Drive Extreme Concurrency for Data Center Software

Software and Services Group

Tony Wu

October 17, 2017

4:34 PM
100%
发现
极客新闻
对落地DevOps理念的一些反思
2017年Elm语言现状调查结果出炉
陆奇：百度的人工智能战略
Java依旧排名第一，然而变化几何?
专栏订阅
专栏
重拾极客精神
提升技术认知

主办方 Geekbang> 极客邦科技 InfoQ

ArchSummit

全球架构师峰会 2017

12月8-9日 北京 · 国际会议中心

APSEC 2017

南京大學 NANJING UNIVERSITY × Geekbang 极客邦科技 | InfoQ

APSEC 2017

24th Asia-Pacific Software Engineering Conference
4-8 December 2017, Nanjing, Jiangsu, China

12月4–8日
中国南京

APSEC
了解详情

AiCon

全球人工智能技术大会 2018

助力人工智能落地

2018.1.13 - 1.14 北京国际会议中心

扫描关注大会官网

About the speaker

Vehicle Design & Dynamics

Control & Autonomous Navigation

Computer Architecture
& Distributed Computing

Joined Intel at Silicon Valley

Software Optimization for
Performance & Scalability

Legal Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks

Cost reduction scenarios described are intended as examples of how a given Intel- based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

No computer system can be absolutely secure.

Intel, the Intel logo, Xeon, Intel vPro, Intel Xeon Phi, Look Inside., are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Microsoft, Windows, and the Windows logo are trademarks, or registered trademarks of Microsoft Corporation in the United States and/or other countries.

Optimization Notice

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel.

Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Agenda

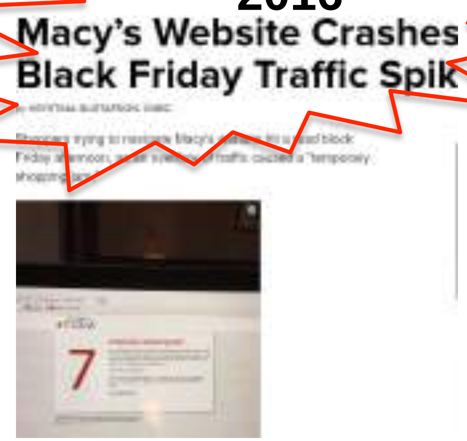
- Introduction to Performance & Scalability
- Methodology
- Case study
- Summary

Software Performance & Scalability

- Performance: an indicator of how well a software application meets its requirements for timeliness
  - Response time: how long it take to respond to a request
  - Throughput: how many requests that can be processed per unit of time
- Scalability: the ability of a system to continue to meet its objective (response time/throughput) as the load increases
  - Extremely important to maintain the responsiveness of a datacenter application as more and more users converge on a site

Why Performance & Scalability Matters?

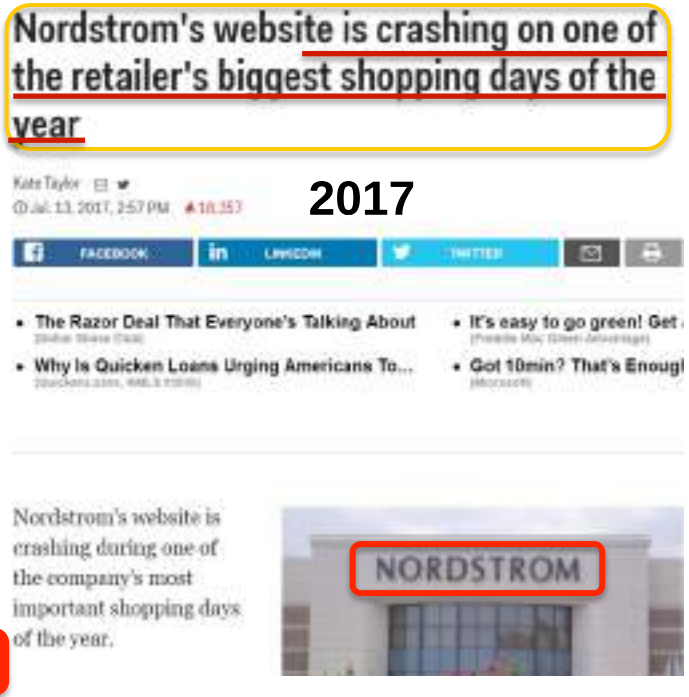
2016



Macy's Website Crashes Black Friday Traffic Spik

[illegible]

Shoppers trying to navigate Macy's website hit a road block Friday afternoon, as a spike in traffic caused a "temporary shopping lag."



Nordstrom's website is crashing on one of the retailer's biggest shopping days of the year

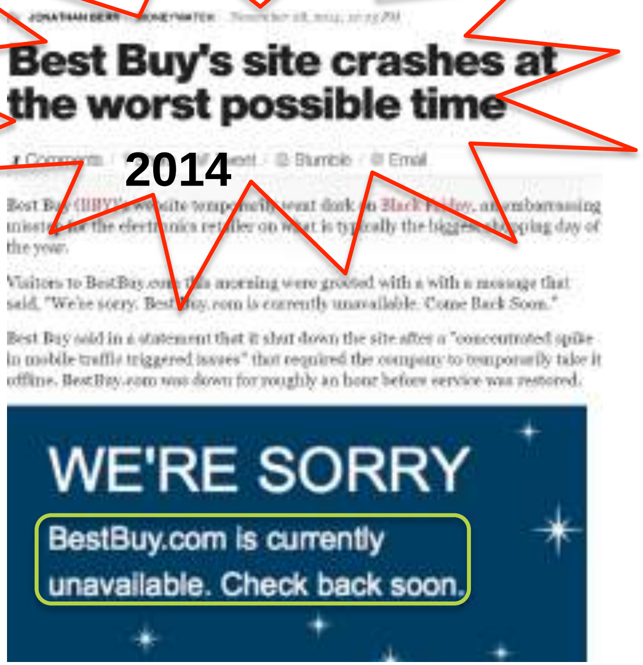
Kate Taylor

Jul. 13, 2017, 2:57 PM ▲10,357

- The Razor Deal That Everyone's Talking About
- It's easy to go green! Get
- Why Is Quicken Loans Urging Americans To...
- Got 10min? That's Enoug

2017

Nordstrom's website is crashing during one of the company's most important shopping days of the year.



By JONATHAN BERR MONEYWATCH November 28, 2014, [illegible]

Best Buy's site crashes at the worst possible time

Comments | [illegible] | Stumble | Email

2014

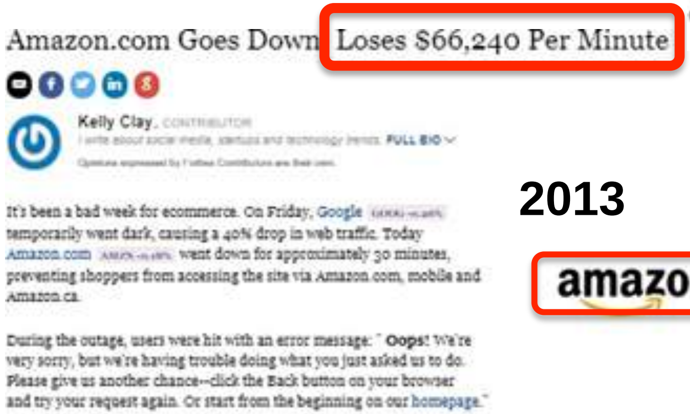
Best Buy (BBY)'s website temporarily went dark on Black Friday, an embarrassing misstep for the electronics retailer on what is typically the biggest shopping day of the year.

Visitors to BestBuy.com this morning were greeted with a with a message that said, "We're sorry. BestBuy.com is currently unavailable. Come Back Soon."

Best Buy said in a statement that it shut down the site after a "concentrated spike in mobile traffic triggered issues" that required the company to temporarily take it offline. BestBuy.com was down for roughly an hour before service was restored.

WE'RE SORRY

BestBuy.com is currently unavailable. Check back soon.



Amazon.com Goes Down, Loses $66,240 Per Minute

Kelly Clay, CONTRIBUTOR

I write about social media, startups and technology trends. FULL BIO

Opinions expressed by Forbes Contributors are their own.

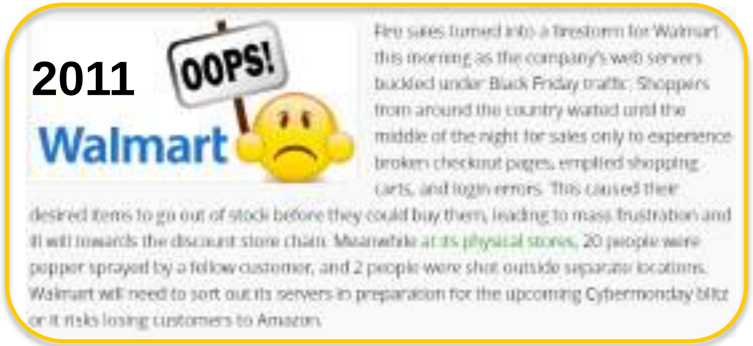
It's been a bad week for ecommerce. On Friday, Google [illegible] temporarily went dark, causing a 40% drop in web traffic. Today Amazon.com [illegible] went down for approximately 30 minutes, preventing shoppers from accessing the site via Amazon.com, mobile and Amazon.ca.

During the outage, users were hit with an error message: " Oops! We're very sorry, but we're having trouble doing what you just asked us to do. Please give us another chance--click the Back button on your browser and try your request again. Or start from the beginning on our homepage."

2013

amazon.com

2011



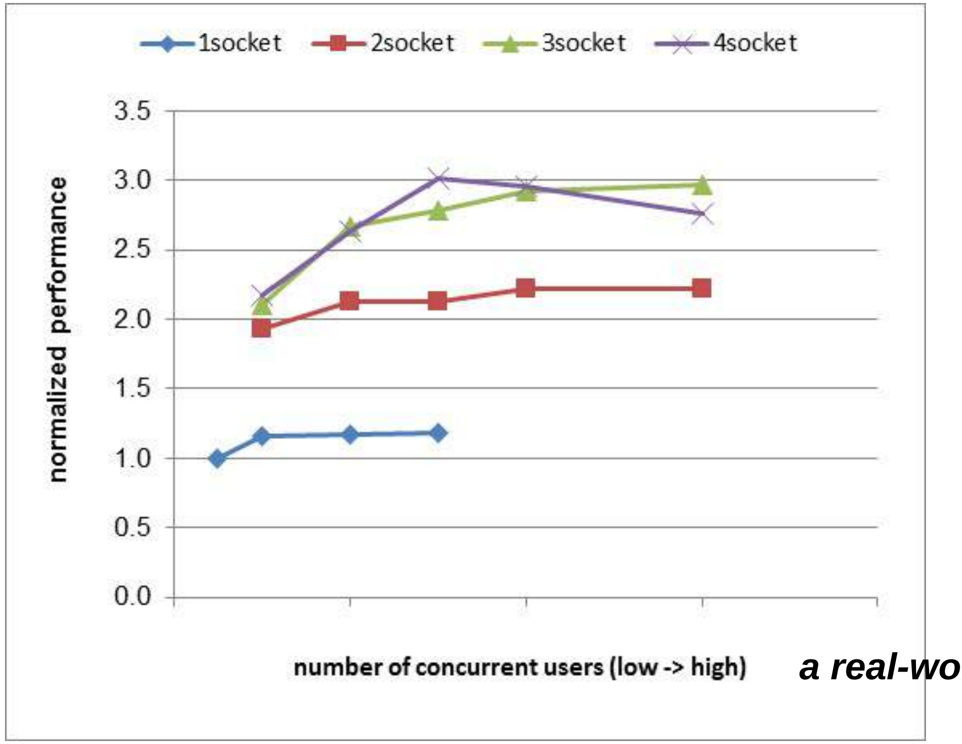
Fire sales turned into a firestorm for Walmart this morning as the company's web servers buckled under Black Friday traffic. Shoppers from around the country waited until the middle of the night for sales only to experience broken checkout pages, emptied shopping carts, and login errors. This caused their desired items to go out of stock before they could buy them, leading to mass frustration and ill will towards the discount store chain. Meanwhile at its physical stores, 20 people were pepper sprayed by a fellow customer, and 2 people were shot outside separate locations. Walmart will need to sort out its servers in preparation for the upcoming Cybermonday blitz or it risks losing customers to Amazon.

Performance & Scalability on Multicore

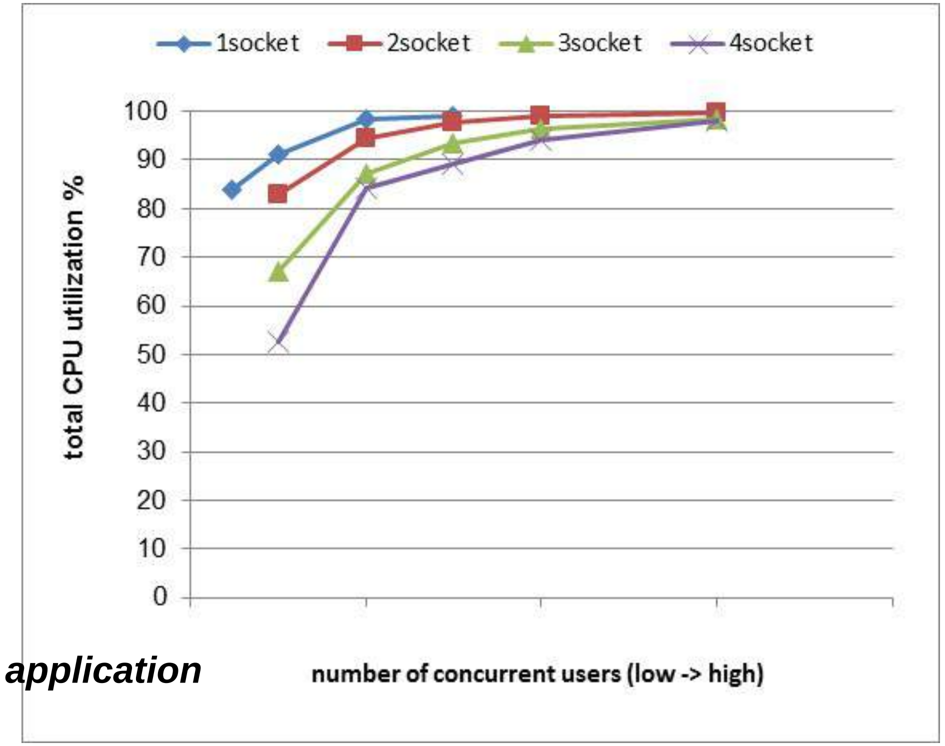
- Concurrency is being able to run multiple tasks in parallel, which can increase the efficiency of an application
- Some problems are parallelism friendly
  - If 1 painter takes 10 hours, 10 painters take 1 hour
- Some aren’t…
  - If 1 boat crosses river in 10 days, 10 boats cross in 10 days
  - But you get 10 boats every 10 days
    - If you pipeline you can get 1 boat per day
  - Bandwidth increase, no latency drop
- May need new algorithms

To achieve high Concurrency is the key

Concurrency Challenge



a real-world enterprise application

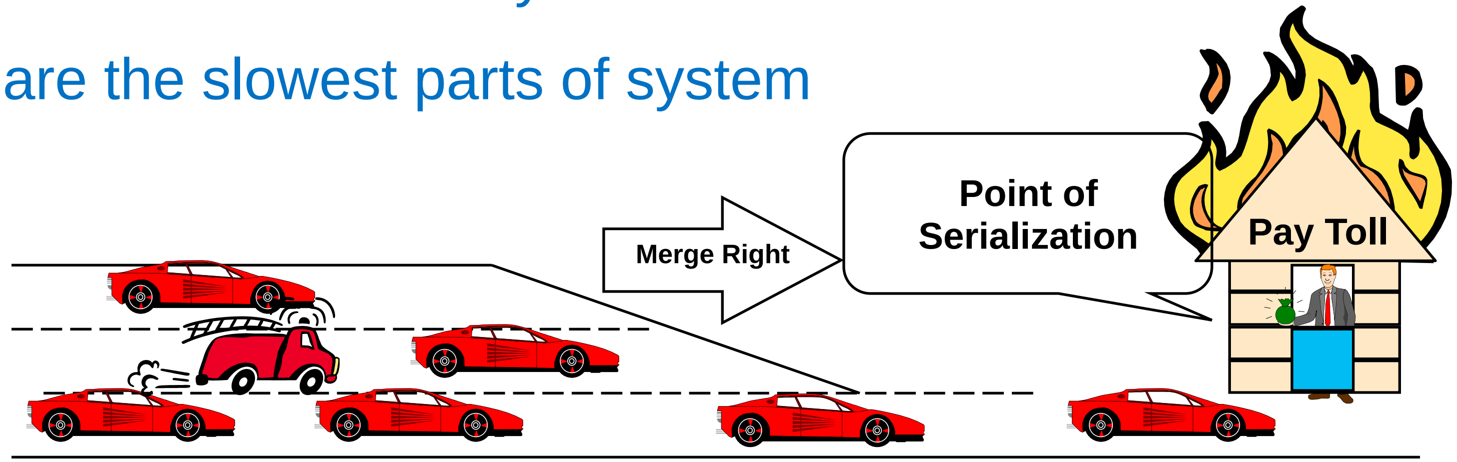


- More and more cores are added into a single system
- more cores = better performance ?
  - multi-threaded code is difficult to write and difficult to test
  - Even multi-threaded enterprise applications do not *automatically* run faster on multi-core servers

Software must be optimized in order to take fully advantage of multicore

How to Increase Concurrency?

- Identify performance & scalability bottlenecks
- Bottlenecks are the slowest parts of system

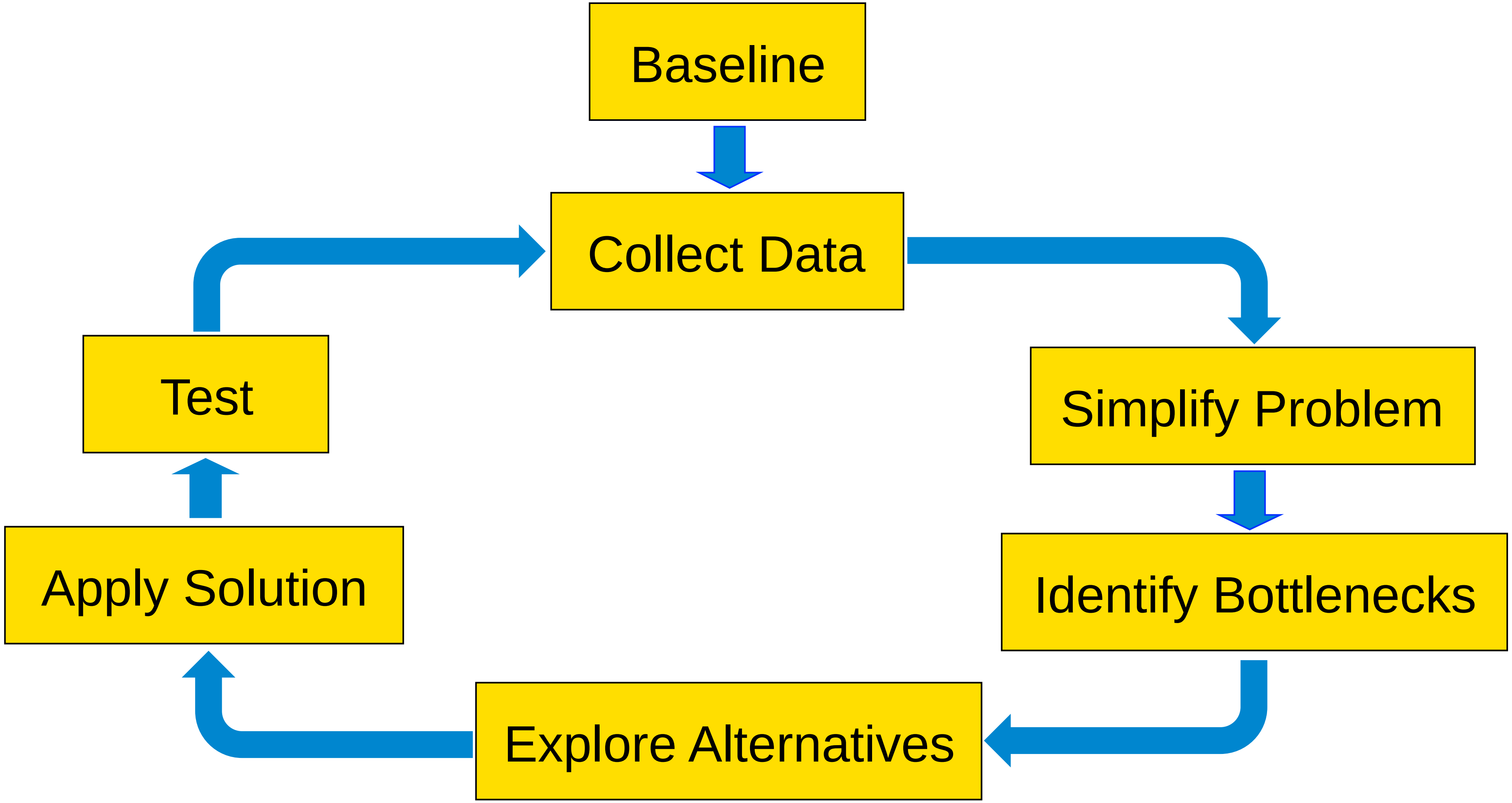


- Points of serialization exist when work must wait for other work to be finished
- The bottleneck eventually determines how much work a system can do per unit time
- The primary bottleneck determines the maximum throughput in a system

Agenda

- Introduction to Performance & Scalability
- Methodology
- Case study
- Summary

Methodology Overview



Baseline

Collect Data

Simplify Problem

Identify Bottlenecks

Explore Alternatives

Apply Solution

Test

Understand the Workload

- A workload reproduces typical stress on a system
  - for individual component
  - for end-to-end system
- A good workload exhibits these characteristics:
  - Measurable: A performance metric exists and can be quantified
  - Reproducible: The measurement is repeatable and consistent
  - Static: The measurement does not vary with time
  - Representative: The work being performed is typical of the stress put on the system under normal operating conditions

Use the Right Tools

- Large variety of tools available to collect data
- Intel® VTune, perf, oprofile tools
  - Powerful tools: maps processor events to source code
- Linux related
  - vmstat, mpstat, sar, iostat, lockstat, strace
- JVM related
  - Java Mission Control, Java Flight Recorder, jcom, Jconsole, ... https://docs.oracle.com/javase/8/docs/technotes/guides/troubleshoot/tooldescr025.html
  - Garbage Collection (GC log)
- Application specific
  - Example: Automatic Workload Repository (AWR) in Oracle Database

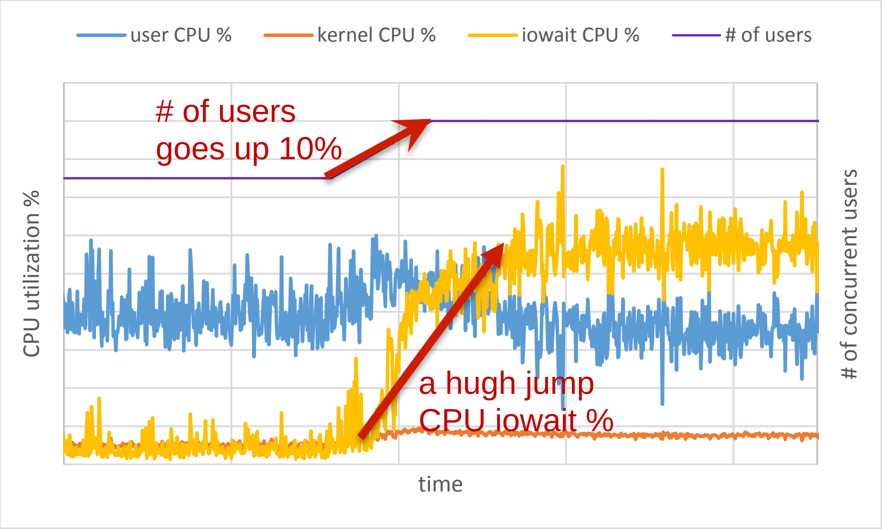
Follow the Top-Down Data-driven Technique

- ❑ System-level
  - CPU
  - Memory
  - I/O
  - Network Usage
  - Context Switch Rate
  - …
- ❑ Software-level
  - Application
  - Process
  - Module
  - Function
  - Instruction
- ❑ Microarchitecture-level
  - Frontend Bound
  - Bad Speculation
  - Backend Bound
  - Retiring
  - Cache optimization
  - …

- Let the results of one iteration direct the next
- Backup and document your data completely and consistently
  - Example: CPU info, BIOS configuration, OS build and customizations, compiler drop and options, software version, environment changes
- Automate whenever possible
  - allow precise repetition of process
  - remove tedium of single steps

An Example: to Simplify the Problem

- A highly complicated real-world workload running on the latest Intel Xeon Server
  - Hundreds of processes running concurrently
  - Many JVM instances of application servers
  - Several database instances
- 50% performance drops when concurrent user # exceeds certain threshold
  - iowait CPU time suddenly jumps to >50%



# of users goes up 10%

a hugh jump CPU iowait %

Processes that are waiting for I/O are commonly in an "uninterruptible sleep" state or "D“

```
for x in `seq 1 1 10`; do ps -eo state,pid,cmd | grep "^D"; echo "----"; sleep 5; done
-----
D  84587 xyz_abc9200 (LOCAL=NO)
D  85002 xyz_abc9200 (LOCAL=NO)
D  85811 xyz_abc9200 (LOCAL=NO)
……
D  11460 xyz_lgwr_abc9200
D  88163 xyz_abc9200 (LOCAL=NO)
D  93066 xyz_abc9200 (LOCAL=NO)
-----
```

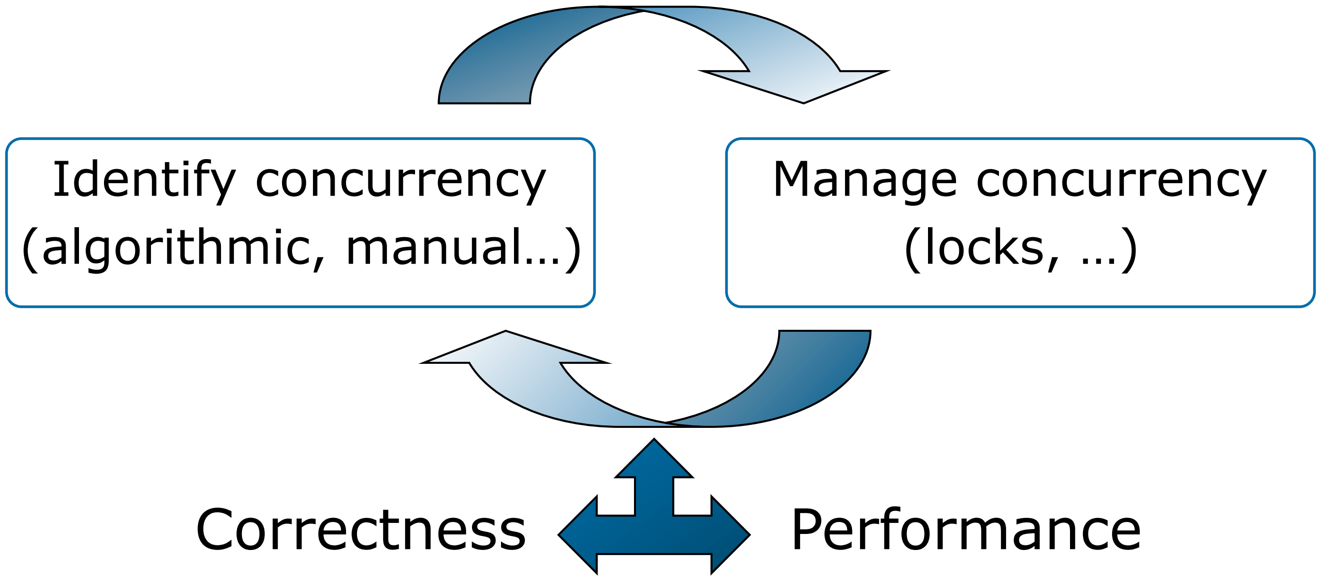
xyz_abc application needs more investigation, as IO Waiting processes are from xyz_abc

Reduce the problem set by identifying problematic application or process

Agenda

- Introduction to Performance & Scalability
- Methodology
- Case study
- Summary

Case 1: Lock Granularity



Identify concurrency (algorithmic, manual…)

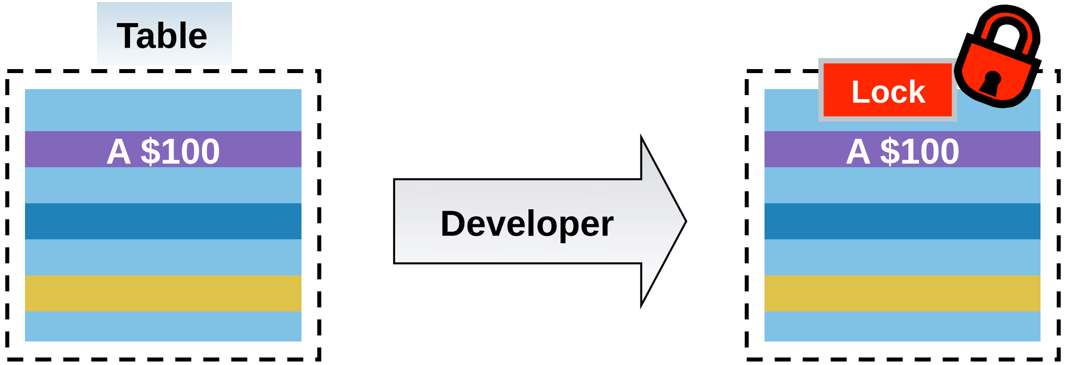
Manage concurrency (locks, …)

Correctness

Performance

Hard to Write Fast and Correct Multi-Threaded Code

Need for Synchronization



Sophie wants $50 from A

- A == $100, A set to $50

Peter wants $60 from A

- A == $100, A set to $40

A should be -10

Sophie wants $50 from A

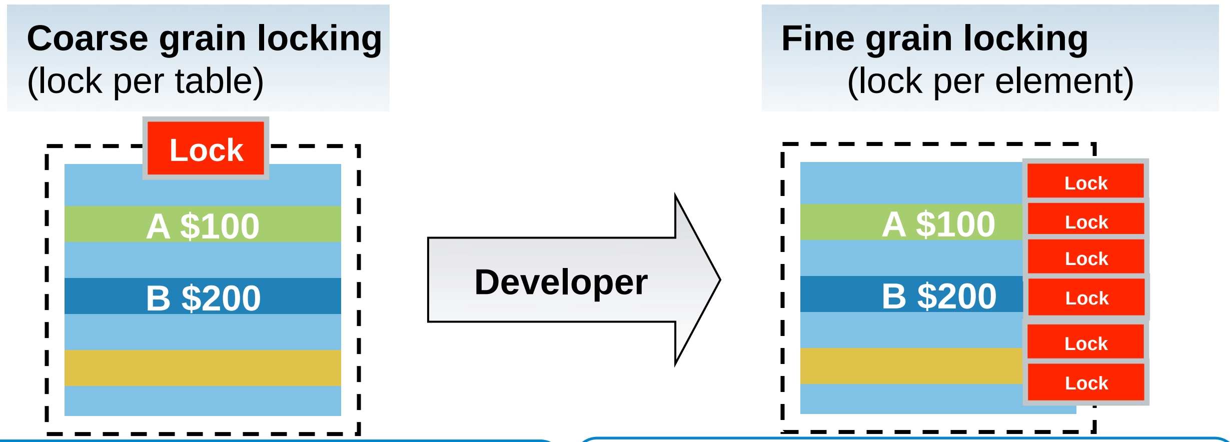
- Sophie locks table
- A == $100, A set to $50

Peter wants $60 from A

- Peter waits till lock release
- A == $50, Insufficient funds

Peter and Sophie saw A == $100. Locks prevent such data races

Lock Granularity Optimization



Coarse grain locking
(lock per table)

Fine grain locking
(lock per element)

Sophie withdraws $20 from A

- Sophie locks table

Peter wants $30 from B

- Waits for Sophie to free table

Sophie withdraws $20 from A

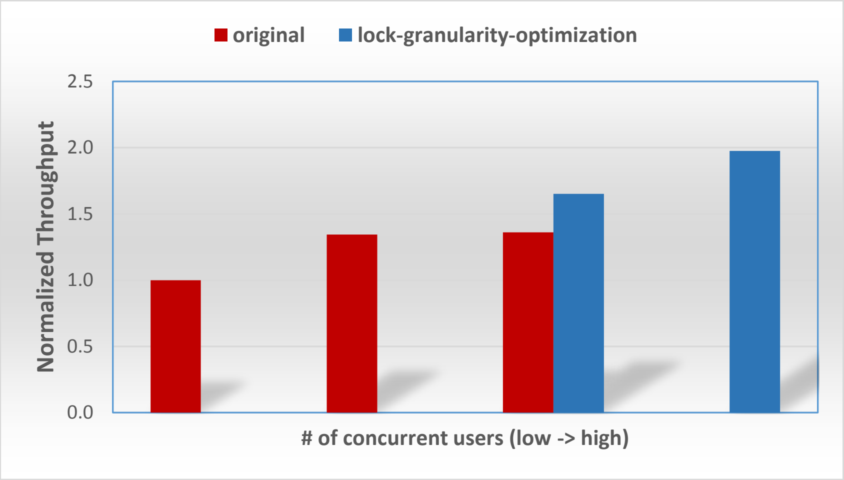
- Sophie locks A

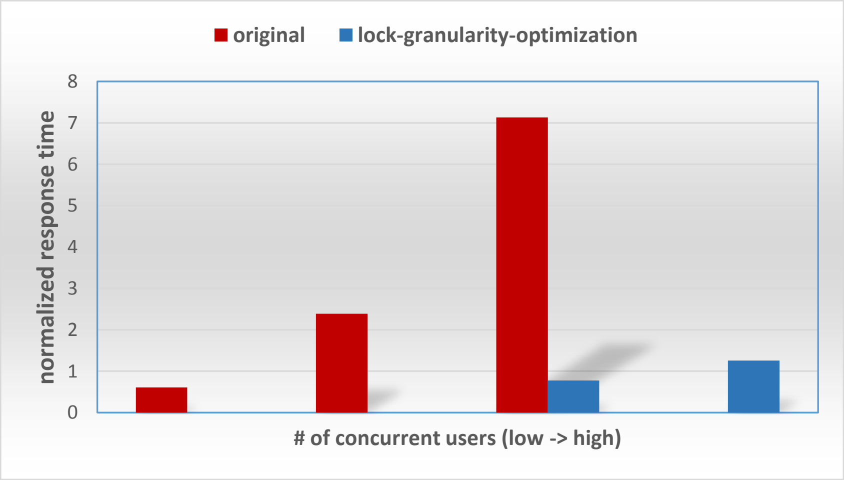
Peter wants $30 from B

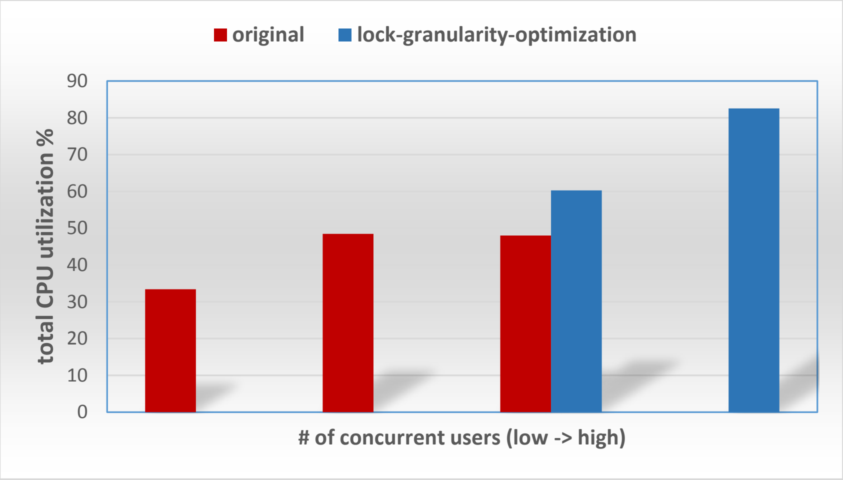
- Peter locks B

Such Tuning is Time Consuming and Error Prone

A real-world Enterprise Application



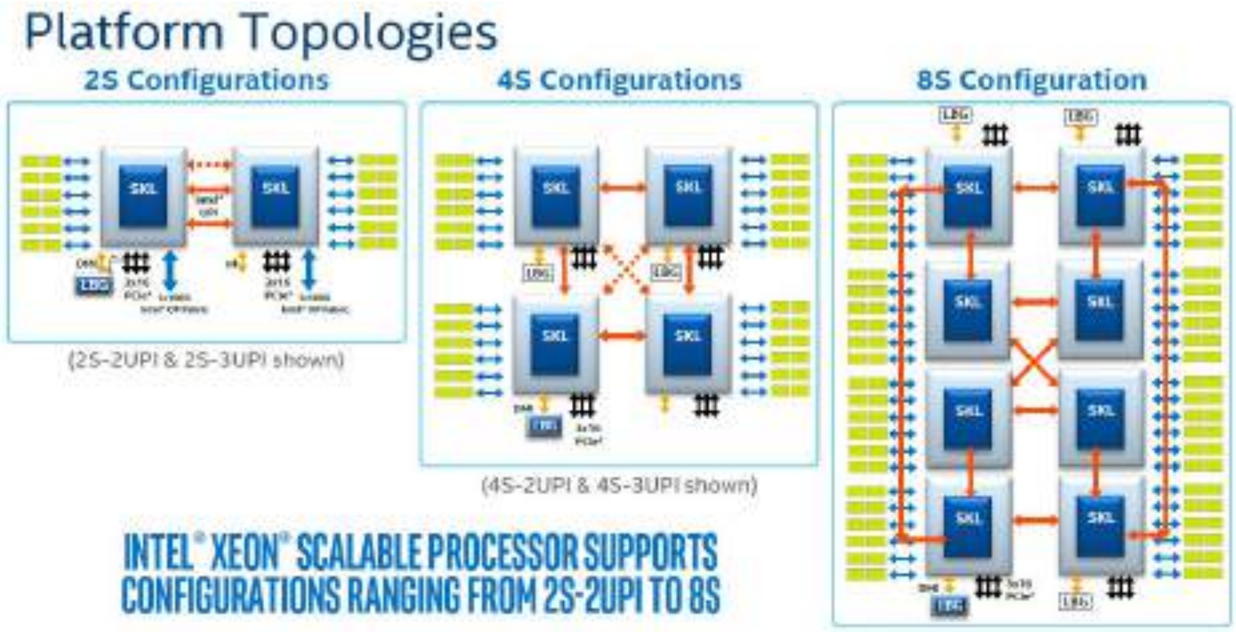




Lock granularity optimization leads to +50% performance gain

Case 2: Lock Locality

- Most existing state-of-the-art server platforms are NUMA-based (Non-Uniform Memory Access)
- As # of sockets increase, remote latency increases nonlinearly
- Monitor # of HITM event when load or # of sockets varies
    - HITM occurs when the snooped address at the responder's cache is in a Modified state


Platform Topologies
2S Configurations
4S Configurations
8S Configuration
SKL
LBG
(2S-2UPI & 2S-3UPI shown)
(4S-2UPI & 4S-3UPI shown)
INTEL® XEON® SCALABLE PROCESSOR SUPPORTS
CONFIGURATIONS RANGING FROM 2S-2UPI TO 8S

Locality becomes critical in large machines

Solution: Lock Affinity

Before optimization

```
readLockProctection
(*(myLock[++myIndex & lockCountMinus1].Get()), …);
```

- Totally N readLock
- A readLock is pseudo-randomly assigned
- <++myIndex & lockCountMinus1> is how the N locks are distributed
  - <myIndex> is a global variable
  - <lockCountMinus1> is set as (N-1)

Optimized with Lock Affinity

```
readLockProctection (myLockType *lock, int mask, bool enabled, …)
  {
    int index;
……
    index  = apicid(); //get the local CPU id
      index &= mask;
……
    myLock = myLock[index].Get();
    myLock->AcquireRead();
  }
```

For a real-world enterprise application

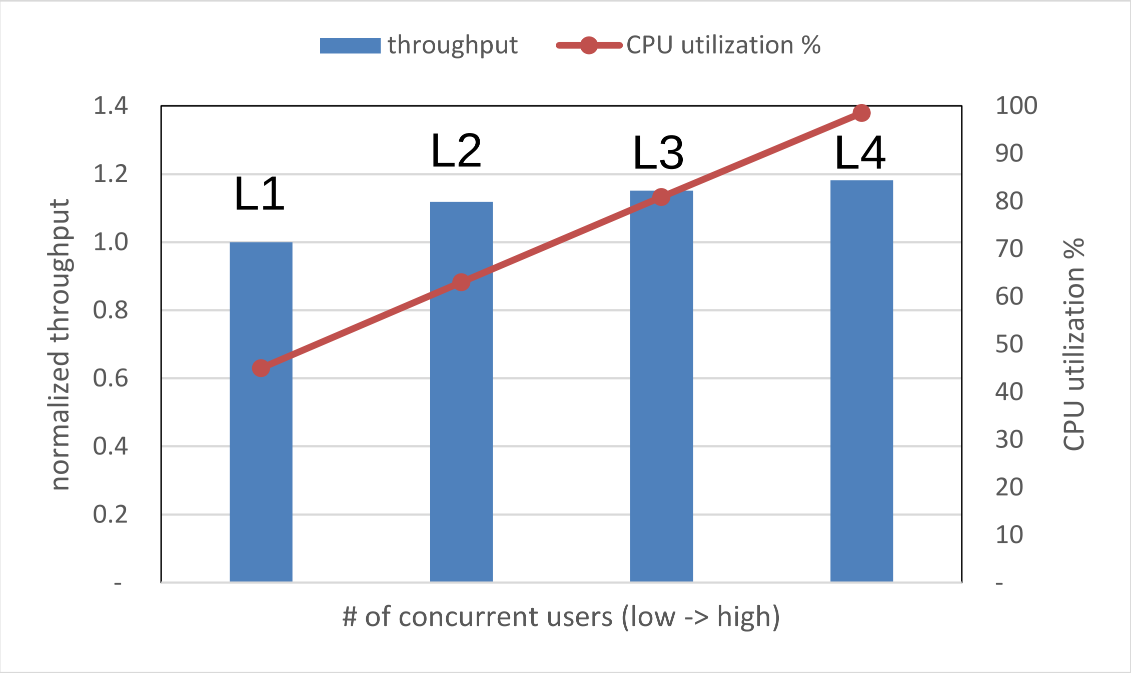
- 20% performance gain on a 4-socket Xeon platform
- 2x performance gain on a 8-socket Xeon platform

Let The CPU Handle the Locks

- **Intel® Transactional Synchronization Extensions (TSX): Instruction set extensions for IA**
  - Transactionally execute lock-protected critical sections
  - Execute without acquiring lock → expose hidden concurrency
  - Hardware manages transactional updates – All or None
- **Hardware does the work of figuring out concurrency**
  - No worry about fine granular locking
  - No worry about lock locality / affinity
- **Intel® Architecture Instruction Set Extensions Programming Reference**
  - https://software.intel.com/sites/default/files/m/9/2/3/41604

Intel TSX make Parallel Programming Easier and Faster

Case 3: Pick right hardware

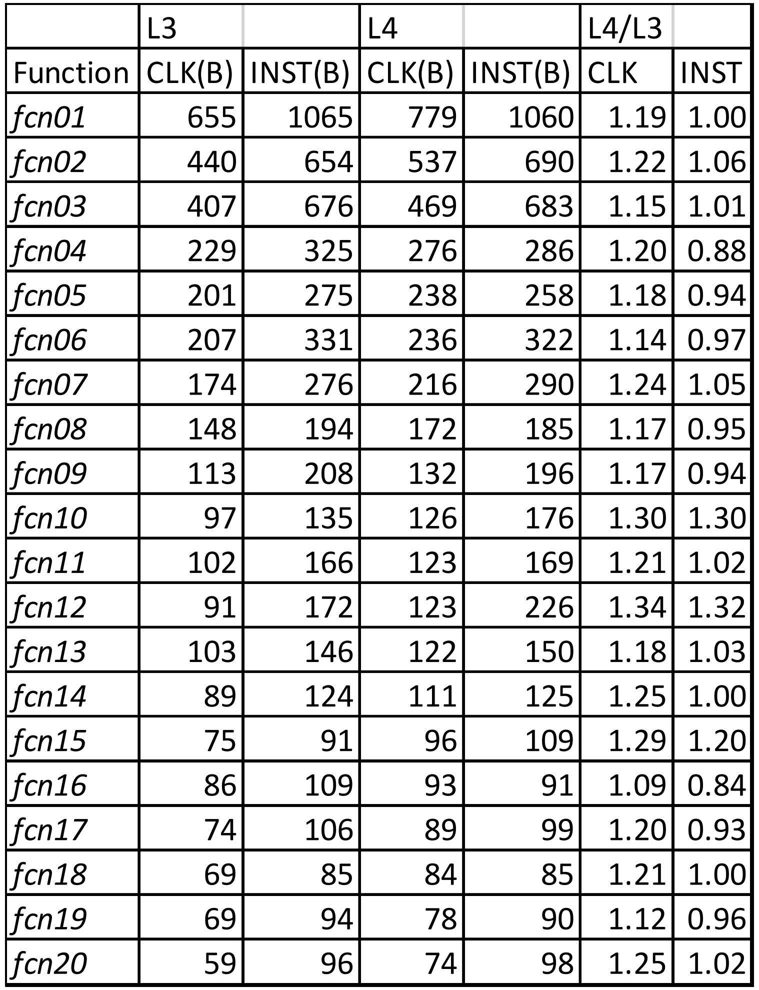


Running a real-world enterprise application on a 2-socket Xeon with 2-UPI

- Load increases: L1→L2
  - throughput increases 12%
  - total CPU utilization increases 40%
- Load increases: L2→L3
  - throughput increases 3% only
  - but total CPU utilization increases 30%
- Load increases: L3→L4
  - throughput increases 3% only
  - but total CPU utilization increases 25%

Why scales poorly?

Case 3: Pick right hardware (2)

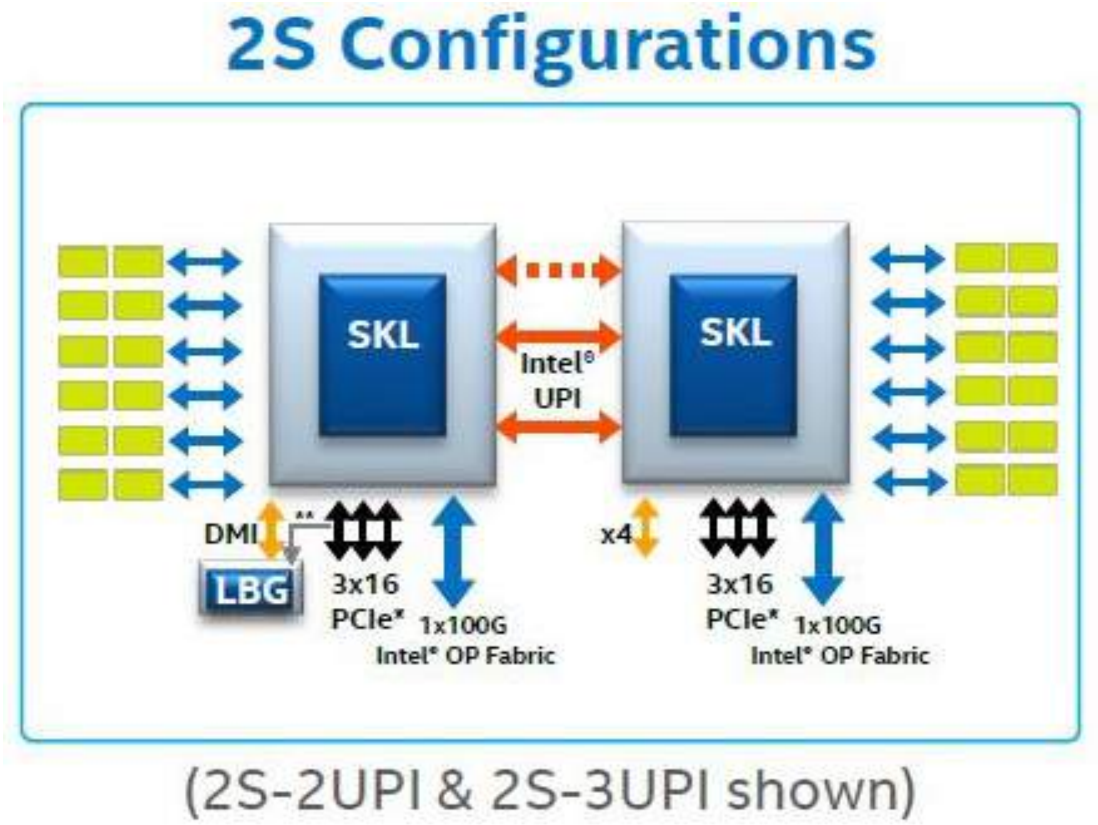


	L3		L4		L4/L3	
Function	CLK(B)	INST(B)	CLK(B)	INST(B)	CLK	INST
fcn01	655	1065	779	1060	1.19	1.00
fcn02	440	654	537	690	1.22	1.06
fcn03	407	676	469	683	1.15	1.01
fcn04	229	325	276	286	1.20	0.88
fcn05	201	275	238	258	1.18	0.94
fcn06	207	331	236	322	1.14	0.97
fcn07	174	276	216	290	1.24	1.05
fcn08	148	194	172	185	1.17	0.95
fcn09	113	208	132	196	1.17	0.94
fcn10	97	135	126	176	1.30	1.30
fcn11	102	166	123	169	1.21	1.02
fcn12	91	172	123	226	1.34	1.32
fcn13	103	146	122	150	1.18	1.03
fcn14	89	124	111	125	1.25	1.00
fcn15	75	91	96	109	1.29	1.20
fcn16	86	109	93	91	1.09	0.84
fcn17	74	106	89	99	1.20	0.93
fcn18	69	85	84	85	1.21	1.00
fcn19	69	94	78	90	1.12	0.96
fcn20	59	96	74	98	1.25	1.02

- CLK (CPU cycle) and INST (instruction retired) are normalized by throughput
- For perfect scaling, the scaling ratio (i.e. L4/L3) should close to 1
- Top 20 functions account for ~70% of total CPU utilization
- From L3→L4, CPU cycles per transaction increase similarly for all functions

No single function stands out as the load increases

Case 3: Pick right hardware (3)



- The hardware is a 2S Xeon with only 2-UPI
- UPI: Intel® Ultra Path Interconnect
    - a coherent interconnect for scalable systems containing multiple processors
- As load increases, UPI Data Transmit bandwidth utilization goes up quickly and close to saturation

2S Xeon with 3-UPI improves the scalability

Agenda

- Introduction to Performance & Scalability
- Methodology
- Case study
- Summary

Summary

- Use your experience and intuition
  - Grow your expertise on concurrency and synchronization
- Understand your application and workload
  - How resource changes impact workload
  - How workload changes impact resources
- Follow a precise process
  - Top-down and data-driven
- Develop your knowledge on hardware
  - Choose hardware wisely

intel®

experience
what's inside™

© 2017 Intel Corporation

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.
*Other names and brands may be claimed as the property of others.

Java is a registered trademark of Oracle and/or its affiliates.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors.

Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.   For more complete information visit http://www.intel.com/performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.