In-Memory Computing SUMMIT | NORTH AMERICA 2018

Auto-Scaling Caches and Data Paths

Irfan Ahmad
CachePhysics

About

Irfan Ahmad

CachePhysics Cofounder
CloudPhysics Cofounder
VMware (Kernel, Resource Management),
Transmeta, 40+ Patents
University of Waterloo
@virtualirfan

CachePhysics

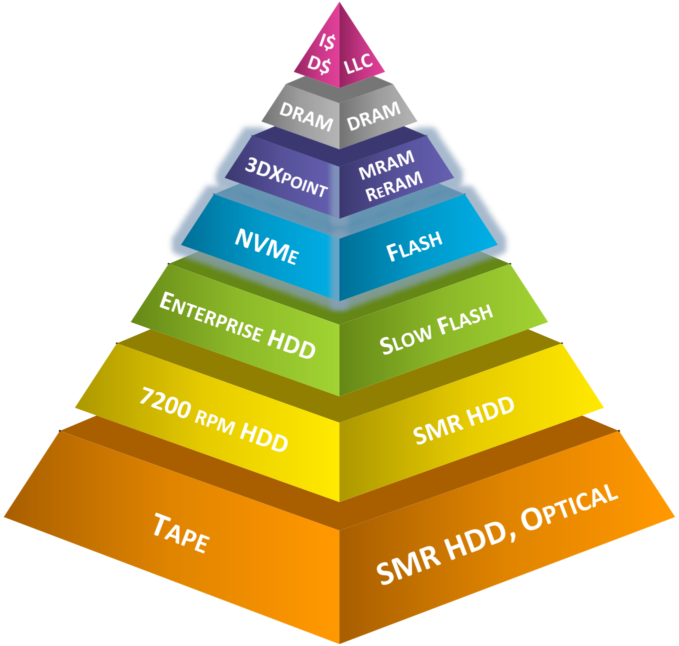
Data Path Monitoring and Modeling Software

Real-time Predictive Modeling of Data Access Patterns
Increasing Performance & Cost Efficiency of Existing Caches
Powering Next-Generation Self-Learning Caches

In-Memory Computation is Hard

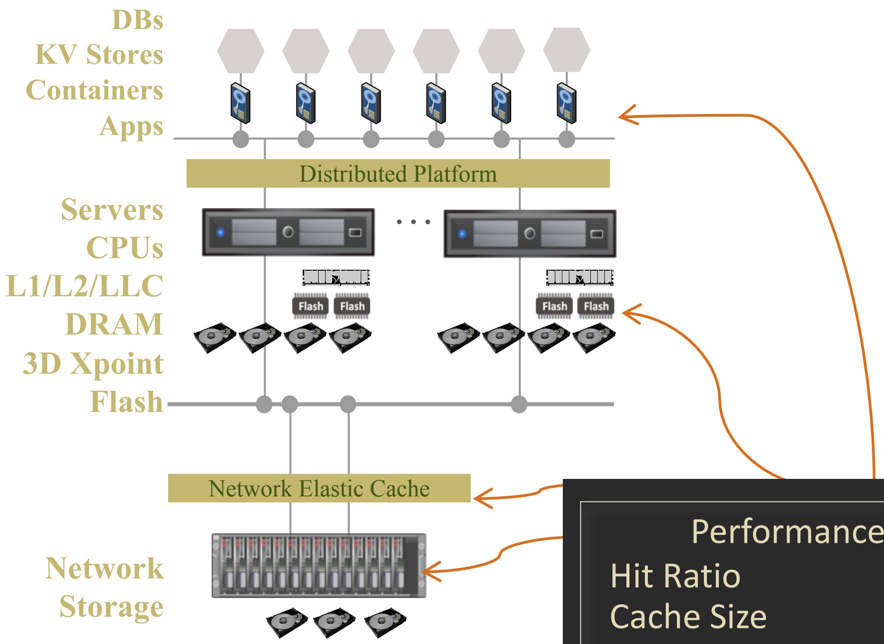
App **Data** needs **changing** daily. Providing **QoS** has become **hard**

Data Path Getting *More* Complex



The **problems** are **getting** much **worse** with increasing hardware complexity

Data Path Performance is Critical



Yet…

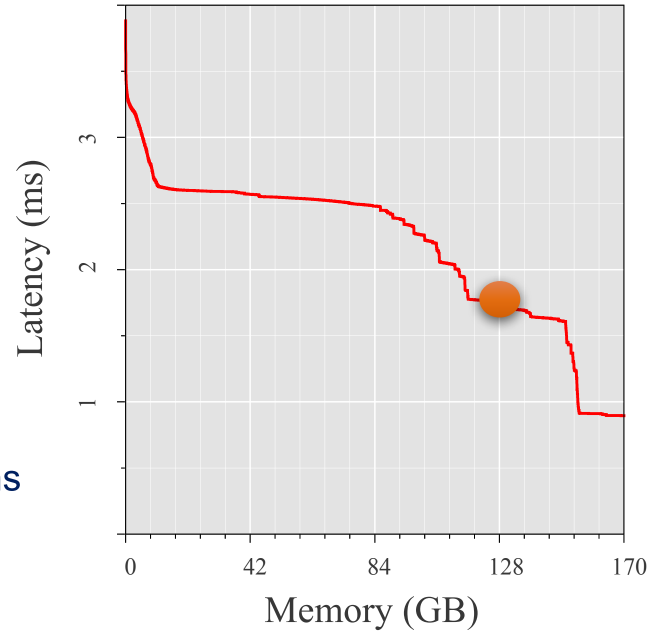
Performance	
Hit Ratio	65%
Cache Size	128 GB
99.5%ile Latency	16.7 ms

Intelligent Management is Non-Existent

- Is this performance good?
- Can performance be improved?
- How much Cache for App A vs B vs …?
- What happens if I add / remove DRAM?
- How much DRAM versus Flash?
- How to achieve 99%ile latency of X μs?
- What if I add / remove workloads?
- Is there cache thrashing / pollution?
- What if I change cache parameters?

Algorithms to the Rescue

Performance	
Hit Ratio	65%
Cache Size	128 GB
99.5%ile Latency	16.7 ms

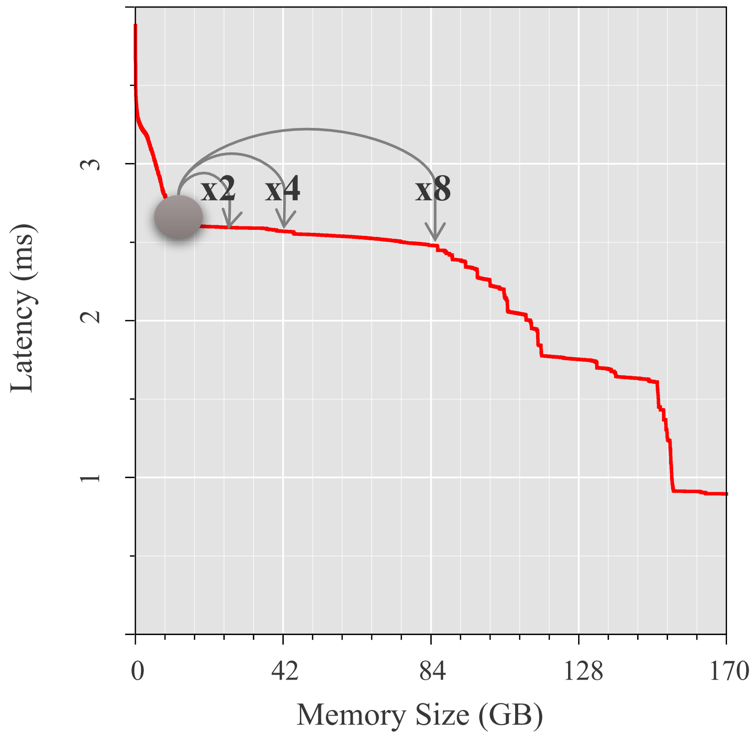


Learn performance model of applications and cache

Predict the performance of workload as *f(memory size, params)*

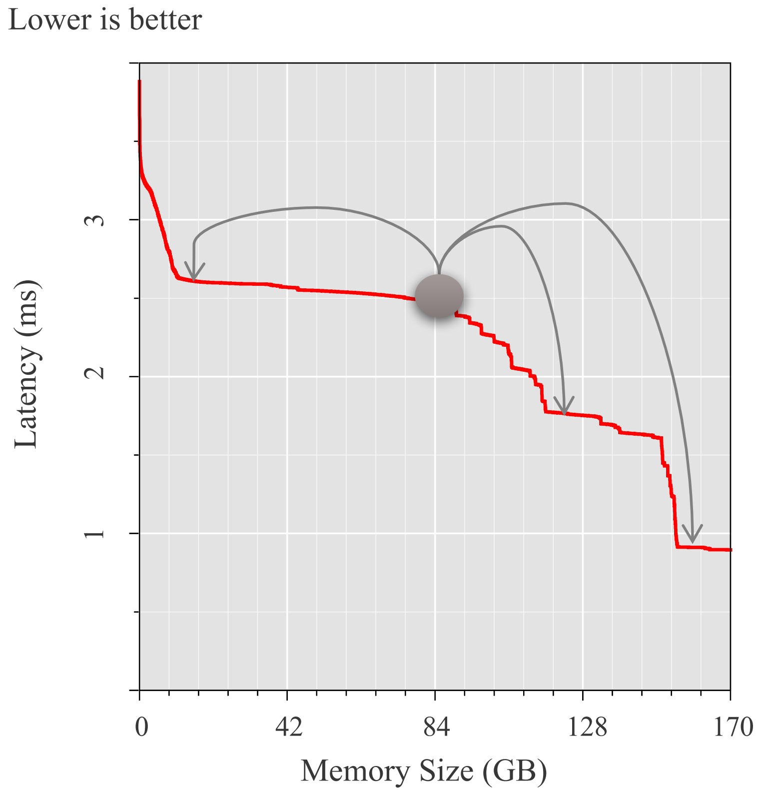
Understanding Performance Models

Lower is better



Decide useful increments of change.

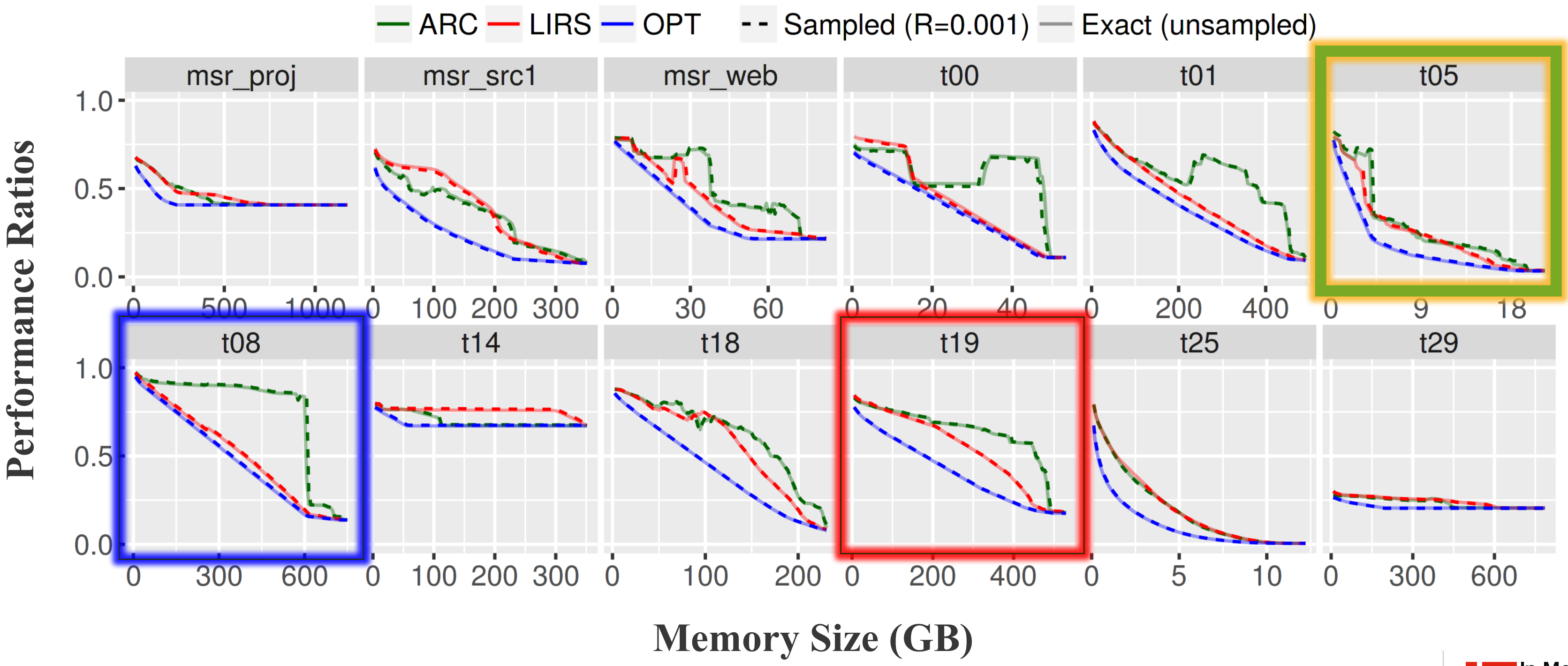
Understanding Performance Models (2)



The only 3 efficient operating points for this workload.

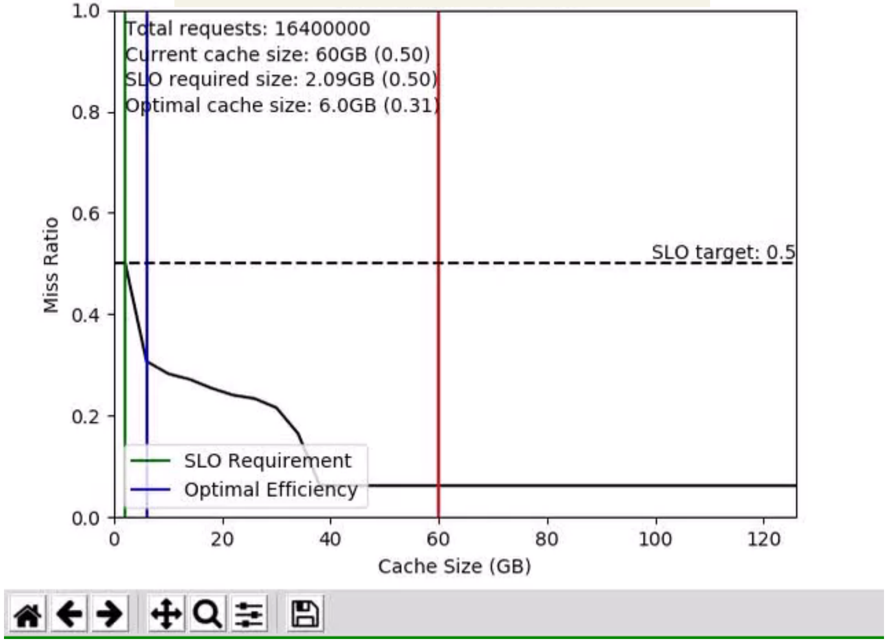
Note: most operating points are highly inefficient.

Production Workload Performance

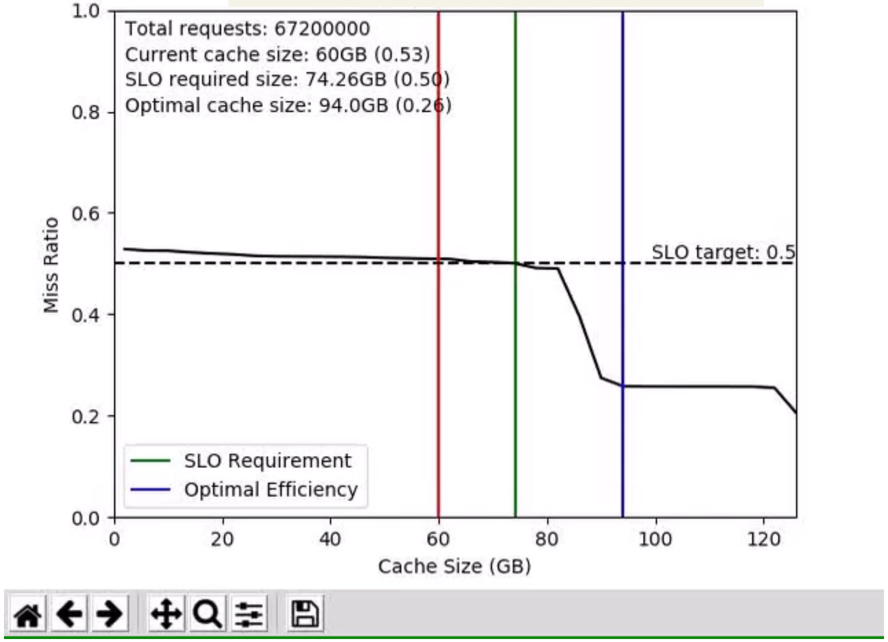


Workload Behavior: Highly Dynamic

High Velocity Workload

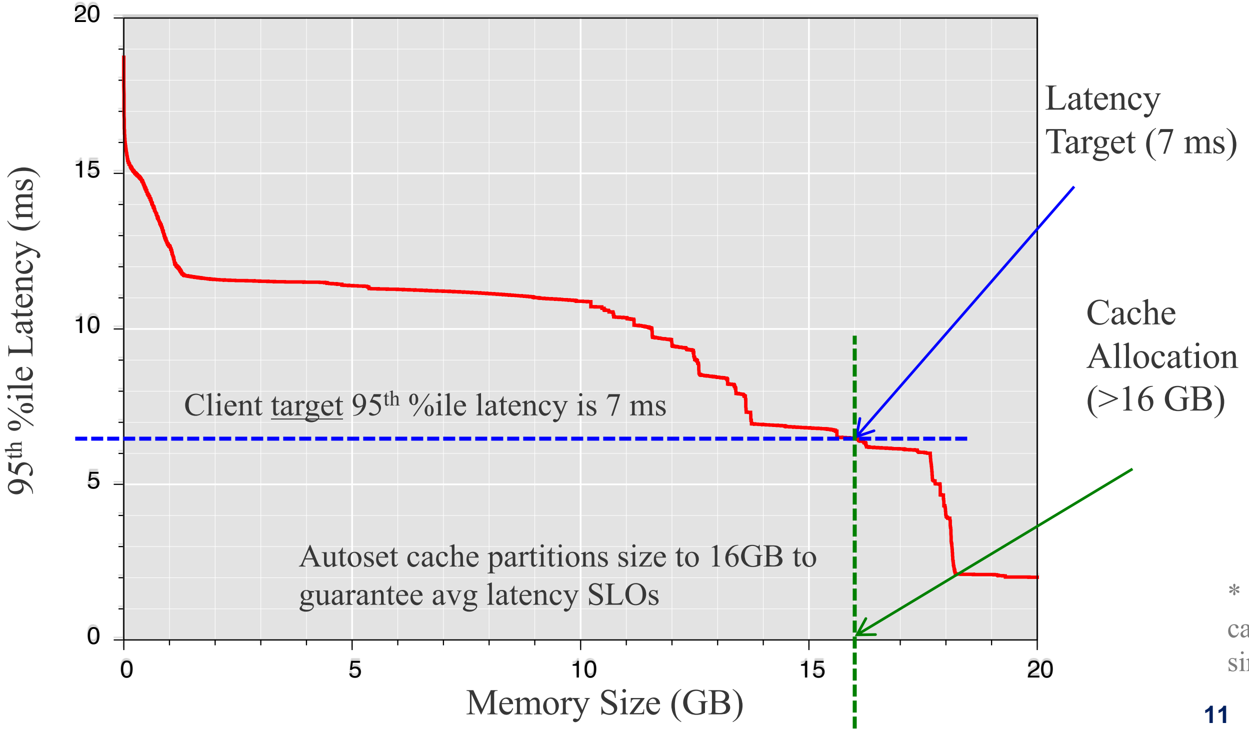


Low Velocity Workload



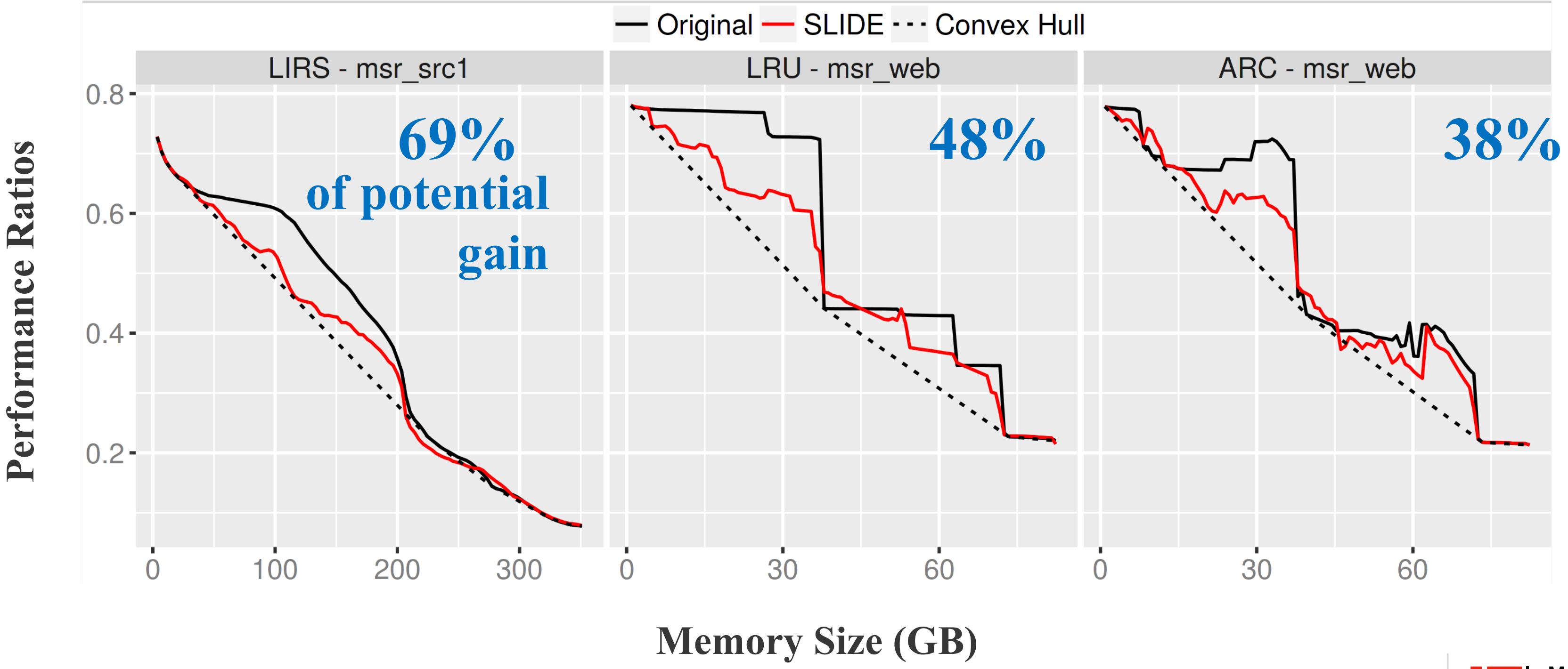
In-Memory Computing SUMMIT NORTH AMERICA 2018

Use Case: Achieving Latency Targets

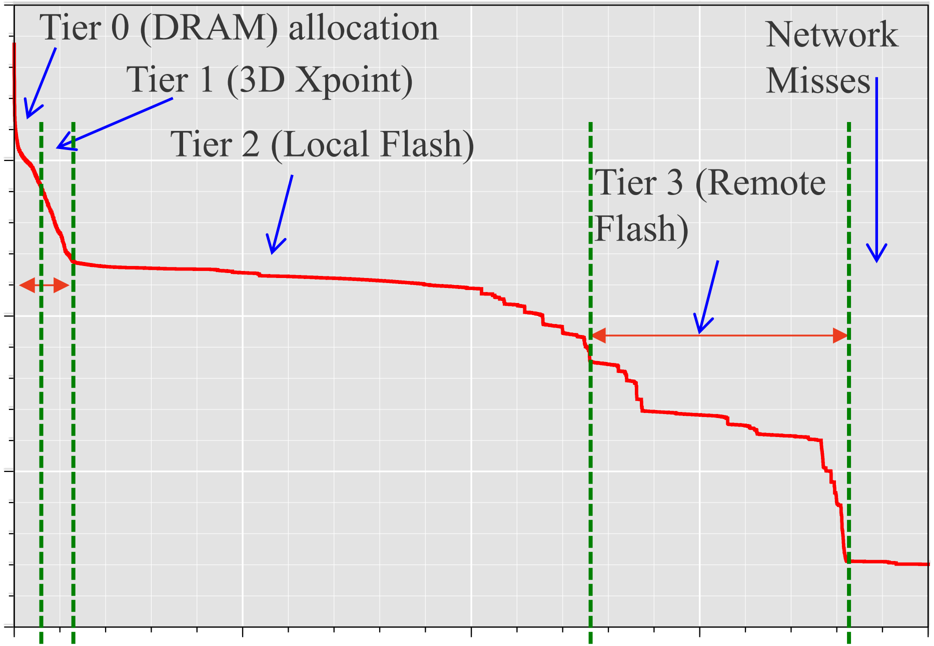


* Throughput targets can be implemented similarly

Use Case: Memory Thrash Remediation



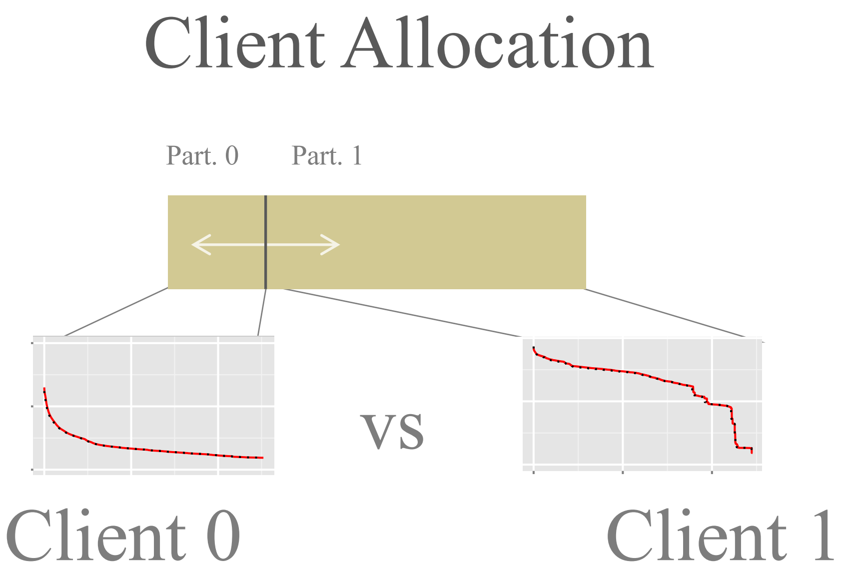
Use Case: Multi-Tier Sizing

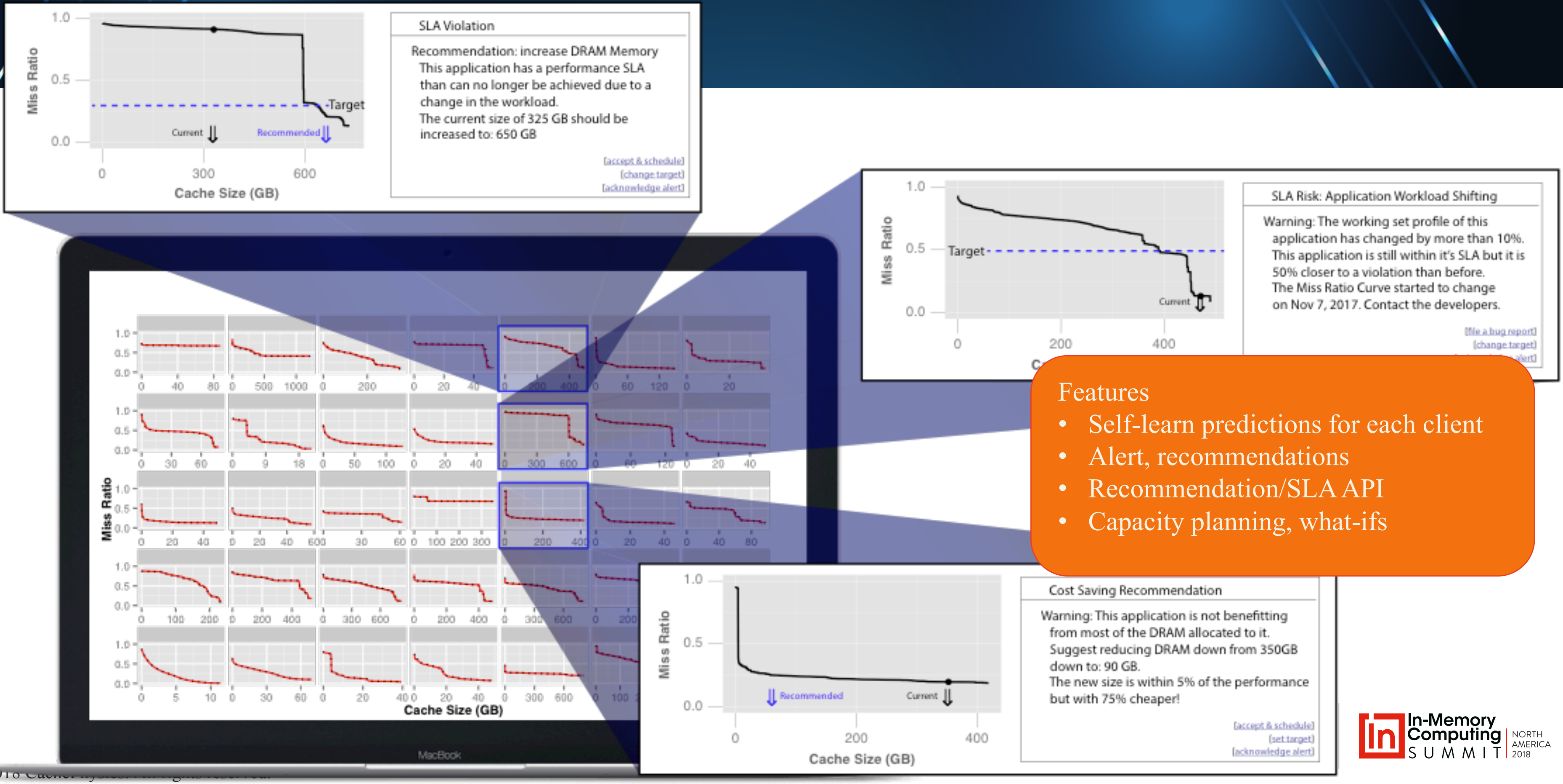


* Can model network bandwidth as a function of cache misses from each tier

Use Case: Per-App Memory Pools

- Improve aggregate cache performance
- Allocate memory based on benefit
- Prevent inefficient utilization / thrashing
- Adapt to changing workload behavior

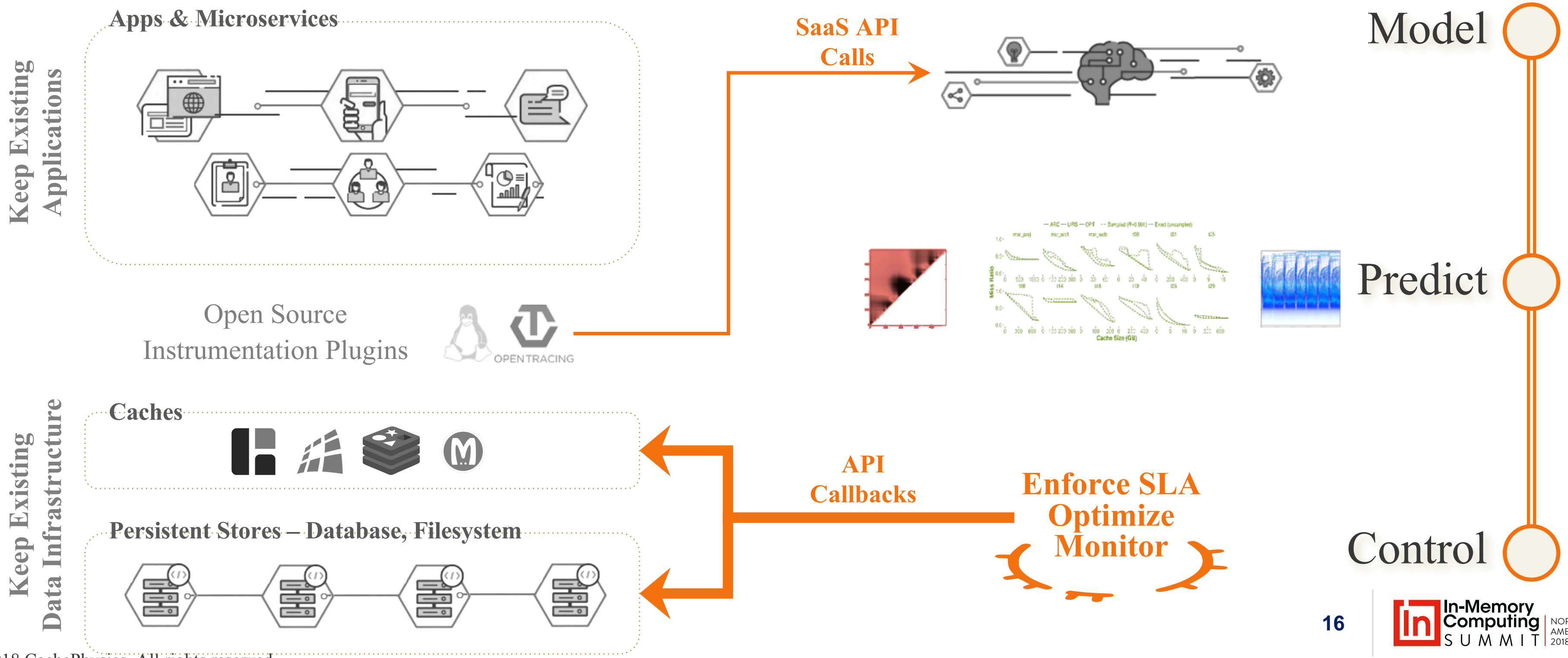




Features

- Self-learn predictions for each client
- Alert, recommendations
- Recommendation/SLA API
- Capacity planning, what-ifs

Architecture



Finally

Takeaways

Tech breakthrough for In-memory Computing

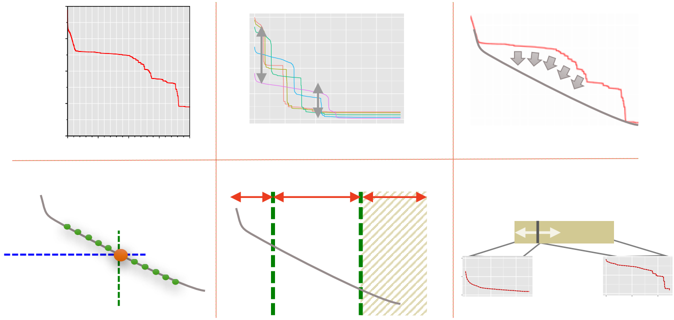
- Optimal cost, same performance
- World 1st/only latency SLOs
- Self-tuning data path
- Auto-scaling data paths

Award-winning technology

Asks

New Customer Projects

- 50% Efficiency Gain Guarantee!
- Latency SLO guarantee
- In-memory compute
- Database, Key-Value store, Filesystem, Disk system Optimization



CachePhysics

irfan@cachephysics.com 650-417-8559 @virtualirfan