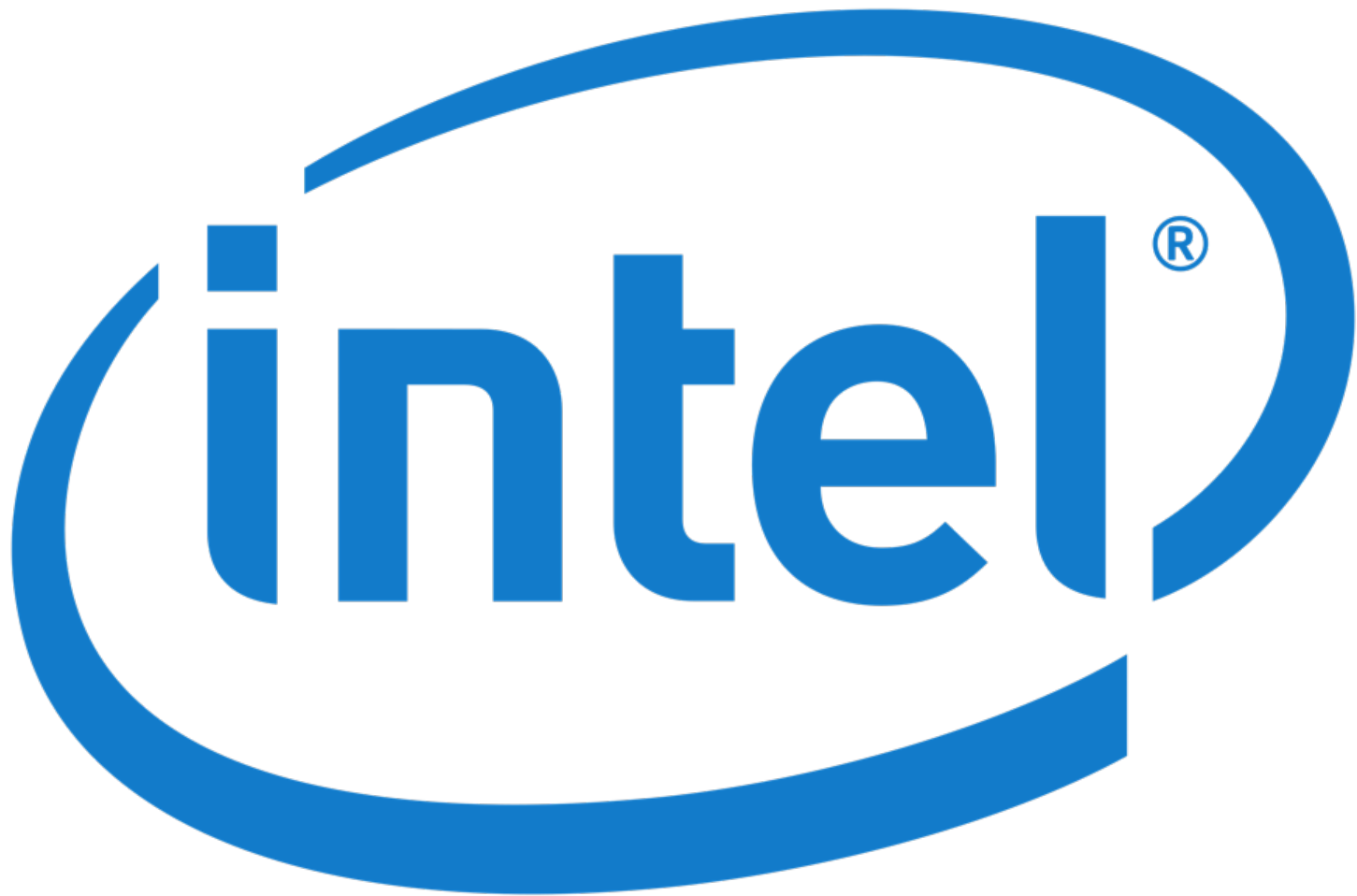
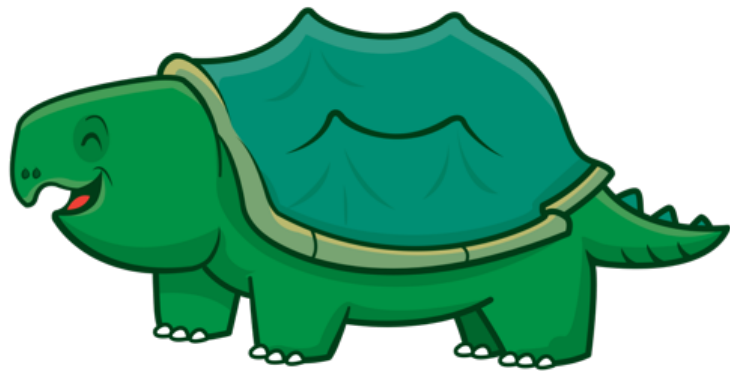


Hi

Nicholas Weaver
@lynxbat

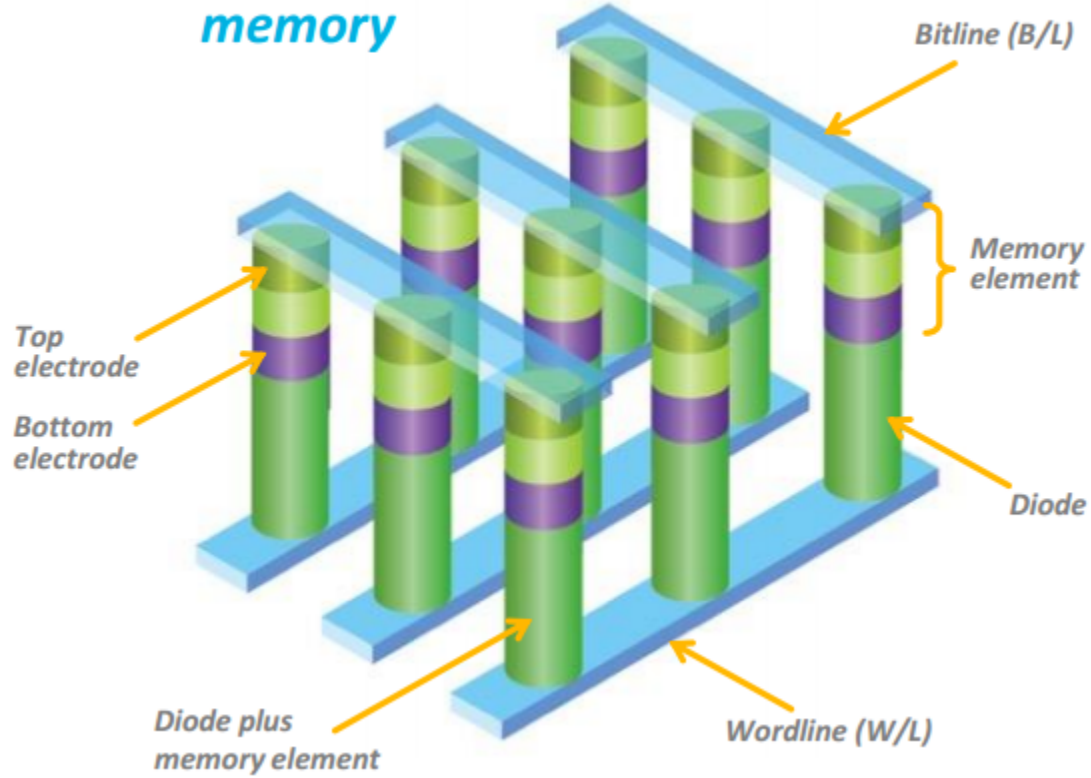








3D XPoint memory





CPU

Memory

Network

Storage

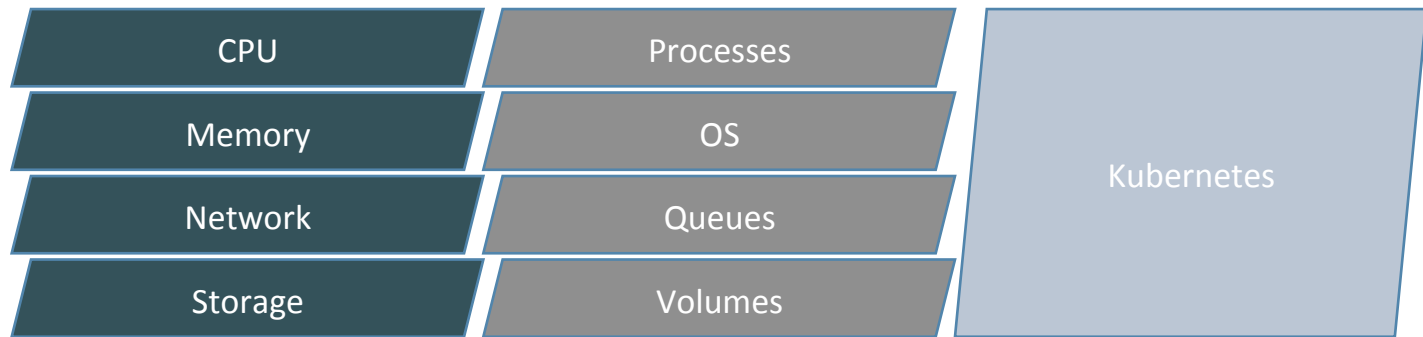


Applications

Services

“Stacks”

Jobs



Data

Hardware

Runtime

Application

Service

Context

Location

Alerting

Tagging

GEO

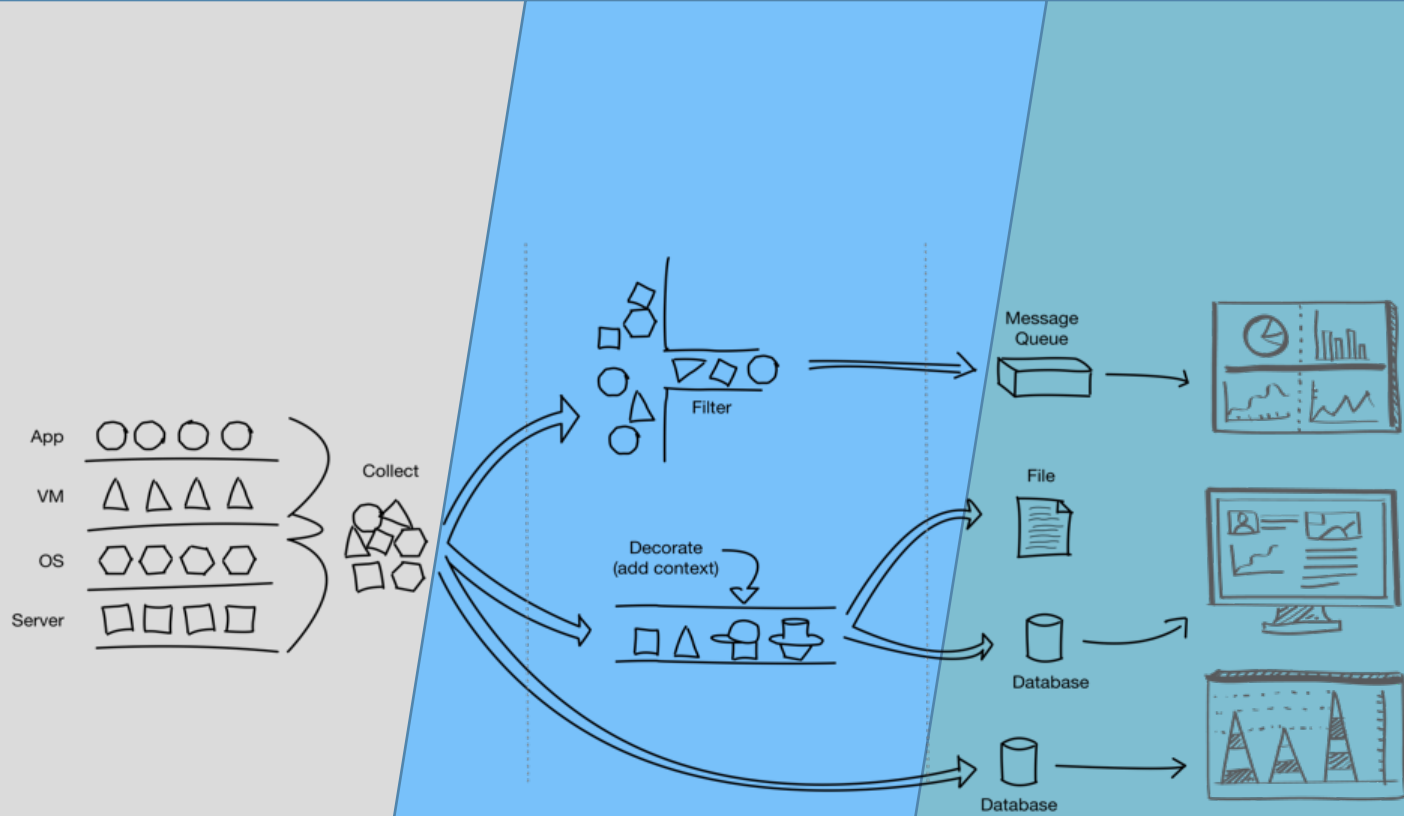
Places

Database

Message Bus

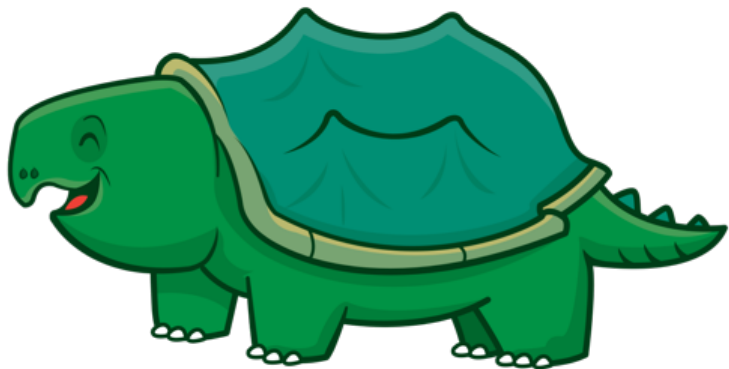
Schedulers

APIs



Snap

An open telemetry framework



The "Behind the scenes" work

- Testing infrastructure
- Automated release and build processes
- Plugin stability testing

The same original features but better

- Tribe clustering
- Dynamic plugin upgrades
- REST API
- Snap Workflow

↑ GRPC ↓

GRPC



Go



before **Dynamic Metrics**

Grab **Interface 1** **Packet Throughput** on Container ID == **Meeseeks**

Grab **Interface 2** **Network Packet Throughput** on Container ID == **Squanchy**

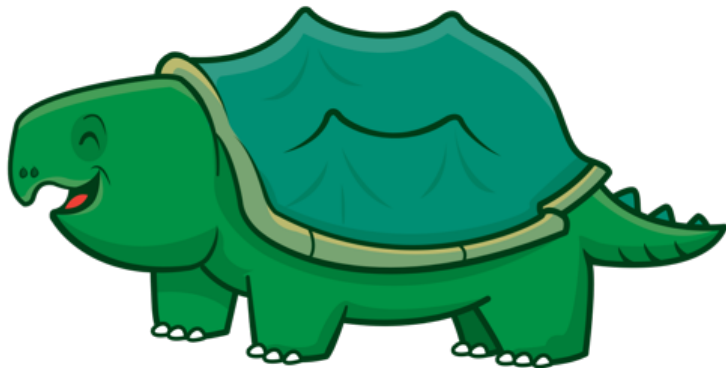
Grab **Interface 1** **Network Packet Throughput** on Container ID == **RickSanchez**

after **Dynamic Metrics**

Grab **Interface** * **Packet Throughput** on Container ID == *

Snap 1.0

An open telemetry framework



Snap applied



kubernetes



Snap features for 2017

Tribe 2.0

Streaming (!)

Embedded Plugins

Tons of new Intel plugins

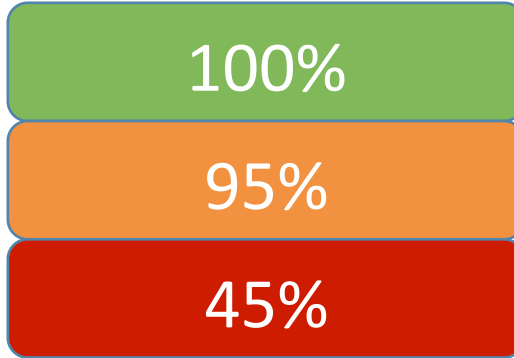
We think Snap itself is kind of cool

But we built it for what is next

What if?

What if?

Hardware health was modeled for you?



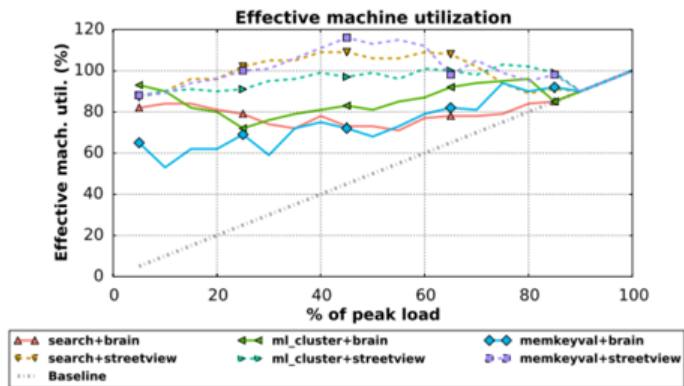
What if?

Predicting failure became the norm rather than reacting to it?



What if?

Workload measurement was predictable?



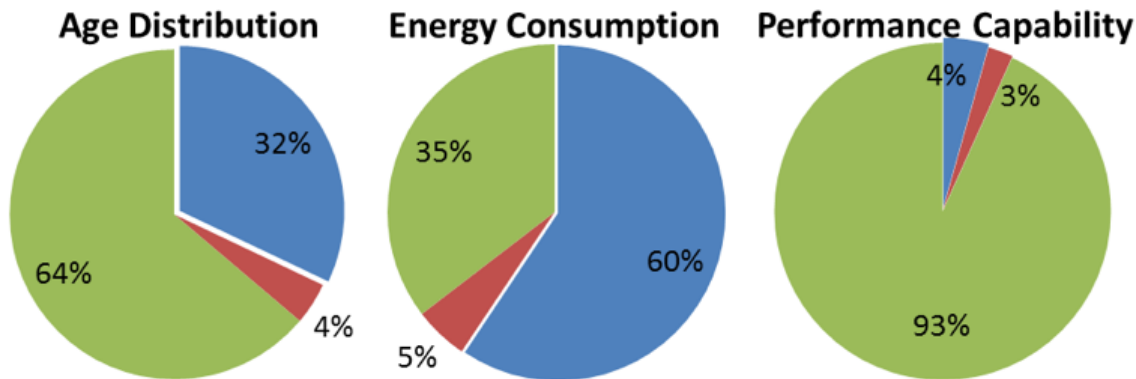
What if?

Colocation was predictable?

Scenario / Load	5%	15%	25%	35%	45%	55%	65%	75%	85%	95%
Caffe	32.0%	33.2%	35.0%	37.9%	43.3%	48.0%	51.5%	58.2%	70.6%	230.8%
Stream 100M	39.6%	63.4%	169.9%	948.3%	1826.6%	2588.3%	2639.9%	2592.5%	2452.2%	2450.3%
memBW	31.8%	35.1%	37.2%	40.9%	45.5%	50.9%	60.0%	77.9%	101.0%	106.8%
L1 Instruction	33.9%	35.7%	37.7%	44.3%	53.6%	76.0%	988.5%	782.3%	1282.4%	950.1%
L1 Data	32.9%	33.9%	37.5%	41.3%	48.2%	55.1%	78.6%	124.1%	479.3%	488.0%
Baseline	32.0%	34.8%	36.4%	38.7%	43.9%	48.1%	52.8%	55.9%	64.3%	75.2%

What if?

Schedulers like Kubernetes implicitly understood power, thermal, workload SLA's, networking queues and more?



What if?

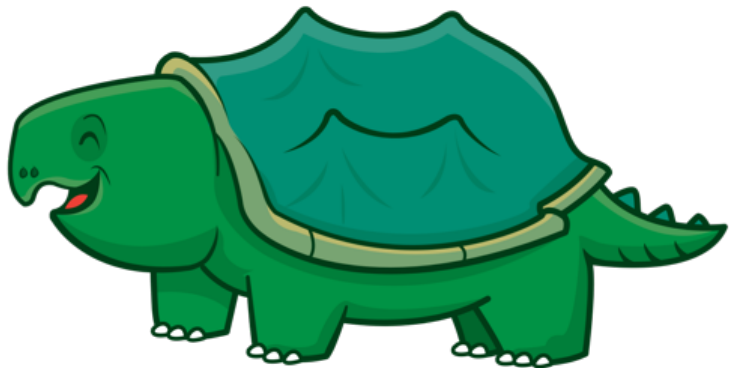
**Hardware was designed to expose telemetry
That is relevant to the workloads today?**

What if?


There was an open source tool that was a foundational component to solving these problems?

Snap 1.0

An open telemetry framework



CALL TO ACTION

- Try Snap: snap-telemetry.io or github.com/intelsdi-x/snap
- Attend **Joel** and **Matt's** session tomorrow for a deeper dive
- Find me and ask me about 
- Reach out to your **Intel** account person or **Intel** partner about Snap

Thanks

