SELF-LEARNING CACHES

IRFAN AHMAD
CACHEPHYSICS

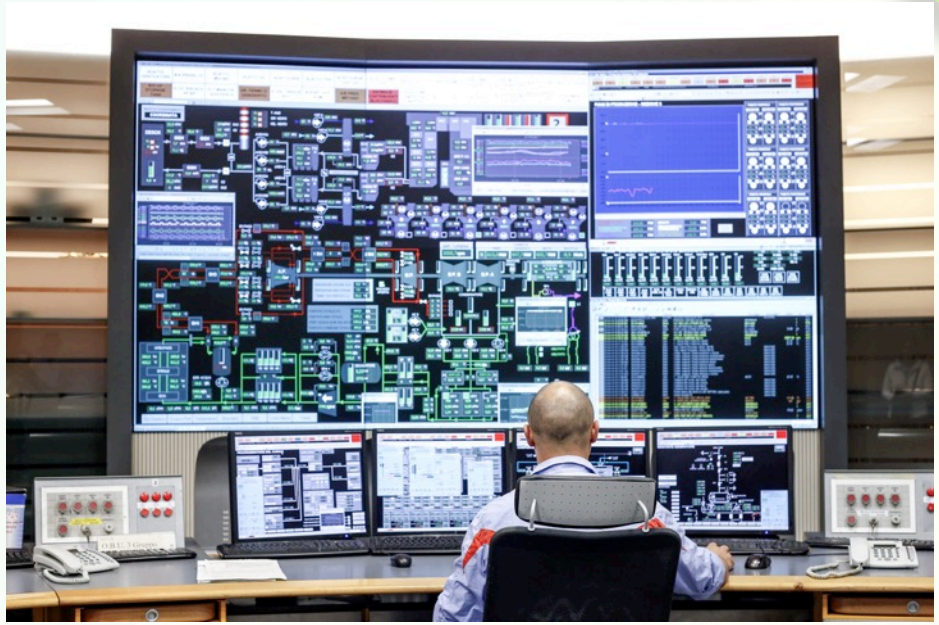
ABOUT

Irfan Ahmad
CachePhysics Cofounder
CloudPhysics Cofounder
VMware (Kernel, Resource Management),
Transmeta, 30+ Patents
Pink Tie from 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

TYPICAL AUTOMATION JOURNEY

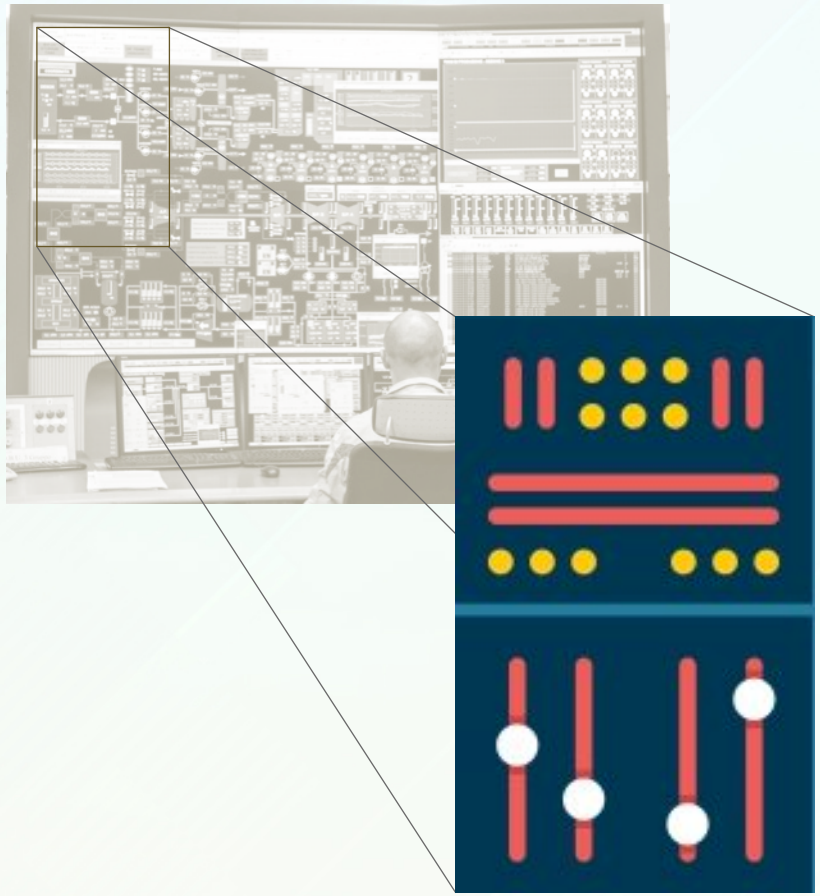


Automation: DONE

Knobs and Levers: LOTS

Photo credit: Opservices.com

POST-AUTOMATION WORLD CHALLENGES



Which **Knobs** to **Turn** and by

How Much?

POST-AUTOMATION WORLD CHALLENGES

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

NEW TWIST: DATA PATH GETTING MORE COMPLEX

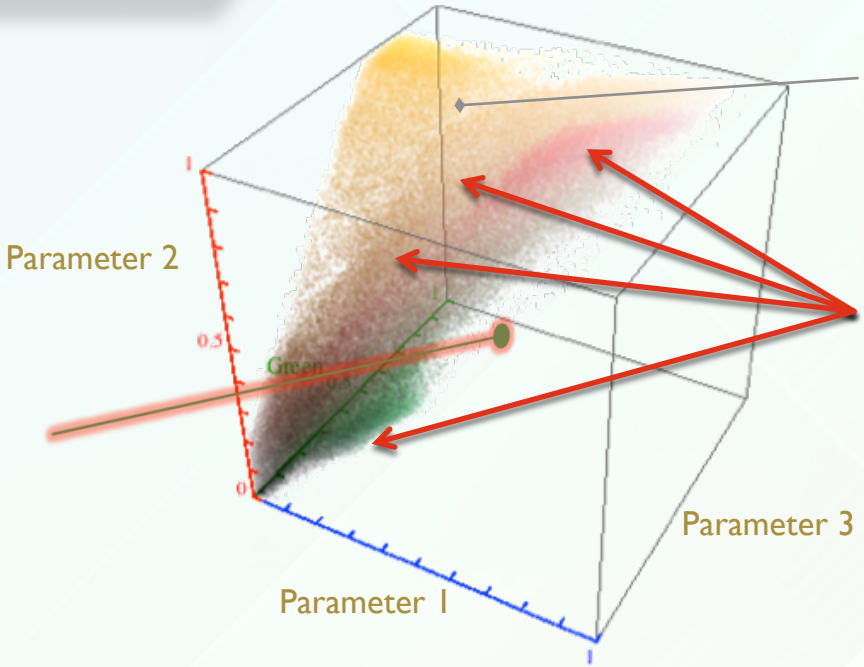
The **problems** are **getting** much **worse** with increasing hardware complexity

What's the Path Forward?

How about a Self-Learning Data Infrastructure?

STATIC DATA INFRASTRUCTURE

Today: parameters chosen based on benchmarking.
One size does not fit all.



Each point represents optimal settings for a single application

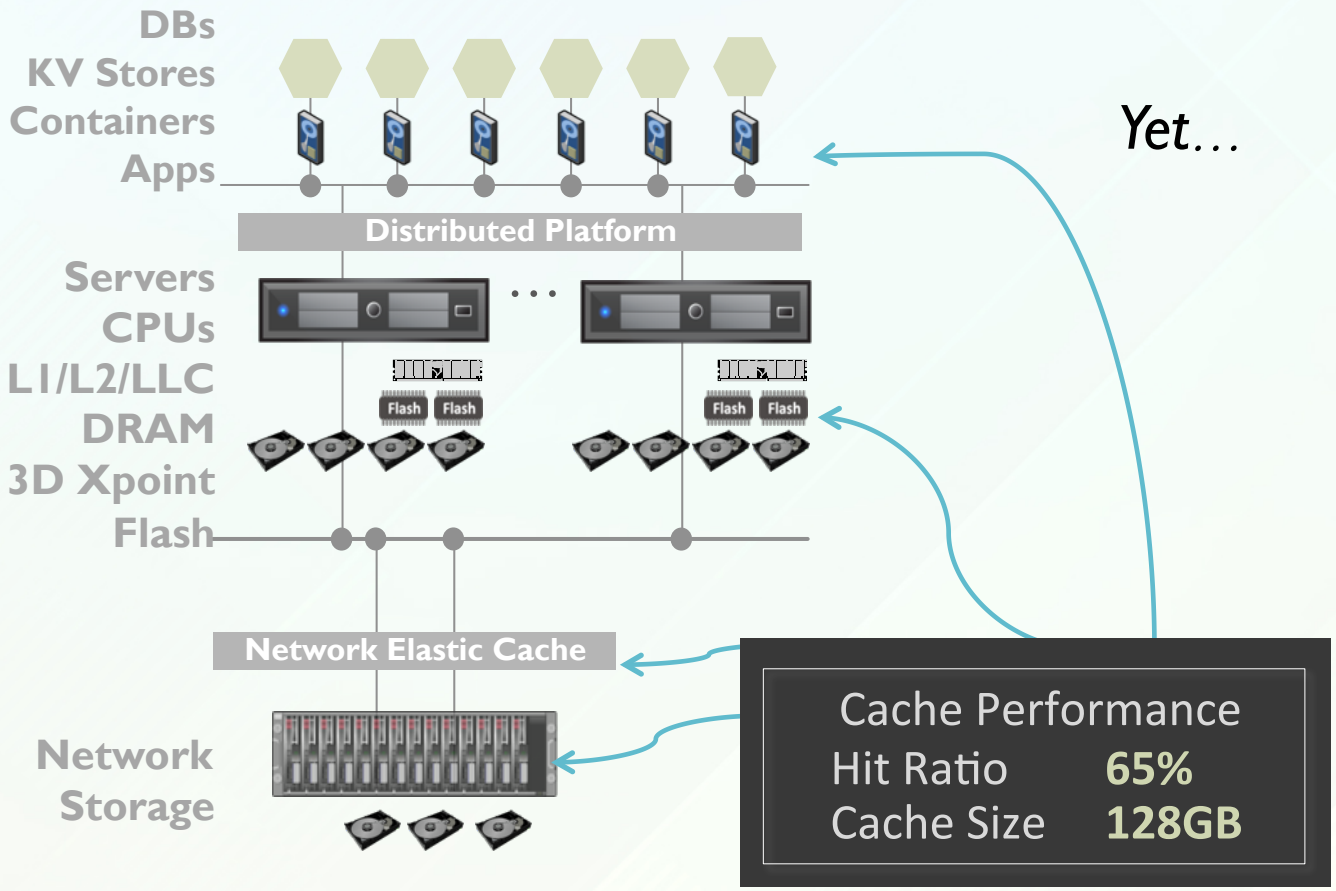
Applications have dramatically different *optimal* size and parameter settings.

Static Data Infrastructure Vulnerable to:

- Thrashing, Scan pollution
- Gross unfairness, Interference
- Unpredictability

⇒ **Overprovisioning**
⇒ **Lack of Control**

CACHES ARE CRITICAL TO *EVERY* APPLICATION



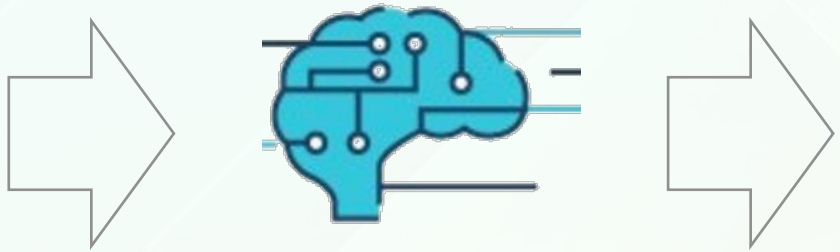
Yet…

Intelligent Cache Management is Non-Existent

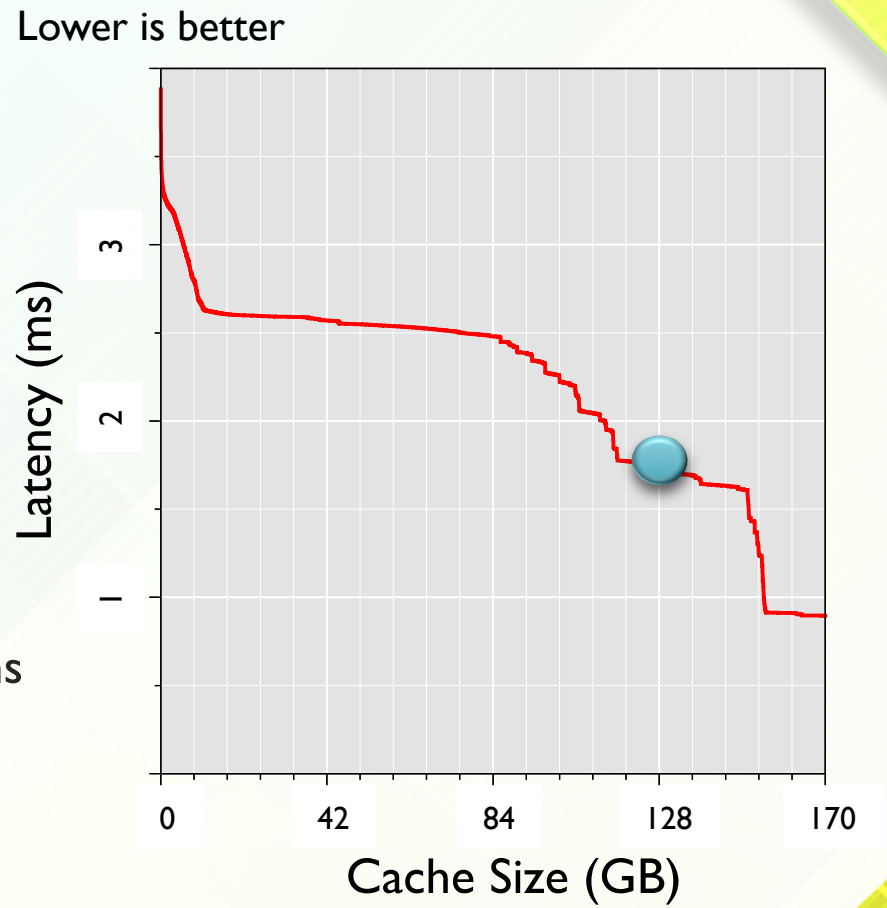
- Is this performance good?
- Can performance be improved?
- How much Cache for App A vs B vs …?
- What happens if I add / remove DRAM?
- How much DRAM versus Flash?
- How to achieve 99%ile latency of X µs?
- What if I add / remove workloads?
- Is there cache thrashing / pollution?
- What if I change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

MODELING PERFORMANCE IN REAL-TIME

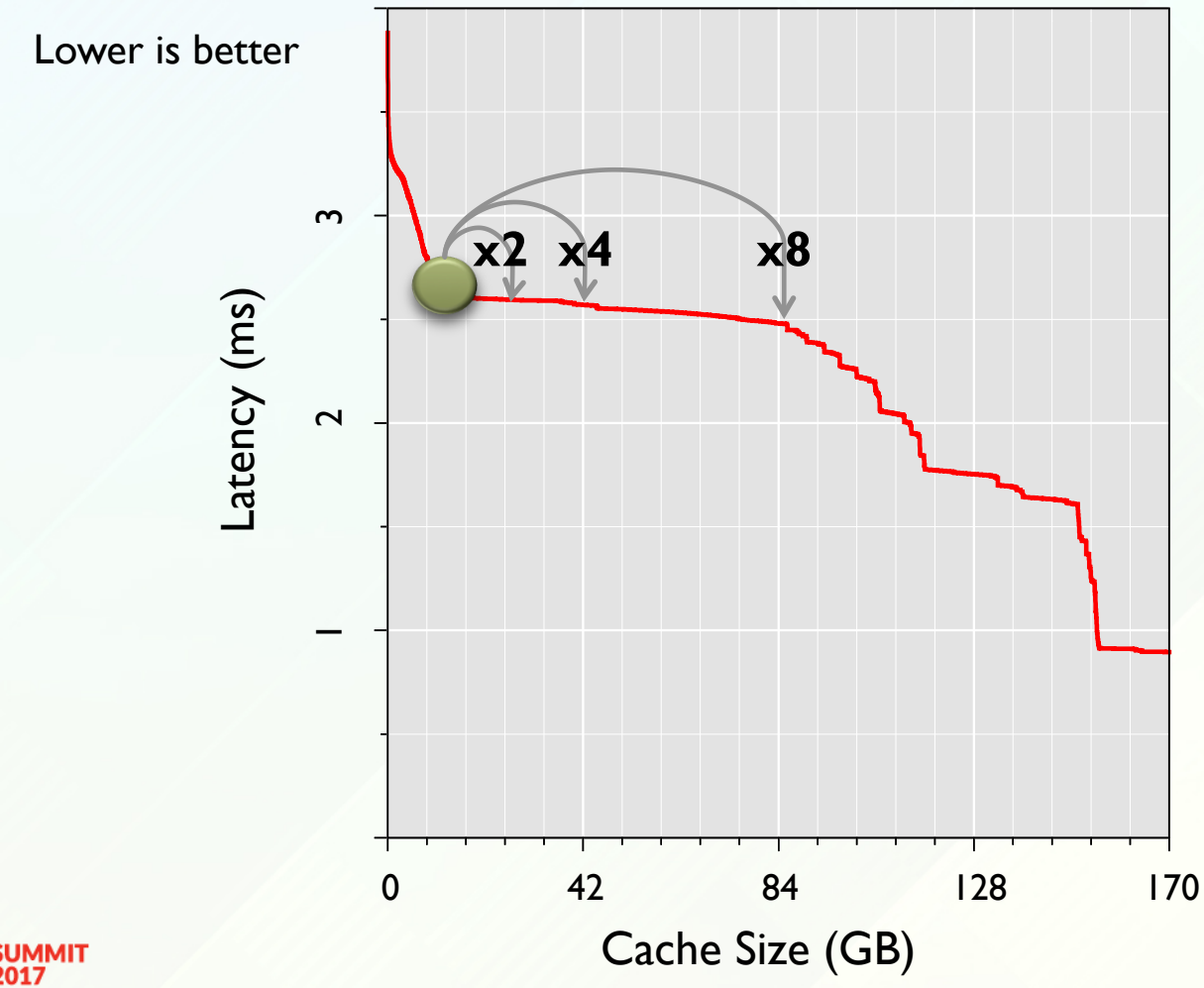
Cache Performance	
Hit Ratio	**65%**
Cache Size	**128GB**



- Learn performance model of applications and cache
- Predict the performance of workload as *f(cache size, params)*



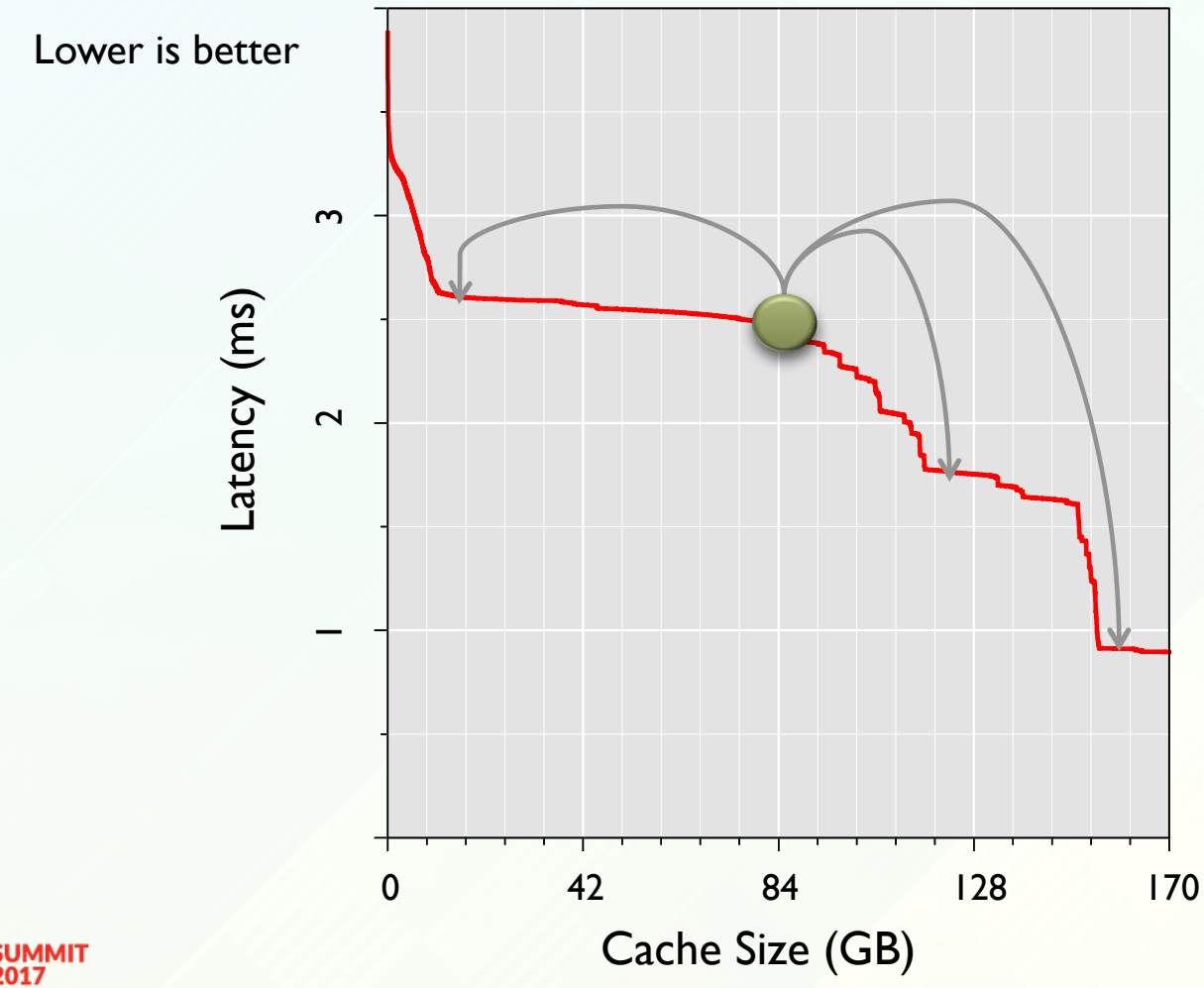
UNDERSTANDING CACHE MODELS



Models help decide useful increments of change.

In this example, no benefit despite an 8x increase in budget.

UNDERSTANDING CACHE MODELS

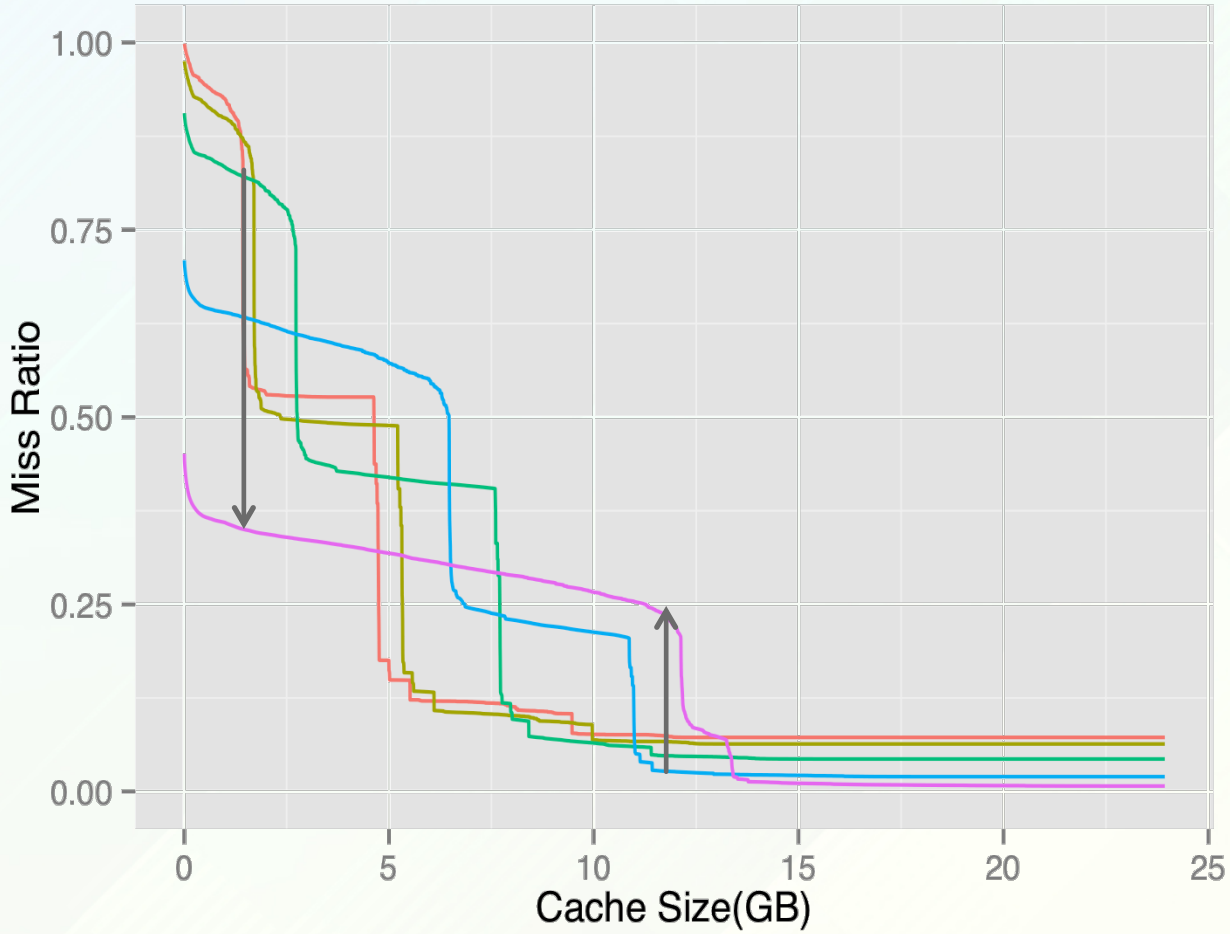


Often, most operating points are highly inefficient.

This cache is operating at the lowest ROI point; equivalent performance to 1/8 the budget.

Arrows represent the efficient operating points.

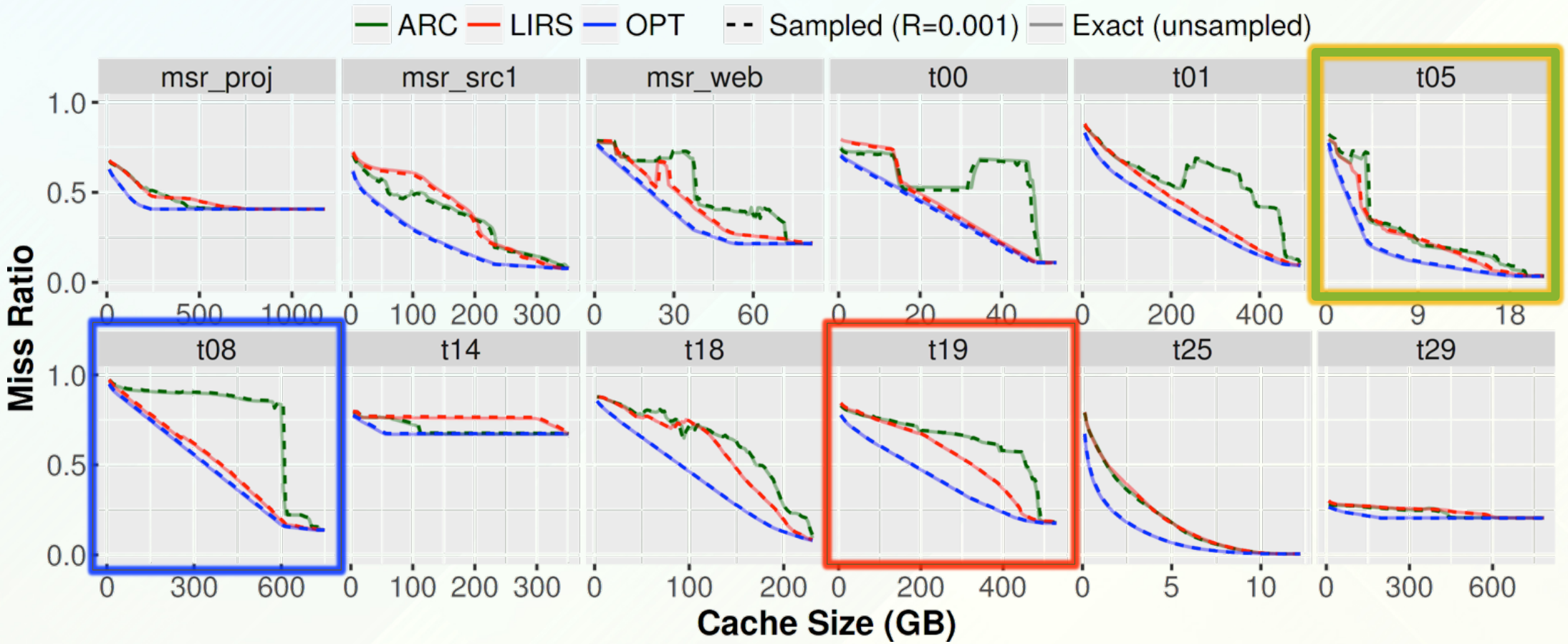
UNDERSTANDING MODEL-BASED ADAPTATION



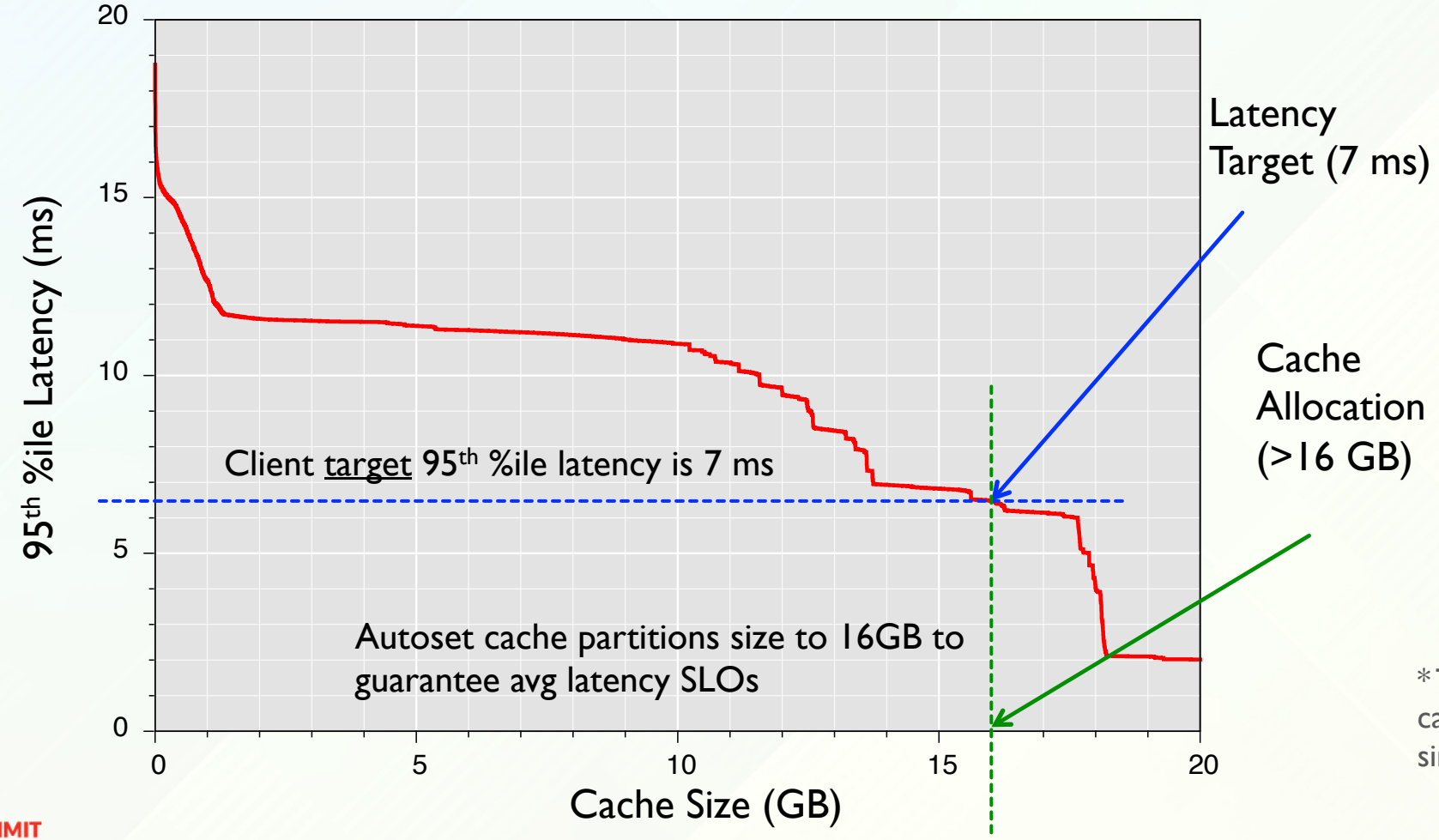
Single Workload. Prediction of performance under different policies.

An self-learning data infrastructure would always pick the optimal.

SAMPLE MODELS FROM PRODUCTION WORKLOADS

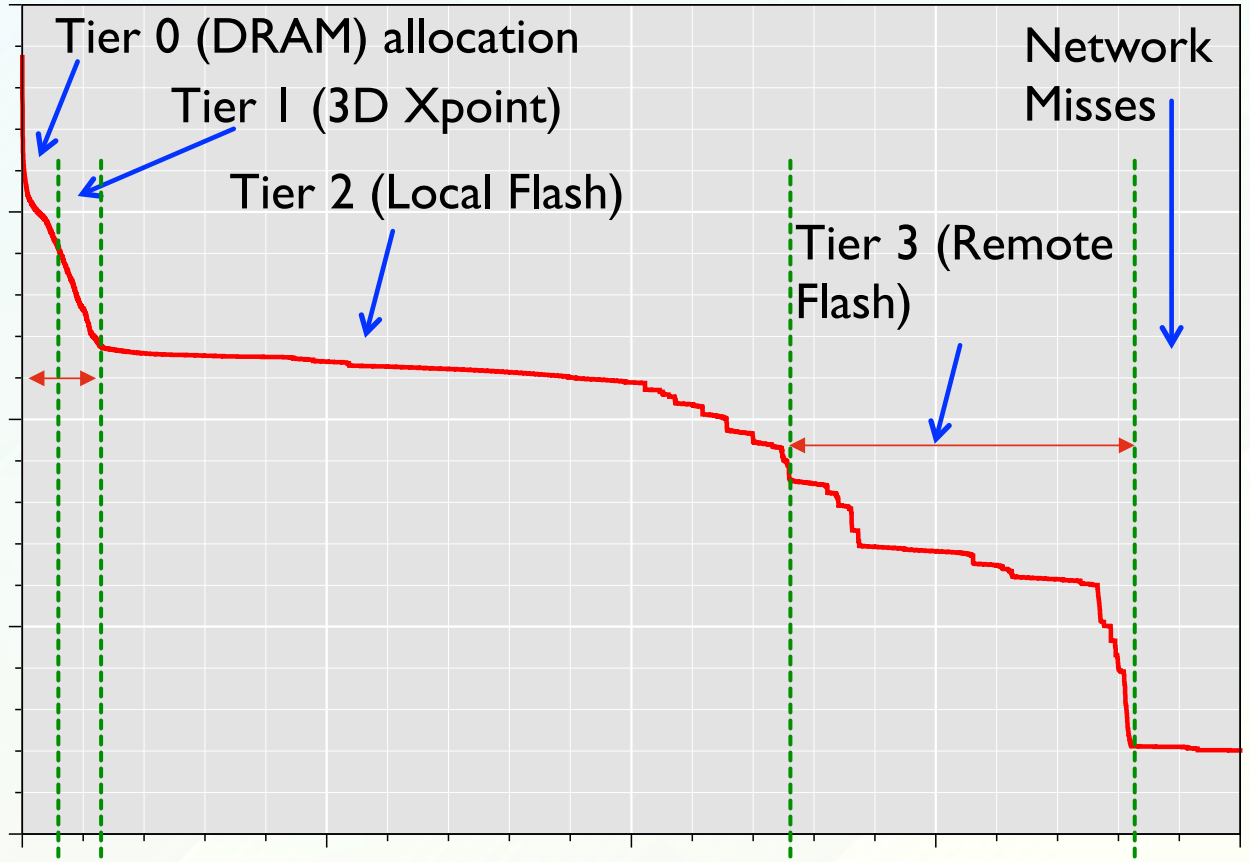


ACHIEVING LATENCY TARGETS



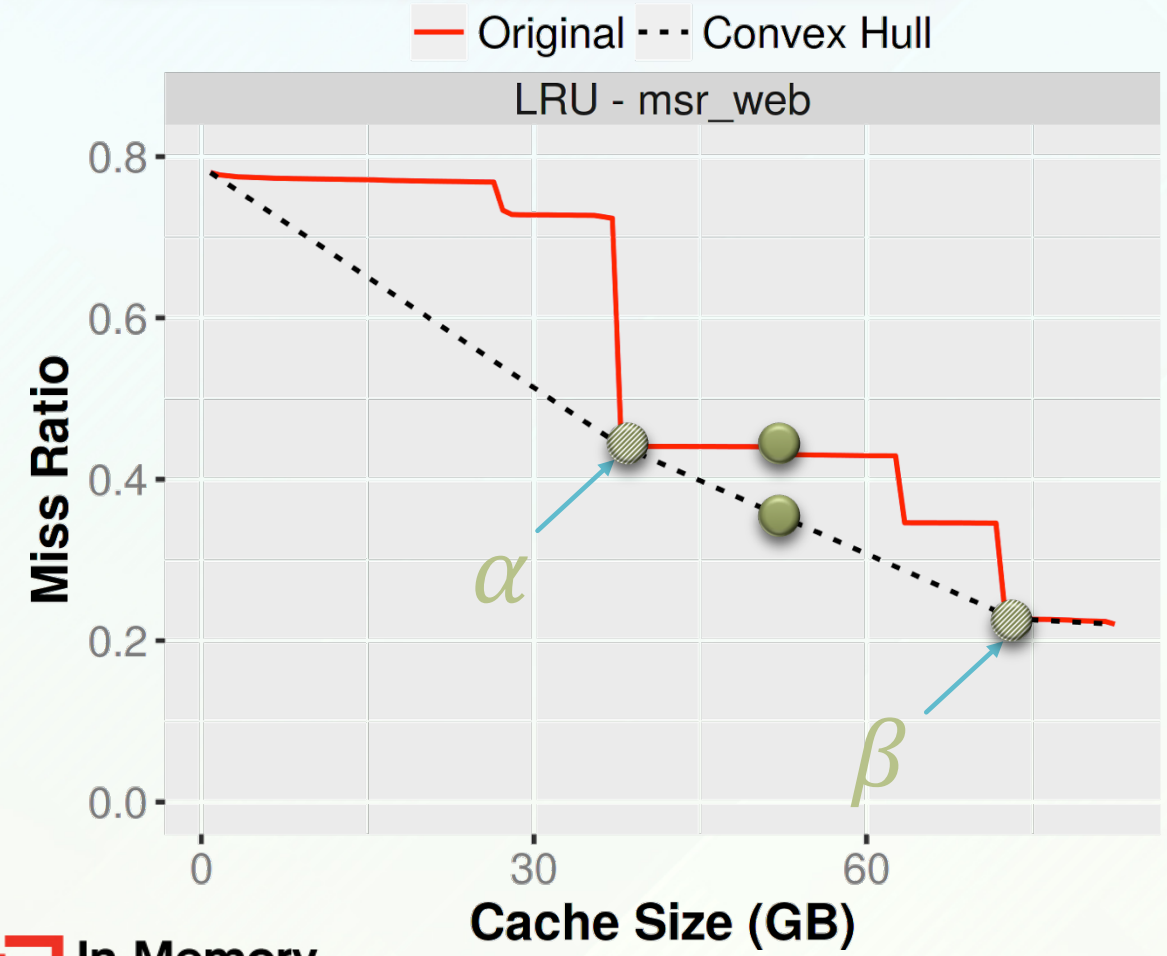
*Throughput targets can be implemented similarly

ACHIEVEING MULTI-TIER SIZING



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

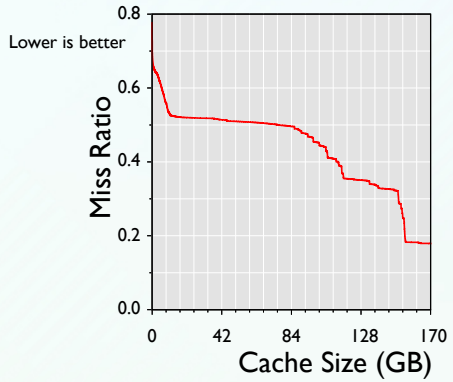
ACHIEVING NEW LEVELS PERFORMANCE



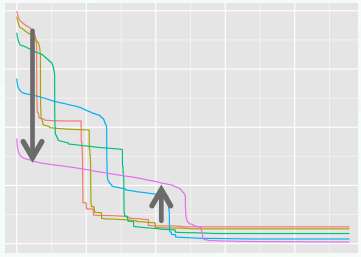
- Thrash remediation algorithm
- Optimal curve bending cache-unfriendly workloads

TOWARDS A SELF-OPTIMIZING DATA PATH

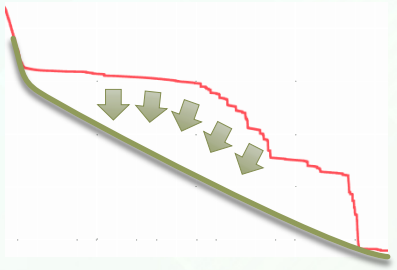
Monitoring



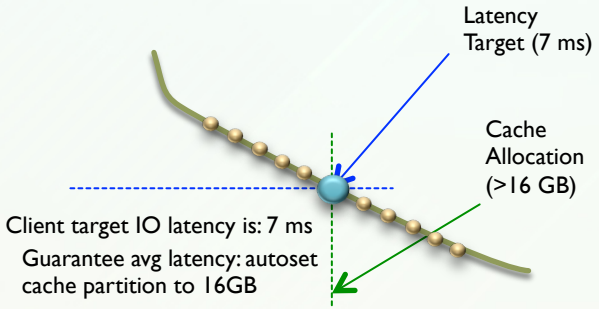
Auto-Select Policies
(dynamic parameters)



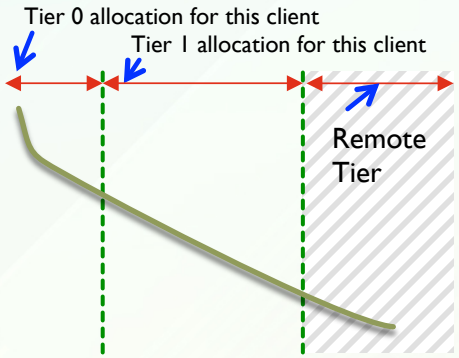
Latency Reduction
(Thrashing Remediation)



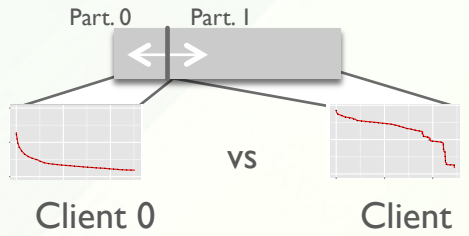
Latency Guarantees



Accurate Tiering



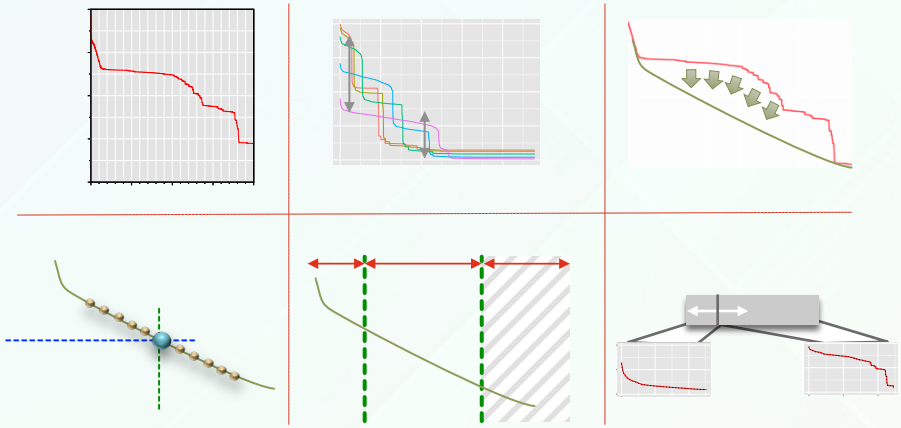
Multi-Tenant Isolation



Results:

- Safely quantify impact of changes
- Often 50-150% cache efficiency improvements ($$)
- Latency SLAs met
- Fewer production fire fights
- Higher consolidation ratios
- Accurate Capacity Planning

In-Memory Computing SUMMIT 2017



CachePhysics

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