Effective Performance Engineering at Twitter-Scale

IOP @ Twitter (2017.4 - 2022.11)

Yao Yue, IOP Systems

Performance Engineering...

"On the second night, he asked for a ladder, climbed up the generator and made a chalk mark on its side."

"They did, and the generator performed to perfection."

Charles Steinmetz, aka "Wizard of Schenectady"

(Steinmetz, itemizing the bill at the request of Henry Ford)

```
Making chalk mark on generator $1.

Knowing where to make mark $9,999.
```

Performance Engineering

Wizardry

Who I am

- Maintained & developed multiple distributed caching services
- Trained by a decade of cache incidents, years of operations, system tuning, and capacity planning
- Founded the performance team (IOP) at Twitter in 2017

- Enjoy musing about (distributed) systems, and integration between emerging hardware and software trends

Why we need performance engineering (more than ever)

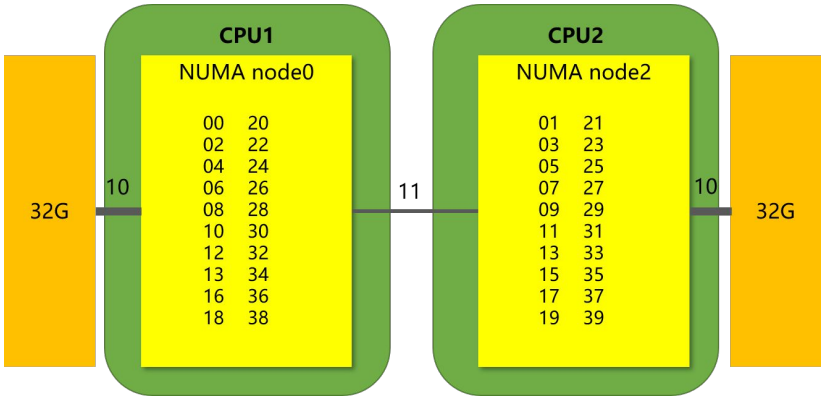
For 30 years, hardware vendors had kept most performance engineers out of a job.

Now, our moment has finally come....

Power Cap

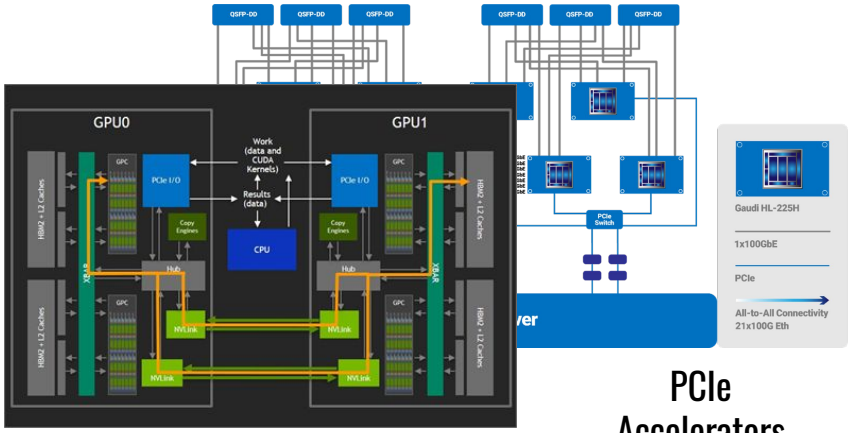
Thermal Cap

Hardware Challenges — lunch is served, but hardly free



NUMA Nodes

Credit: github.com/LyleLee



PCIe Accelerators

If you think programming against 80 CPU cores is hard, well, I have bad news for you…

ous

Programmable NIC

RDMA/RoCE

MRDIMM

CXL

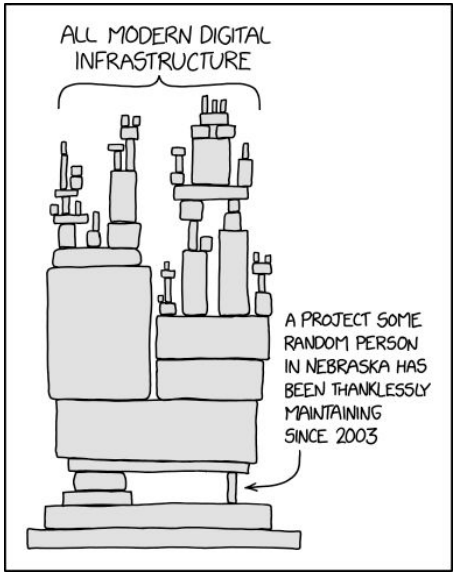
SIMD & Crypto instructions

SR-IOV

Disag

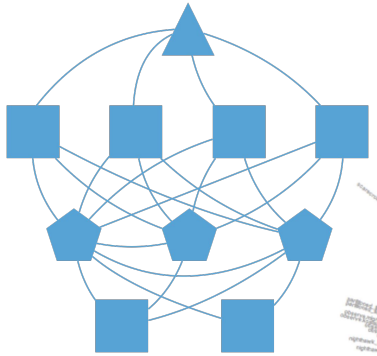
Software Challenges — modern software is anti-hero

The stack is incredibly deep and wide.



Credit:
[Randall Monroe]

The service ecosystem is a jungle — heterogenous, diverse, and connected.



It's the complexity, duh!

Performance is a System Property/Problem

Systems — models of (many) parts and relationships

Performance — a *counting* exercise

- ➔ of resources in a system
- ➔ at the right granularity
- ➔ that changes over time

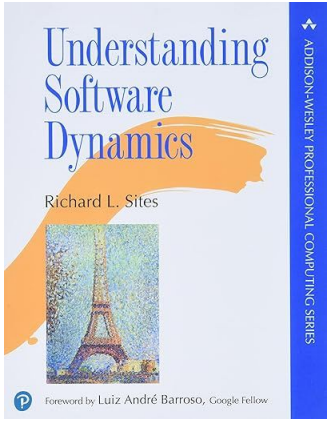
Security	**and**
Availability	**or**
Performance	**sum**

System Property Intuitions

Reliability = Availability x Performance

Cost = Availability ÷ Performance

“How Slow Should It Be?”



- Priorities

  Structure > Plumbing > Finish

- Predictability

  tail latency and consistency

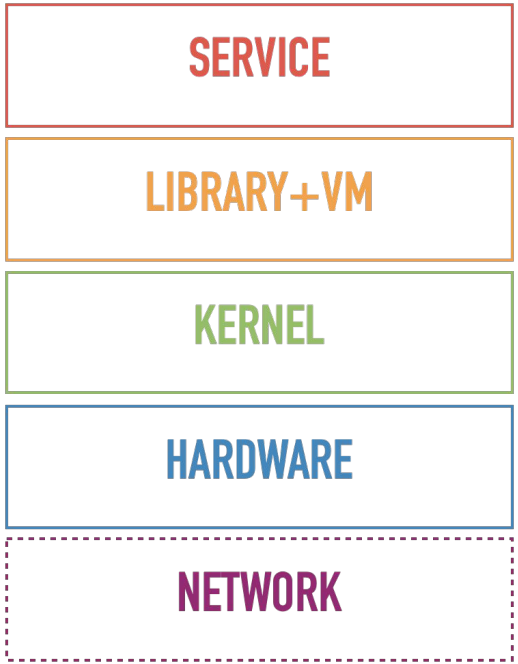
High performance services are all alike;

every under-performing service is under-performing in its own way.

Performance Engineering at Scale

Software Runtime as a Local System

- Characterize relationship between layers and components ("**`System`**")
- Metrics for different resources and components ("**`counting`**")
  - CPU, memory, network, storage…
  - syscall, scheduler…
  - GC
  - func call, ...



How We Curated Data

- Rezolus: a performance telemetry agent
  - Pluggable, largely eBPF-based samplers
  - High frequency observability
  - Extremely lightweight
- Long Term Metrics (LTM): full-stack metrics
  - From topline to hardware utilization
  - Hardware, OS, container, JVM, Finagle, application, Rezolus, env metadata
  - Minutely metric resolution with multi-year retention
  - Normalization and verification
  - All datasets query-ready with SQL

Generating Signal

Aggregating Signal

How We Used Data (to Get Insights)

- EasyPerf & EasyPerf UI for accessible GC wins
- Fleet health reports (a new team was born)
- Ranked utilization reports

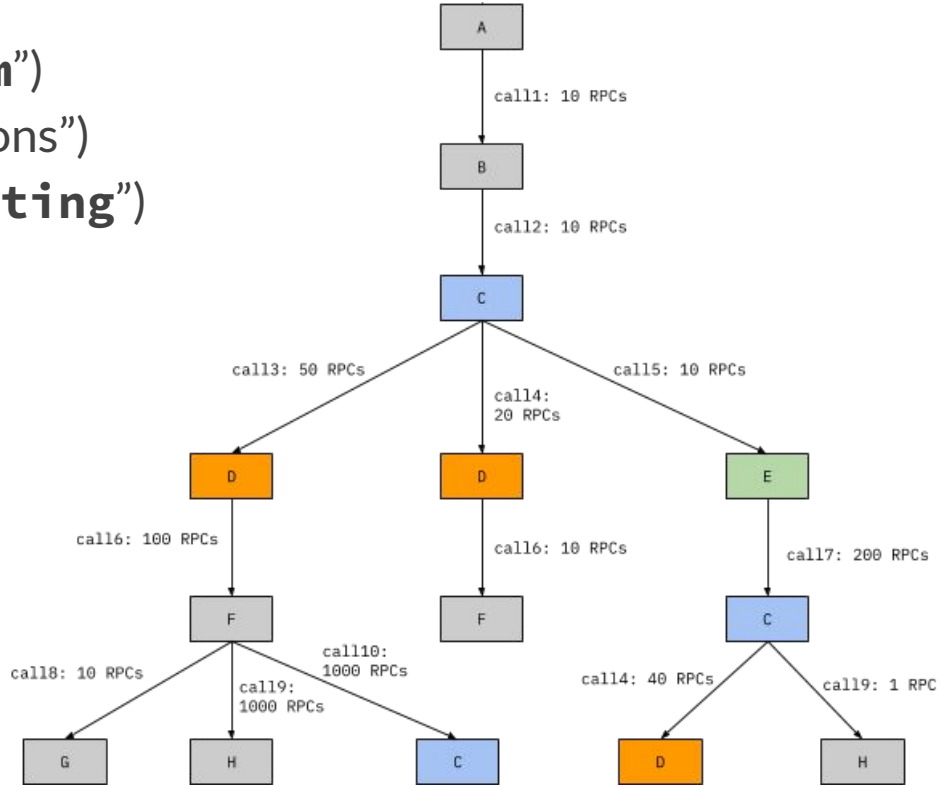
What the team did

- Capacity engineers derived utilization from LTM
- Service owners conducted performance analysis and added metrics
- HWEng, datacenter planners, and SiteOps preferred our machine metadata dataset over the canonical databases

What others did

Application as a Distributed System

- Edges between services (“**`System`**”)
- Metrics and attributes (“annotations”) associated with each edge (“**`counting`**”)
  - send/recv timestamps
  - request/response sizes
  - status labels
  - (any free formed data)



How We Curated Data

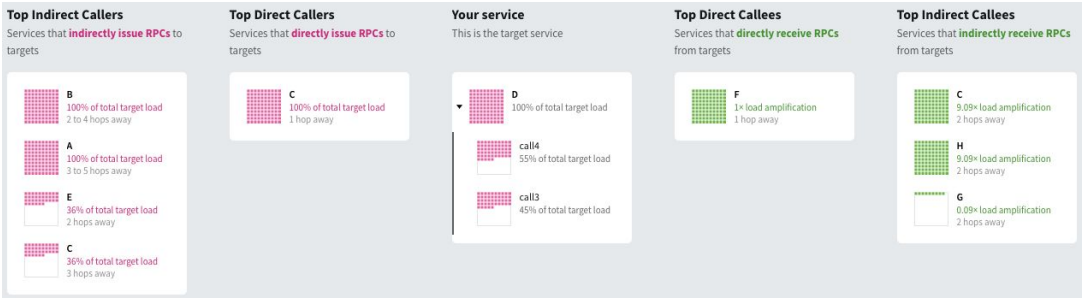
- Higher quality traces
  - Tracing data is fraught with incomplete data, clock drift, ill-formatted fields…
  - Continuous data validation and fixes
  - Methodically set & recorded sampling rate
  - Redesigned and re-implemented trace collection service
- Created a trace aggregation pipeline
  - Separation of annotations and the call graph
  - Multiple levels of indices: traces, span, annotation; and mappings between them
  - All datasets query-ready with SQL

Generating Signal

Aggregating Signal

How We Used Data (to Get Insights)

- Service Dependency (Graph) Explorer



- Latenseer: a causal model of e2e latency distribution
- Many analyses on demand

What the team did

- Data Privacy Team created a data propagation graph on top of service dependency graph

What others did

Impact of Our Work

- ❏ Eliminated the noisy neighbor problem via kernel changes
- ❏ Better configuration & bin-packing of largest services
- ❏ Direct optimization of largest services
- ❏ Better hardware selection for largest services
- ❏ >100MM savings overall

- ❏ Datacenter and cloud migration readiness
- ❏ Graceful degradation readiness
- ❏ Numerous other analyses on-demand

On Deck / In Progress

Platforms

- ❏ Augur: Fleet-wide, continuous perf profiling and aggregation
- ❏ Systems Lab: a performance testing platform

Products

- ❏ Trace Latency Visualizer

How do Performance Engineers Fit In?

They don't.

Questions Facing A Performance Engineering Team

The Fundamentals

- Business value alignment
  - Topline
  - Bottomline
- Decision making structure
  - Top-down
  - Bottom-up

Strategic Questions

- Who are my customers?
- Who is my champion?
- How to provide incentives?
- How to convince others?
- How to get recognized / promoted?

Other Thoughts

- Embodying and fostering a culture of performance
  - Amplify and facilitate *other* people's performance work
  - Be a beacon, an advocate, and a warehouse of ideas
- The unique boom-and-bust cycle of performance engineering
  - Often trending the opposite way as the business
  - Solution: Trading between front- and back-stage work

How it actually happened

There was *no* blueprint in the beginning.

I was thinking
~~really hard~~
occasionally

The Philosopher's Approach to Performance

Does performance have meaning?

Can performance be judged objectively?

Is performance the same everywhere, all the time?

How can I find performance?

Am I alone in my pursuit for performance?

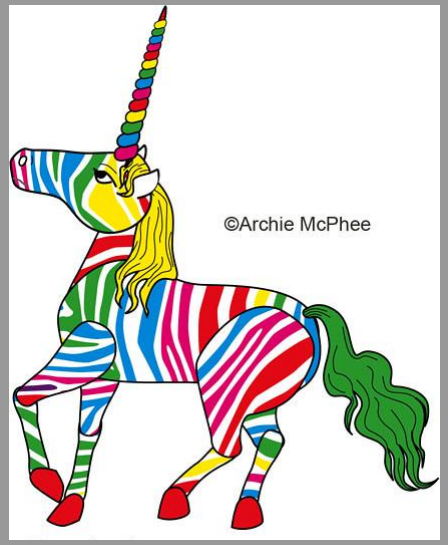
How do I share performance?

(ideal)

(reality)

The ideal performance engineer —

has background in hardware,
is good at reading/writing software,
has dealt with large-scale operations,
speaks fluent analytics,
understands the language of business,
can communicate effectively with all kinds of job roles and all levels of leadership. (Walks on water…)



We need a team.

We need a diverse team.

Find those that are [much] better than you [at some of the things].

The First Phase (2017-2018, bootstrapping)

- Perf telemetry work commenced
- Lots of “odd jobs”
  - Consultation and debugging
  - Prototypes of ideas
  - Small wins
  - Favors (I somehow got involved in GDPR…)
- Writing down our vision & methodology

4 people

All internal, SRE heavy

The Second Phase (2018-19, laying foundations)

- ❏ Tracing work commenced
- ❏ Long Term Metrics work commenced
- ❏ Surveying new technologies for optimization opportunities
- ❏ Rezolus achieved initial success
- ❏ Offering more consultation

~8 people

Added hardware and analytics expertise

The Third Phase (2020-2021, maturing)

- ❏ Primary datasets become usable
- ❏ Consulting now largely driven by inbound requests
- ❏ Frequently participated in key projects or crisis response
- ❏ Started to build more products
- ❏ Internal/External branding effort via talks and publishing

10+ people

Added UX and domain expertise

The Fourth Phase (2022, an expansion cut abruptly)

- ❏ Coordinating several multi-team efficiency projects
- ❏ Planning and building the next batch of platform investments
- ❏ More investment into accessibility of our data via product and publications
- ❏ Negotiated an embedding model for localized performance engineering

~10 people

Teams sprouted around us

Lessons Learned

The technical and social considerations of performance engineering are equally important.

Scalable Performance Methodology

- Strength in number: perf needs a signal-to-insight pipeline, and an opportunity-to-impact funnel
- Best places to look: infrastructure with broad impact, or areas with significant footprint
- Create platforms & products that enable a large number of individuals and teams to do similar work on their own
- Design the team to fit the broader organizational
- Outreach is serious work: education, consulting, collaboration, embedding, credit sharing…

People make work happen. Each person's strengths and personality matter.

Building a Good Performance Team

- Treasure excellence
- Seek diversity
- Embrace chance
- Be helpful, be generous, make friends, build a culture

What happened to IOP after Twitter?

IOP SYSTEMS

Yao Yue | Mihir Nanavati | Yuri Vishnevsky | Xi Yang | Brian Martin | Sean Lynch

Turning performance engineering expertise into platforms and products that could benefit *everyone*.

Let's talk about Performance

Yao Yue
yao@iop.systems
https://iop.systems

Appendix

Who We Were

Full-timers: Yao Yue, Brian Martin, Anatole Shaw, Andy Wilcox, Rebecca Isaacs, Dan Luu, Jonathan Simms, Kunal Trivedi, Michael Leinartas, Xi Yang, Mihir Nanavati, Sean Lynch

Part-timers: Yuri Vishnevsky, Oleksandr Kuvshynov

Interns: Juncheng Yang (x2), Sean Lynch (x2), Danieil Skrinikov, Lexiang Huang, Anna Karanika, Yazhuo Zhang

And many, many friends throughout Twitter 1.0: SRE, KaOS, HWEng, CSL, Capacity Planning, Revenue, Fleet Health, Kite (Service Identity & Chargeback), SiteOps, Core Services, TPM, Finance, …

What We Published

Open Source

Rezolus, rpc-perf, Pelikan Cache, cache-trace, Density Plot, misc Rust utilities, …

Blogs

Metric analytics, Trace analytics, Cache incidents, Using ADQ, Rezolus

Papers

OSDI'20 (TOS V17.3), NSDI'21, FAST'23, HotOS'23, SOSP'23, SoCC'23, NSDI'24

Talks

YOW! 2017, SNIA PMEM Summit 2020, SDC 2020 (keynote), Strange Loop 2021, P99 Conf 2022, QConSF 2022