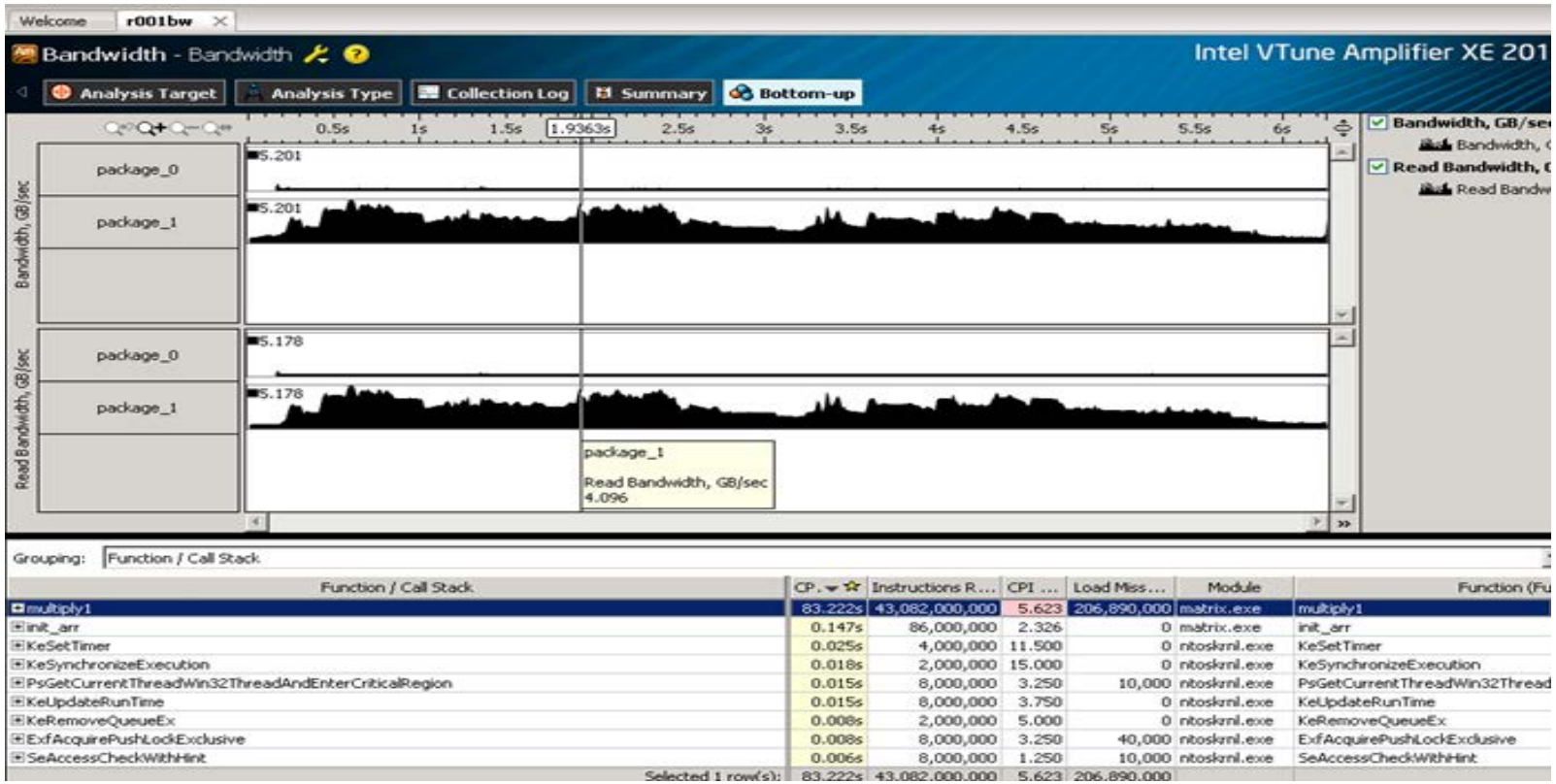






Application Tuning
Resource Utilization

- Memory Bound?



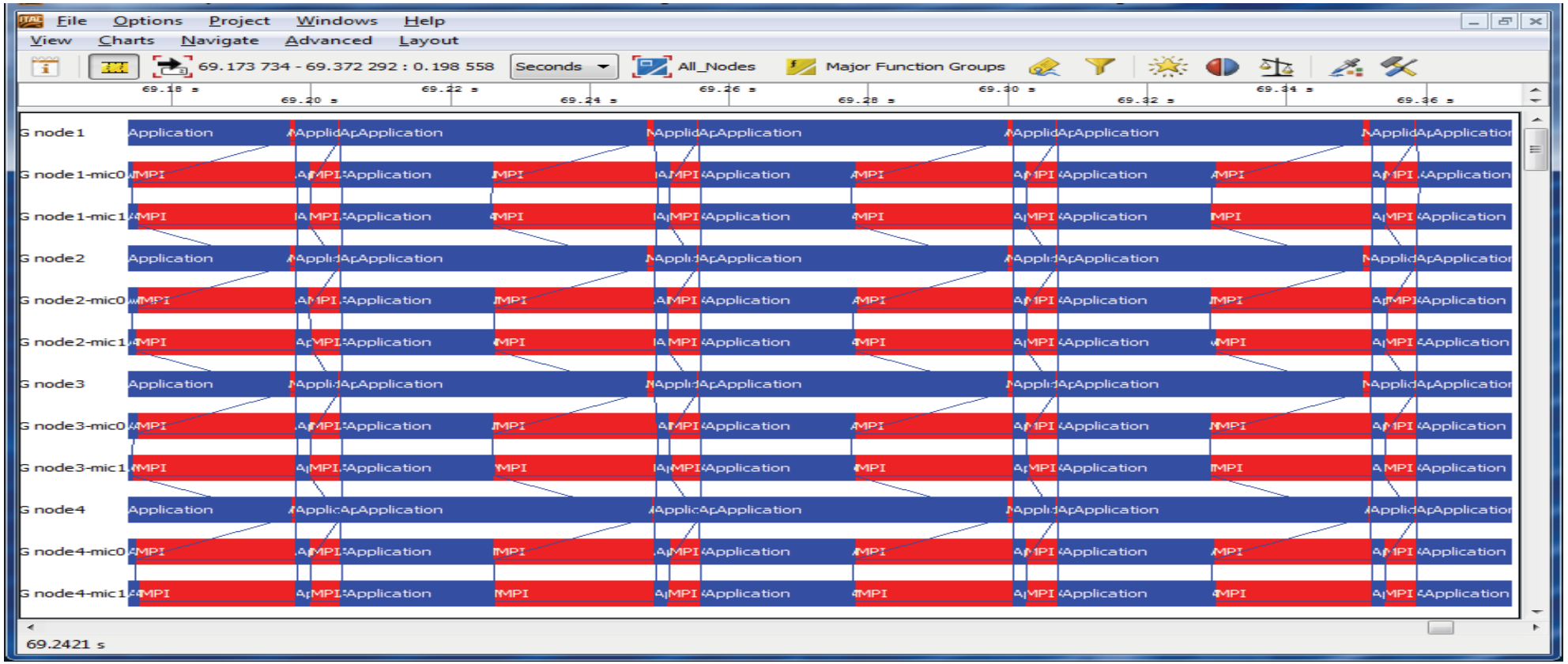
- Know your max theoretical memory bandwidth

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Application Tuning
Resource Utilization

MPI applications have added communication complexity



Intel® Trace Analyzer and Collector: http://intel.ly/traceanalyzer-collector

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Application Tuning

What's Next?

- If your Hotspots are common algorithms:
  - Look for optimized libraries
- If your Hotspots are uncommon:
  - Compiler optimizations
  - Expert analysis and refactoring of an algorithm
    - The opposite of "low-hanging fruit"
  - Deeper analysis of hardware performance
    - More on this later
- If the system is underutilized:
  - Add parallelism - multi-thread or multi-process
    - OpenMP, TBB, MPI, etc…

- Tools can help you determine where to look and may identify some issues.
- Some tools may provide suggestions for fixes.
- In the end – the developer and/or expert has to make the changes and decisions – there is no silver bullet.

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

What is a *cache miss*?

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

What happens if we reduce the number of *cache misses*?

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Performance Monitoring Unit (PMU)

- Registers on Intel CPUs to count architectural events
  - E.g. Instructions, Cache Misses, Branch Mispredict
- Events can be counted or sampled
  - Sampled events include Instruction Pointer
- Raw event counts are difficult to interpret
  - Use a tool like VTune or Perf with predefined metrics

Argonne NATIONAL LABORATORY
ECP EXASCALE COMPUTING PROJECT

Raw PMU Event Counts or Metrics – you choose as you like

Grouping: Function / Call Stack

Function / Call Stack	CPU_CL...	CPU_CLK_U...	INST_RETIRE...	L1D_PEND_...	OFF..	BR_MISP...	CPU_CLK_U...	CYCLE_AC...	CYCLE_AC...	DTL...	DTLB_LO...	DTLB_L...	DTL...	DTLB_ST...	DTLB_S...	ICACH..
grid_intersect	13,604,020,406	14,118,021,177	12,572,018,858	6,344,009,516	0	52,001,170	14,924,022,386	5,408,008,112	4,264,006,396	0	234,000,351	26,000,039	0	7,800,234	0	
sphere_intersect	8,706,013,059	9,134,013,701	8,494,012,741	4,238,006,357	0	15,600,351	9,464,014,196	3,016,004,524	2,808,004,212	0	104,000,156	26,000,039	0	10,400,312	0	
grid_bounds_intersect	984,001,476	1,004,001,506	672,001,008	104,000,156	0	15,600,351	962,001,443	312,000,468	286,000,429	0	0	0	0	0	0	
__kmp_end_split_barrier	676,001,014	624,000,936	460,000,690	0	0	0	0	0	0	0	0	0	0	0	0	
__kmp_x86_pause	228,000,342	224,000,336	122,000,183	0	0	10,400,234	0	0	0	0	0	0	0	0	0	
shader	216,000,324	242,000,363	142,000,213	104,000,156	0	0	208,000,312	104,000,156	52,000,078	0	0	0	0	2,600,078	0	
Raypnt	206,000,309	210,000,315	208,000,312	0	0	0	234,000,351	52,000,078	78,000,117	0	0	0	0	0	0	2,600,03
pos2grid	204,000,306	248,000,372	180,000,270	26,000,039	0	0	390,000,585	26,000,039	52,000,078	0	0	0	0	0	0	
tri_intersect	168,000,252	208,000,312	180,000,270	0	0	0	104,000,156	78,000,117	52,000,078	0	52,000,078	0	0	0	0	
VScale	124,000,186	126,000,189	164,000,246	0	0	0	234,000,351	52,000,078	0	0	0	0	0	0	0	
__kmp_yield	96,000,144	98,000,147	200,000,300	0	0	0	0	0	0	0	0	0	0	0	0	
Selected 1 row(s):	13,604,020,406	14,118,021,177	12,572,018,858	6,344,009,516	0	52,001,170	14,924,022,386	5,408,008,112	4,264,006,396	0	234,000,351	26,000,039	0	7,800,234	0	

Grouping: Function / Call Stack

Function / Call Stack	Clocktic...	Instructions Retired	CPI Rate	MUX Reliability	Filled Pipeline Slots: Retiring	Filled Pipeline Slots: Bad Speculation	Unfilled Pipeline Slots (Stalls): Back-End Bound	Front-end Bound: Front-End Latency	Front-end Bound: Front-End Bandwidth
grid_intersect	14,118,021,177	12,572,018,858	1.123	0.946	0.246	0.033	0.647	0.063	0.012
sphere_intersect	9,134,013,701	8,494,012,741	1.075	0.965	0.250	0.065	0.619	0.057	0.009
grid_bounds_intersect	1,004,001,506	672,001,008	1.494	0.958	0.227	0.000	0.715	0.104	0.000
__kmp_end_split_barrier	624,000,936	460,000,690	1.357	0.000	0.000	0.000	0.792	0.167	0.042
pos2grid	248,000,372	180,000,270	1.378	0.636	0.367	0.000	0.633	0.000	0.131
shader	242,000,363	142,000,213	1.704	0.860	0.322	0.000	0.946	0.000	0.027
__kmp_x86_pause	224,000,336	122,000,183	1.836	0.000	0.000	0.000	0.971	0.000	0.029
Raypnt	210,000,315	208,000,312	1.010	0.897	0.093	0.279	0.567	0.000	0.062
Selected 1 row(s):	14,118,021,177	12,572,018,858	1.123	0.946	0.246	0.033	0.647	0.063	0.012

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

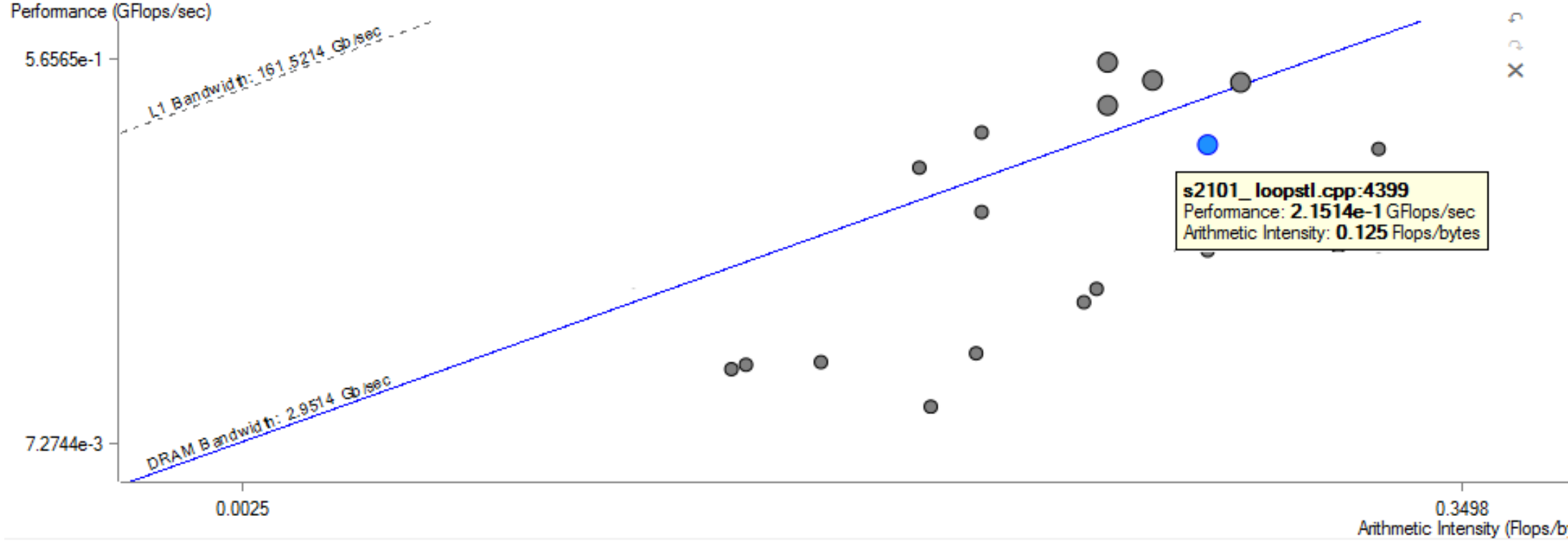
Look for:

- Confirmation
- Surprises

Do not skip either

Roofline Report

in Intel Advisor



Source | Top Down | Loop Assembly | Recommendations | Compiler Diagnostic Details

File: loopstl.cpp:4399 s2101_

Line	Source	Total Time	%	Loop Time	%	Traits
4399	for (i__ = 1; i__ <= i__2; ++i__)	0,031s		1,859s		
4400	aa[i__ + i__ * aa_dim1] += bb[i__ + i__ * bb_dim1] * cc[i__ + i__	1,828s				FMA
4401	* cc_dim1];					

…supplements AI-based analysis with a dynamic FLOP/s profile and peak FLOPs and memory sub-system throughput levels providing enlightening “bounds and bottlenecks” analysis for complex workloads.

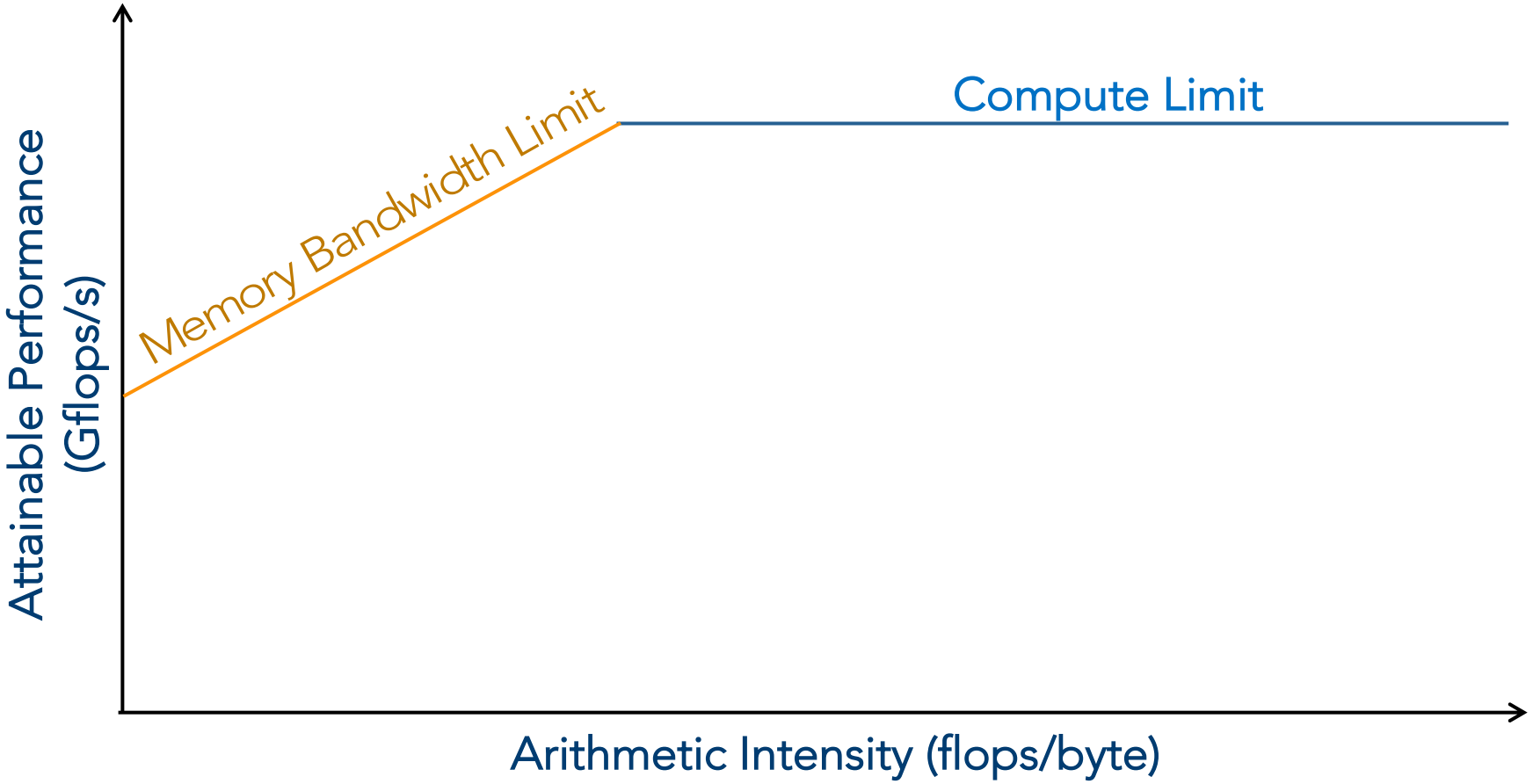
Learn more:

http://tinyurl.com/atpesc-roofline

Argonne NATIONAL LABORATORY

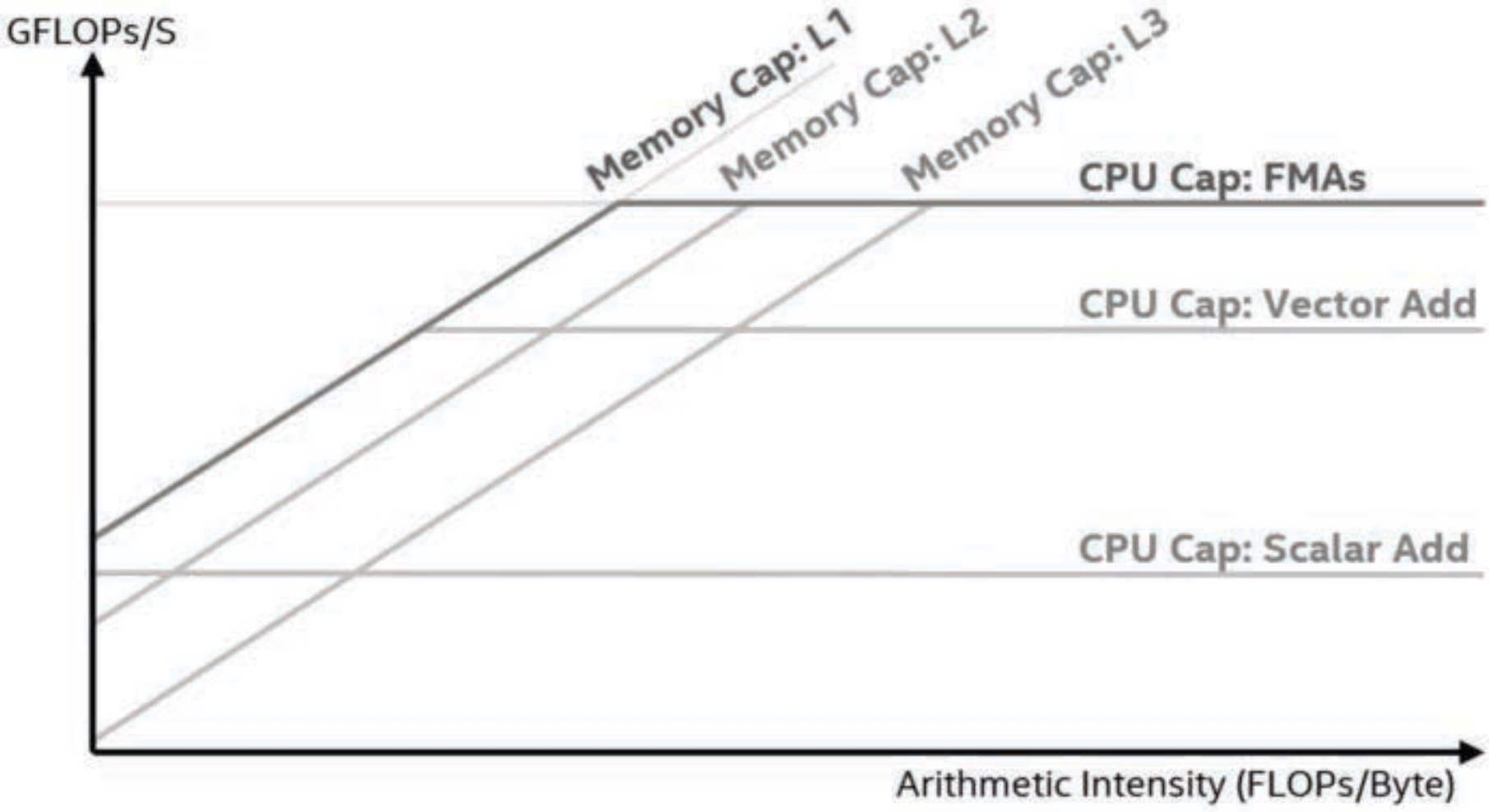
ECP EXASCALE COMPUTING PROJECT

Roofline = Visualize a Performance Model



Learn more:
http://tinyurl.com/atpesc-roofline

There Are Different Ceilings



Learn more:
http://tinyurl.com/atpesc-roofline

Ask “Why am I Here?” and “Where am I going?”

It is always useful to ask the questions:

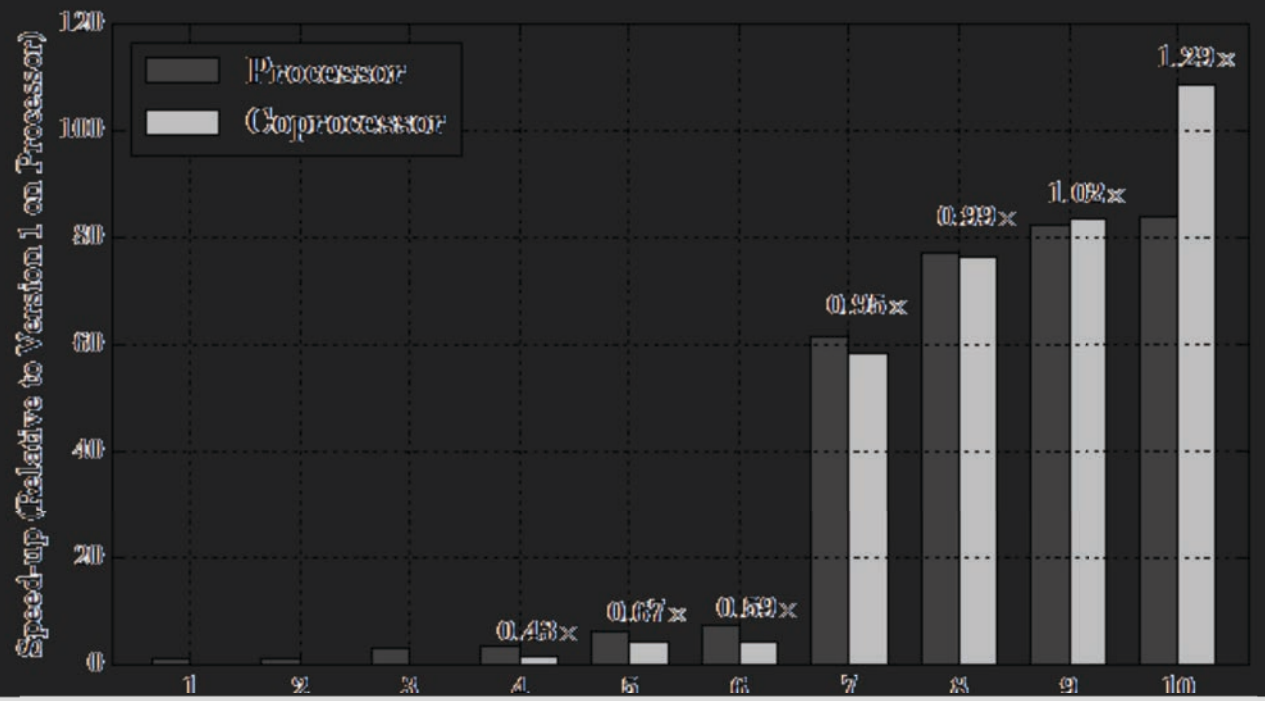
“why am I not on a higher ceiling?” and

“what should I do to reach it?

Learn more:
http://tinyurl.com/atpesc-roofline

LIVE

extremecomputingtraining.anl.gov



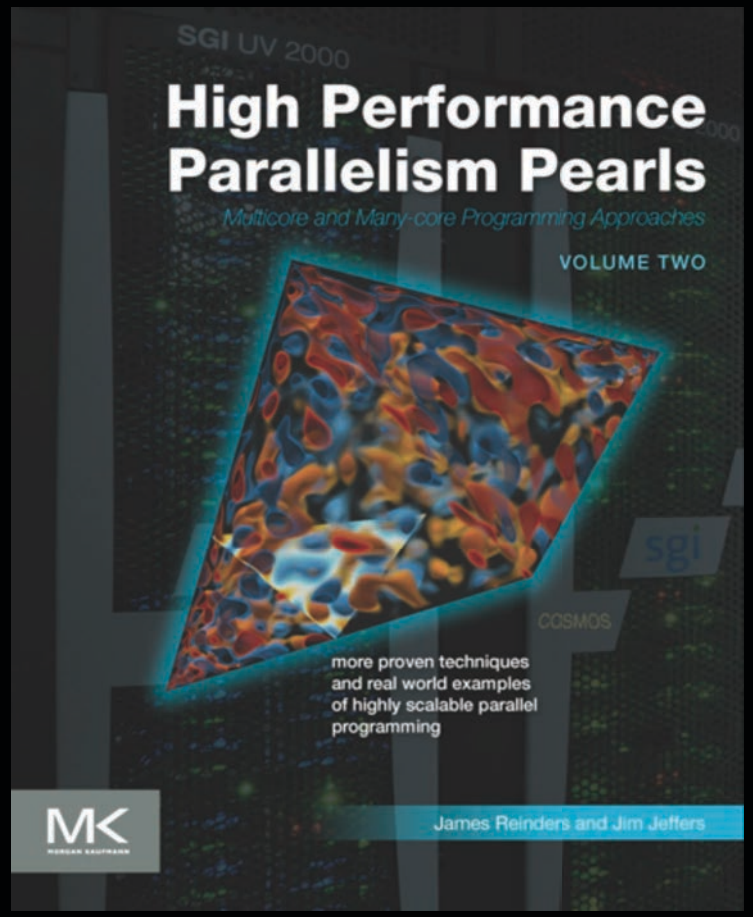
BREAKING NEWS

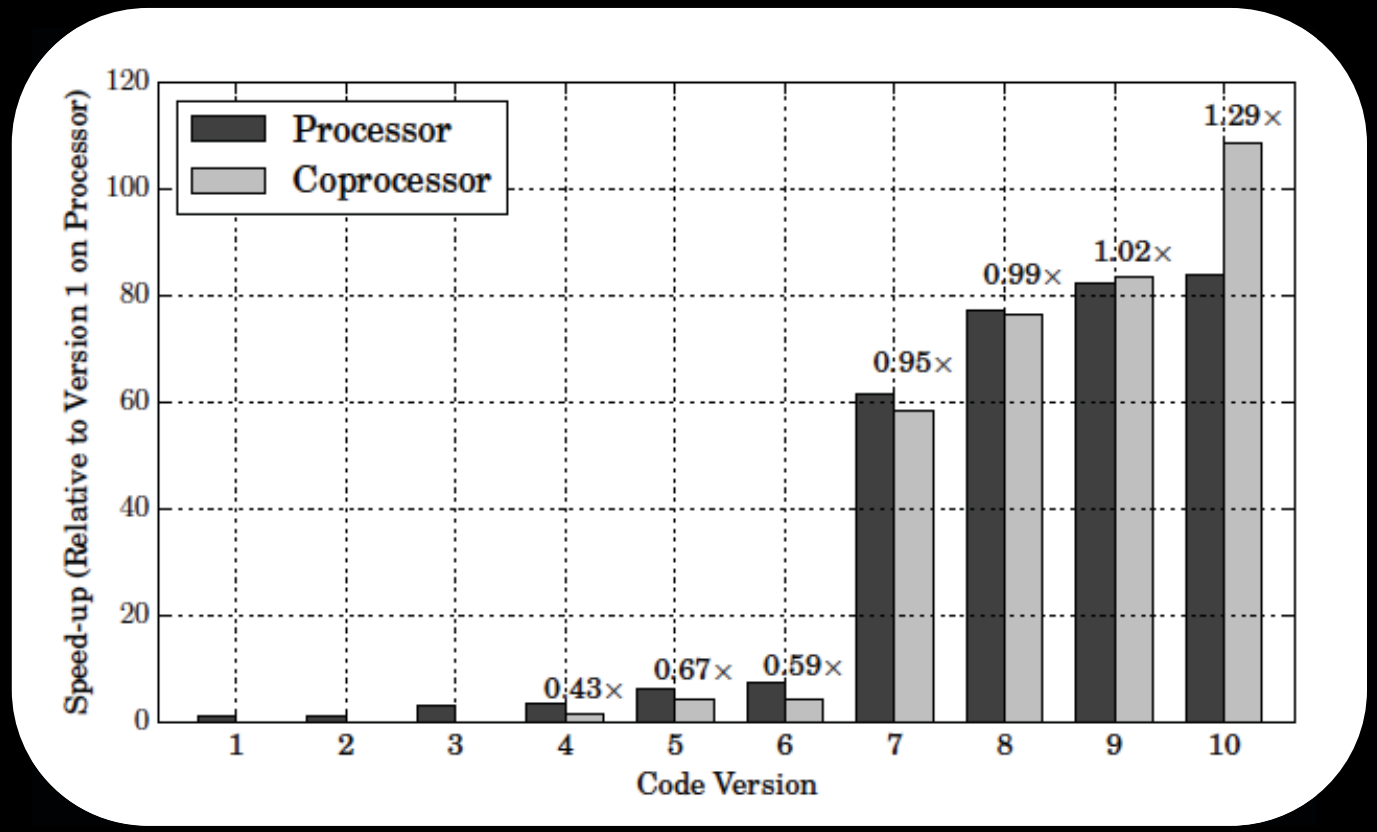
WHAT TOOLS WILL NEVER TELL YOU

23:28

ATPESC STUDENTS BEST EVER! ACCORDING TO SOURCES CLOSE TO CHICAGO

#Moderncode: COSMOS

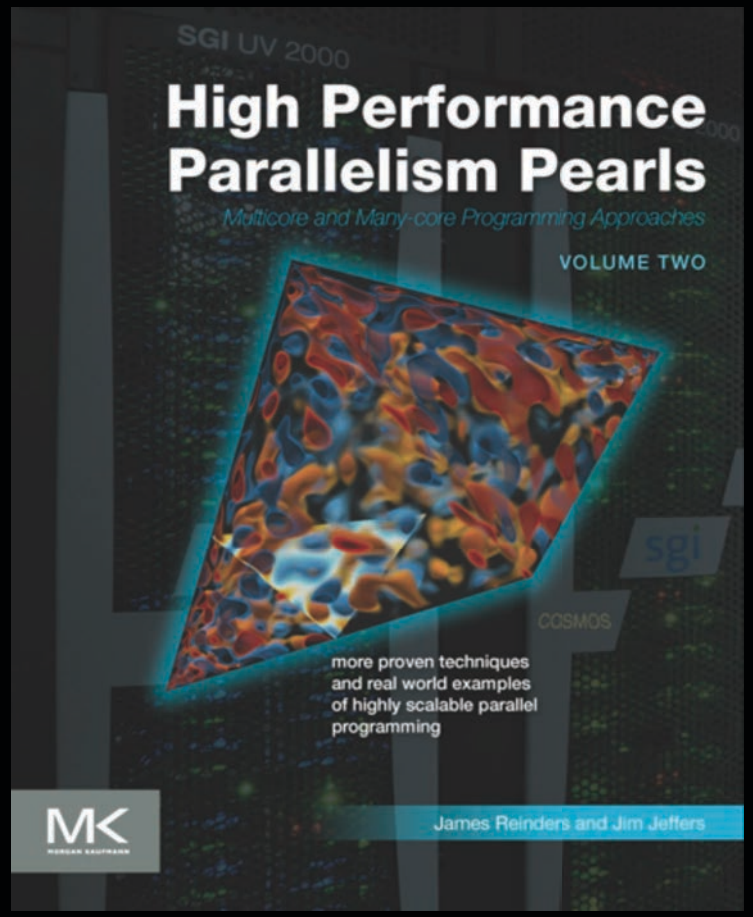


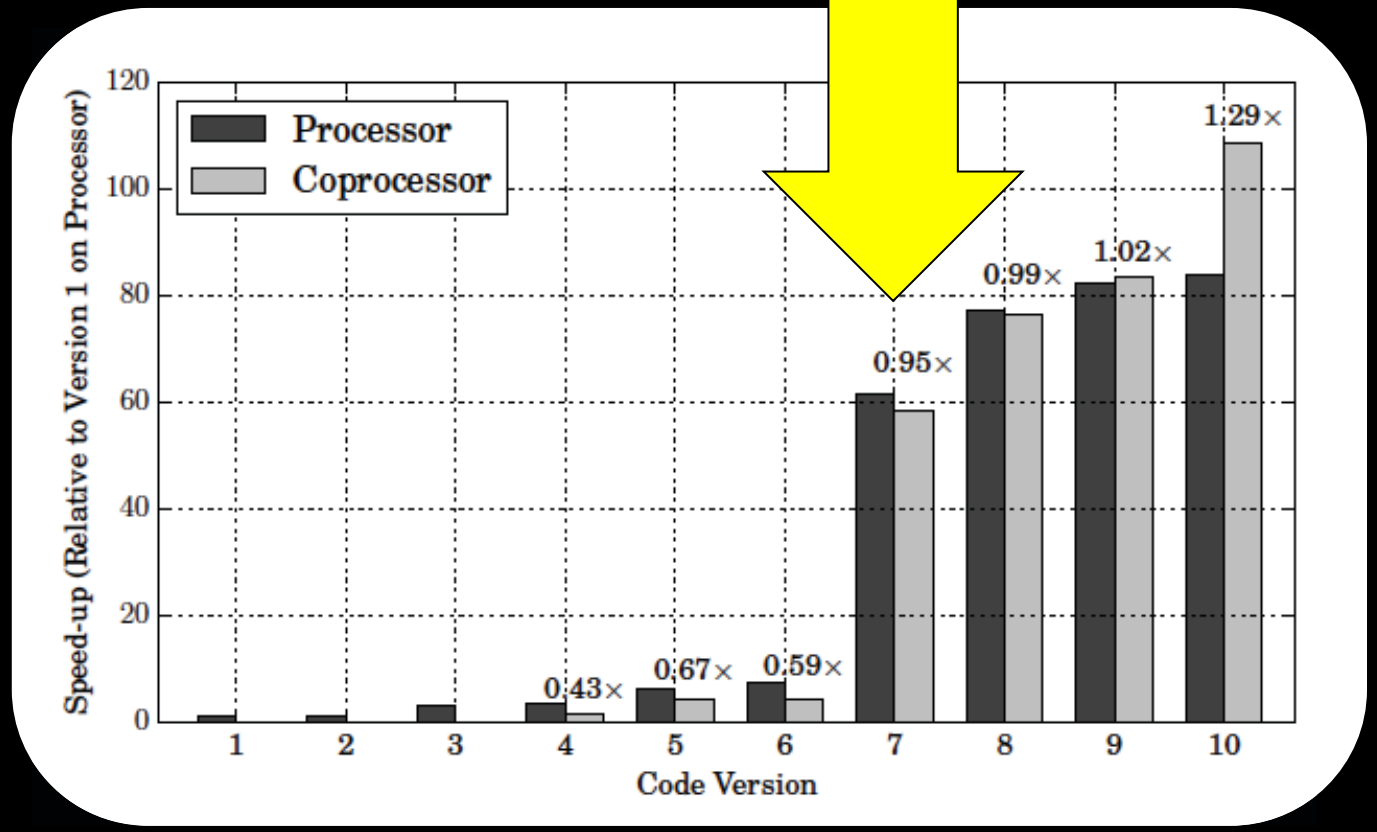


Book Cover Background: Photo of the COSMOS@DiRAC SGI UV2000 based Supercomputer manufactured by SGI, Inc and operated by the Stephen Hawking Centre for Theoretical Cosmology, University of Cambridge. Photo courtesy of Philip Mynott. Book Cover Foreground: 3D visualization of statistical fluctuations in the Cosmic Microwave Background, the remnant of the first measurable light after the Big Bang. CMB data is from the Planck satellite and is the topic of Chapter 10 providing insights into new physics and how the universe evolved. Visualization rendered with Intel's OSPRay ray tracing open source software by Gregory P. Johnson and Timothy Rowley, Intel Corporation.

#Moderncode: COSMOS

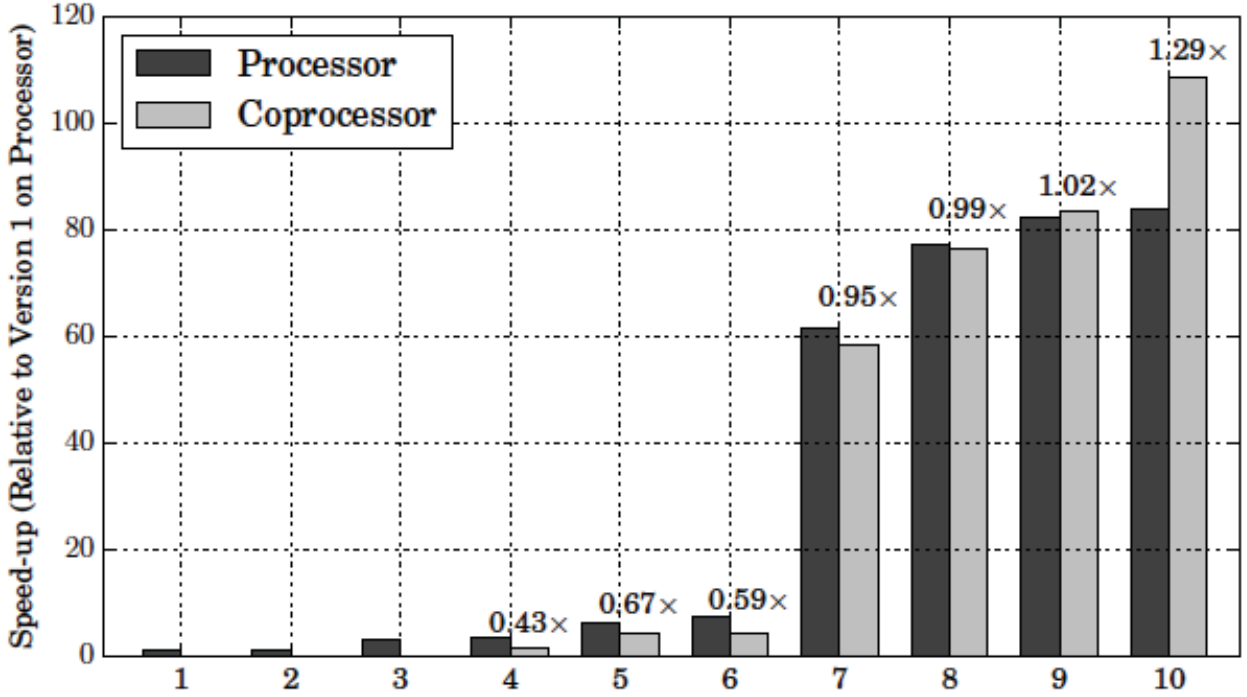
What?





Book Cover Background: Photo of the COSMOS@DiRAC SGI UV2000 based Supercomputer manufactured by SGI, Inc and operated by the Stephen Hawking Centre for Theoretical Cosmology, University of Cambridge. Photo courtesy of Philip Mynott. Book Cover Foreground: 3D visualization of statistical fluctuations in the Cosmic Microwave Background, the remnant of the first measurable light after the Big Bang. CMB data is from the Planck satellite and is the topic of Chapter 10 providing insights into new physics and how the universe evolved. Visualization rendered with Intel's OSPRay ray tracing open source software by Gregory P. Johnson and Timothy Rowley, Intel Corporation.

High Performance Parallelism Pearls
Cosmic Microwave Background Analysis: Nested Parallelism in Practice
Volume 2, Chapter 10



We find that using a simple trapezium rule integrator combined with hand-selected sampling points (to improve accuracy in areas of interest) provides sufficient numerical accuracy to obtain a physically meaningful result, and the reduced space and time requirements of this simplified method give a speed-up of O(10x).

Version	Processor (s)	Coprocessor (s)	Comment
1	2887.0	-	Original code.
2	2610.0	-	Loop simplification.
3	882.0	-	Intel® MKL integration routines and function inlining.
4	865.9	1991.6	Flattened loops and introduced OpenMP threads.
5	450.6	667.9	Loop reordering and manual nested threading.
6	385.6	655.0	Blocked version of the loop (for cache).
7	46.9	49.5	Numerical integration routine (Trapezium Rule).
8	37.4	37.7	Reduction with DGEMM.
9	35.1	34.5	Data alignment (for vectorization).
10	34.3	26.6	Tuning of software prefetching distances.

http://www.intel.com/performance

LIVE

extremecomputingtraining.anl.gov

KEEP CALM AND MIND YOUR ALGORITHMS

BREAKING NEWS

WHAT TOOLS WILL NEVER TELL YOU

23:28

ATPESC STUDENTS BEST EVER! ACCORDING TO SOURCES CLOSE TO CHICAGO

surprised
verify
hunt
follow-up
tools
trust
tuning
use
back
expectations
methodology
validate
prepare
wag
performance
roofline
sure
preferred
follow
surprises
make
let
discrepancies
envelope
model

KEEP
CALM
AND
THINK
PARALLEL

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT

Performance Tuning: How to Prepare to be Surprised, and then… How to Hunt for the Surprises

- **Have Expectations**
  - make sure you *can* be surprised
- **Validate your expectations**
  - follow-up on discrepancies
- **Follow a methodology**
  - Use tools but don't let them wag you

Have a Roofline Model
(back of envelope preferred)

Learn more:
http://tinyurl.com/atpesc-roofline

Argonne NATIONAL LABORATORY

ECP EXASCALE COMPUTING PROJECT