

SNIA DEVELOPER CONFERENCE



By Developers FOR Developers

Hyatt Regency Santa Clara, CA  
September 15-17, 2025

A decorative graphic consisting of a series of dots forming a wave that starts as a thin purple line on the left and transitions through yellow and white to a light blue wave on the right.

# ~~Total Cost and Performance of SSDs~~

Andy Banta, Magnition

Scott Shadley, Solidigm

[www.sniadeveloper.org](http://www.sniadeveloper.org)

SNIA DEVELOPER CONFERENCE



By Developers FOR Developers

Hyatt Regency Santa Clara, CA  
September 15-17, 2025

A decorative graphic consisting of a series of dots forming a wavy line that starts as a solid purple line on the left and transitions into a dotted pattern of yellow, orange, and light blue dots on the right.

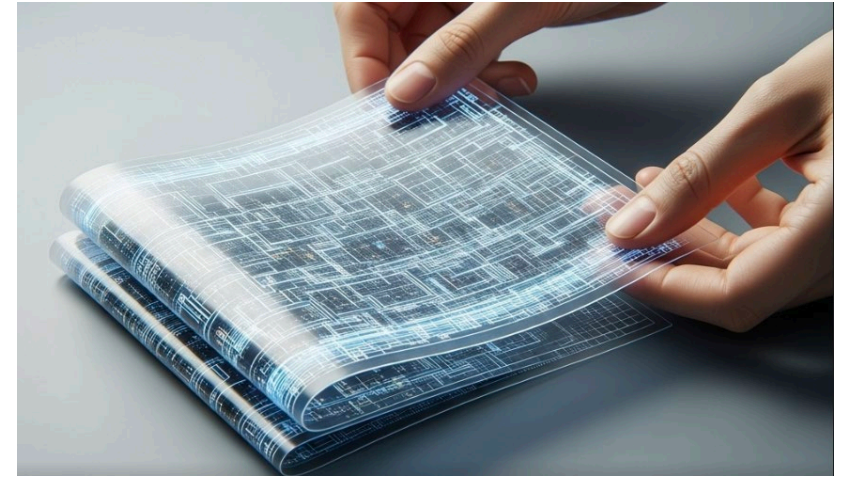
# You're Paying Attention to the Wrong Things

When Andy and Scott Find Time to Make Cool Slides

[www.sniadeveloper.org](http://www.sniadeveloper.org)

# You're paying attention to the wrong things

- Technology fabric folds like laundry
- Apples to orangutans comparisons
- Meyers junctions leak data
- Fastest path cartography simplified



Paying attention to the wrong thing:

# Press Releases rather than existing product

➤ Making noise is one thing, presenting real products is another story

➤ Announcements doesn't relate to production environments

**Perpetual Dawn**  
October 27, 2077

Groundbreaking artificial general intelligence platform  
Unify all digital experiences

- Sentient algorithms  
Quantum-entangled  
ont vetroveniding
- Quantum-entagled  
data streams
- reality-blending user interfaces  
reodending user interfaces

**"Perpetual Dawn is not just a product;  
is is the next stage of human evolution."**  
**Dr. Aris Thorne**

Availability subject to uneretaseable technological brakkwanuans and  
rewriting of fundamental of physics.

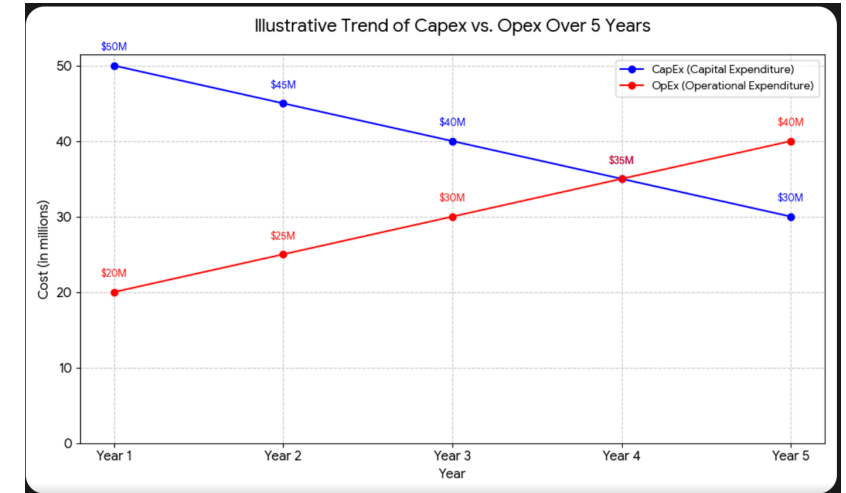
**Dell PowerScale Doubles Density  
with Solidigm 122TB SSDs**

**SOLIDIGM. + DELL Technologies**

Paying attention to the wrong thing:

## Price/GB rather than Total Costs

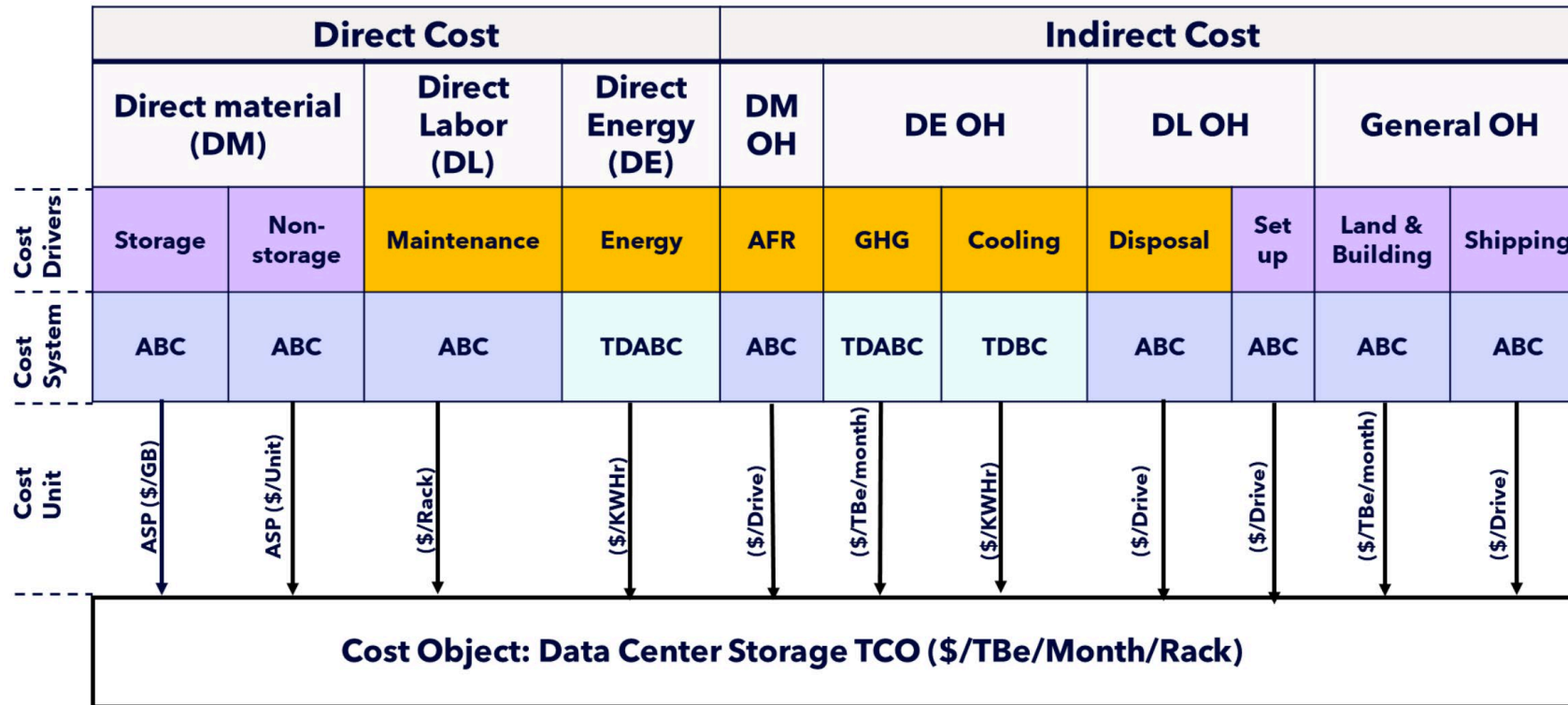
- Purchasing decisions made on Bad Math
- \$/GB is a static measure, TCO Matters
- Since 2000, US average power rates have increased 2.85% annually
  - Compounded over 5-years, that's 11.5%



Paying attention to the wrong thing:

# Price/GB rather than Total Costs

## Cost system in TCO model



	CapEx
	OpEx
	ABC (Activity based cost)
	TDABC (Time Driven ABC)

© 2024 Solidigm. All rights reserved.

Paying attention to the wrong thing:

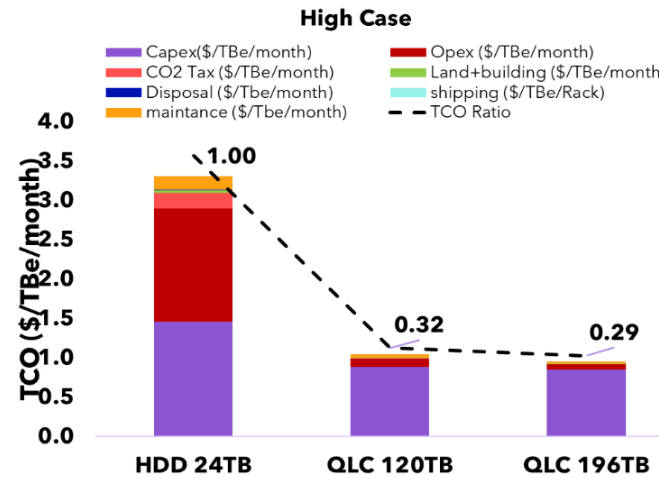
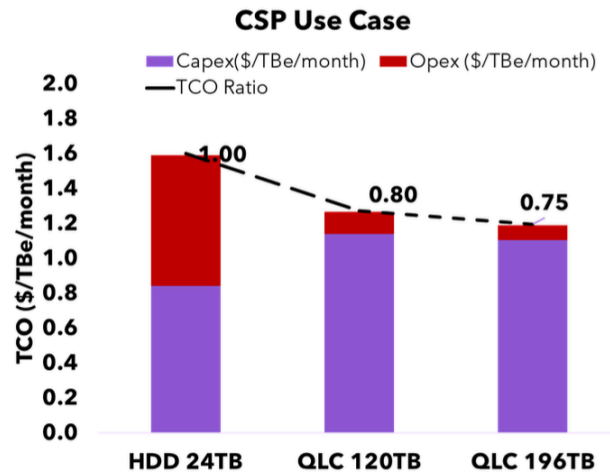
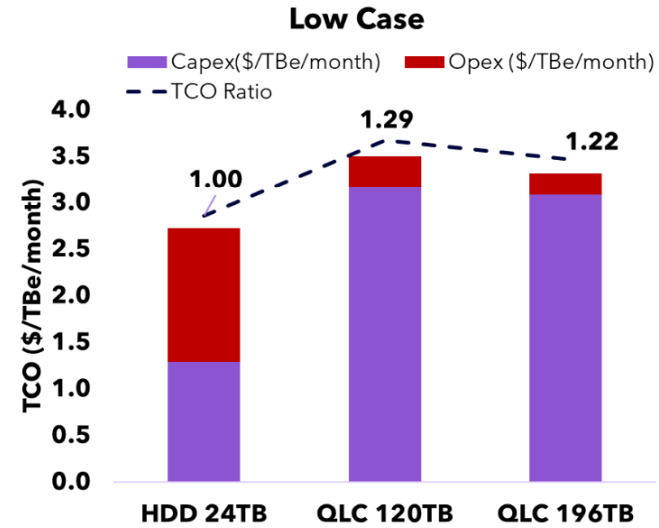
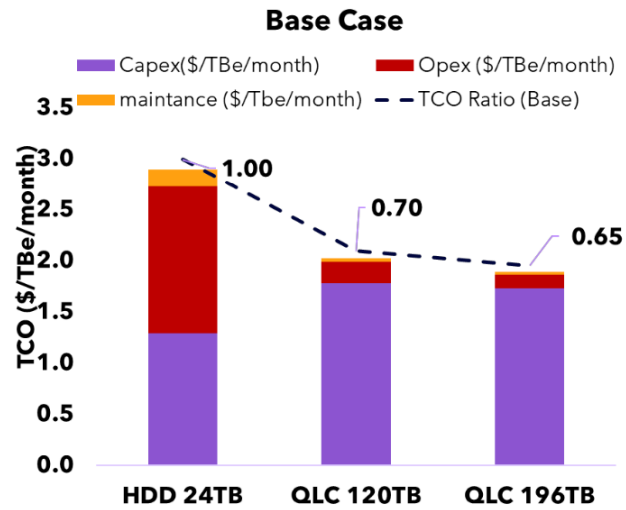
# Price/GB rather than Total Costs

## Changes in TCO model

Cost Pools	Change summary	SNIA Model	This Model	Notes
<b>CAPEX</b>	HW configuration	Static	Dynamic	More reliable model
	Shipping Cost		✓	Favors High density and lower weight SSDs
	Land & Building Cost		✓	
	Drive's Replacement cycle		✓	3/5/7 yr. Drive replacement
	Others	✓	✓	ASP, Drive Density
<b>OPEX</b>	\$/KWh Idle Power Active Power Activity Factors	✓	✓	
<b>Advanced OPEX</b>	Workload Mix		✓	Higher Perf Tepid favors QLC
	AFR	Static	Dynamic	1.3% per 1% AFR
	TVM (Time Value of money)		✓	12% compounded monthly
	Maintenance cost		✓	
	Disposal Cost		✓	
	GHG TAX		✓	Average ~50\$/lb
<b>Choice of Architecture</b>	In Line Data Reduction	✓	✓	
	Redundancy (RAID)	✓	✓	

© 2024 Solidigm. All rights reserved.

# Paying attention to the wrong thing: Price/GB rather than Total Costs

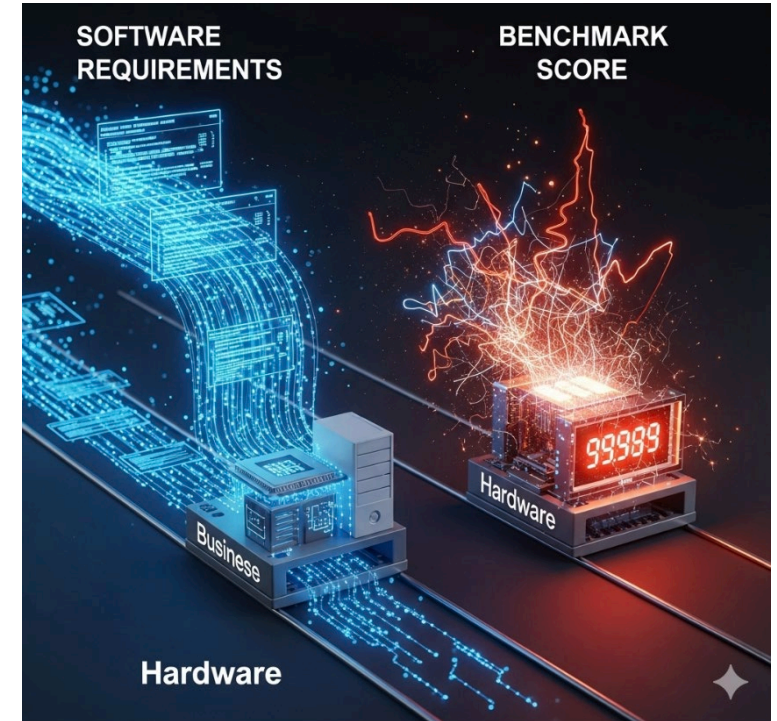


© 2024 Solidigm. All rights reserved.

Paying attention to the wrong thing:

# Purchases made on hardware specs rather than design

- Performance decisions made around hardware specs
  - Software Requirements drive hardware, not the other way around
  - Running benchmarks is not the purpose of your business
  - Hardware is often tuned to perform well on benchmarks



Paying attention to the wrong thing:

## Purchases made on hardware specs rather than design

- Performance decisions made around hardware specs
  - Not based on Software Requirements
  - Running benchmarks is not the purpose of your business
  - Hardware is often tuned to perform well on benchmarks

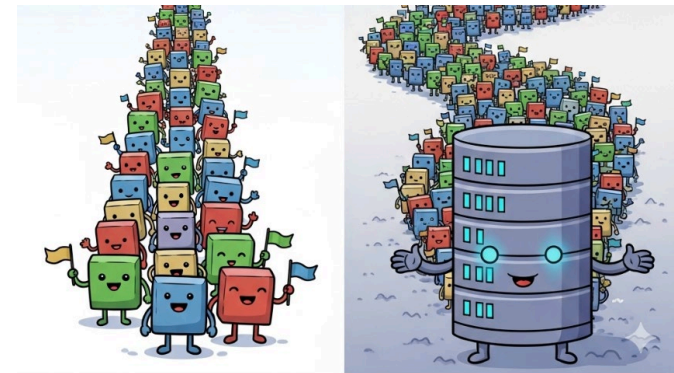
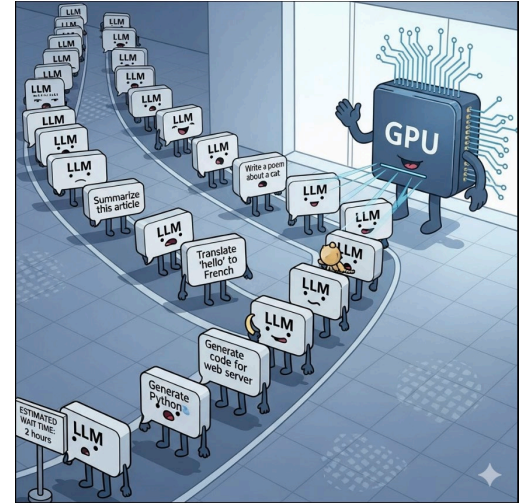


**2012 Volkswagen Golf TDI**

Paying attention to the wrong thing:

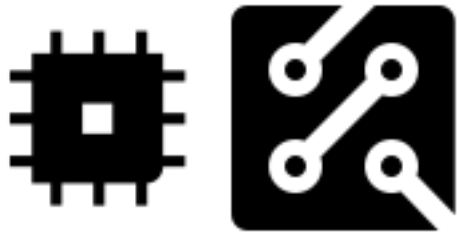
# Purchases made on hardware specs rather than design

- Off-the-shelf hardware assures resource usage
  - Pre-made systems will have idle resources and are likely bought in excessive quantities.
  - Current LLM models incapable of saturating existing GPU systems
- There's always a bottleneck
  - Design Focus at the system level to determine as little pain as possible
  - SATA vs NVMe queue depth
- HW sales vendors aren't interested in the intricacies of a solution
  - If you haven't sized the problem, they won't either



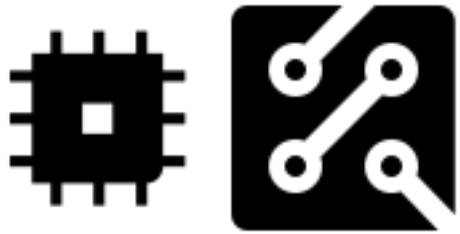


# Engineering Means Simulation



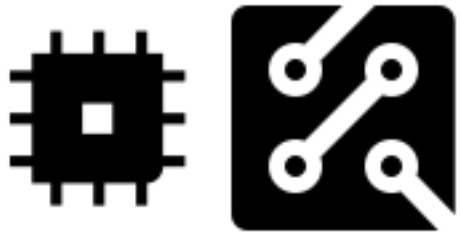


# Engineering Means Simulation



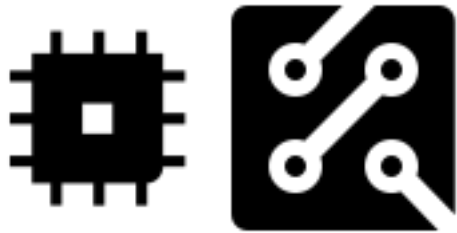


# Engineering Means Simulation



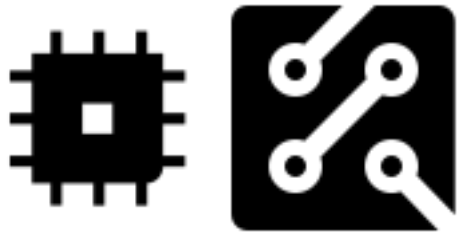


# Engineering Means Simulation





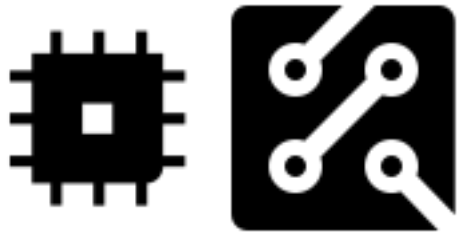
# Engineering Means Simulation



Did you hear about  
the bridge built by  
Beer?



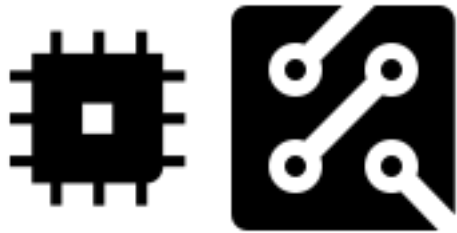
# Engineering Means Simulation





# Engineering Means Simulation

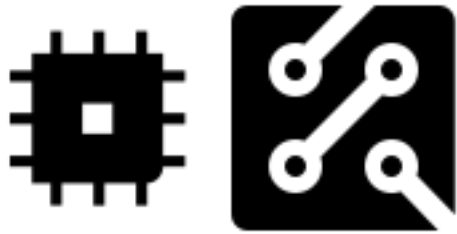
- Cheaper, faster, more flexible than system building





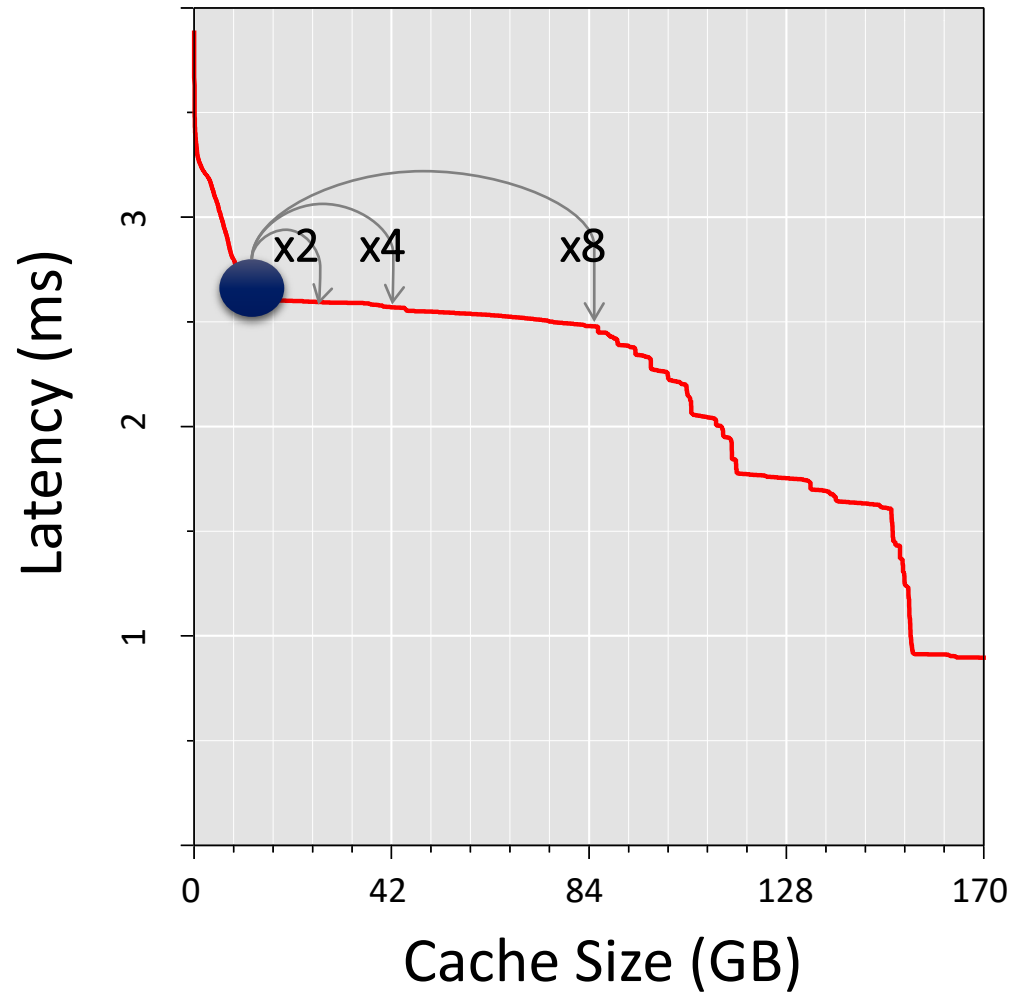
# Engineering Means Simulation

- Cheaper, faster, more flexible than system building
- Engineering design uses simulations, why not software?



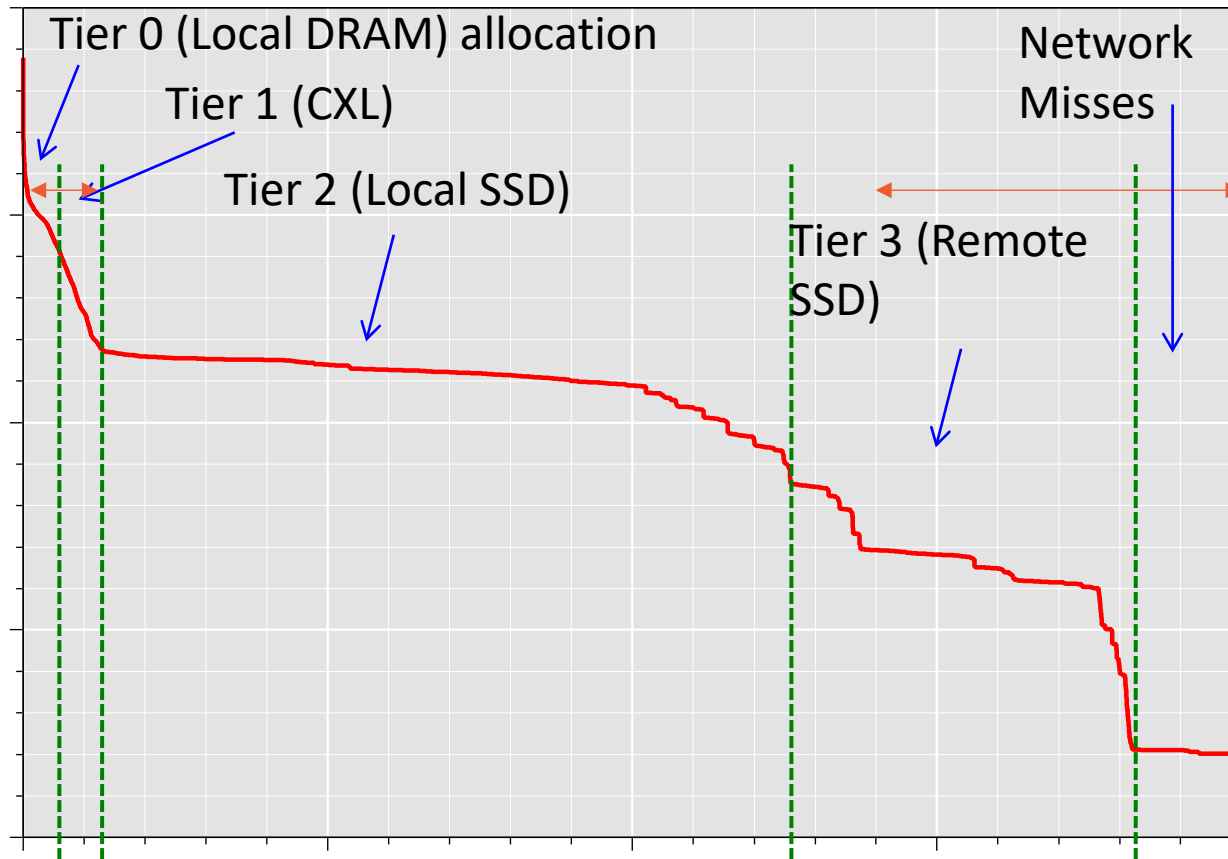
# Understanding Tiering Models

Lower is better



Simulated workloads demonstrate without expensive testing

# Multi-Tier Sizing



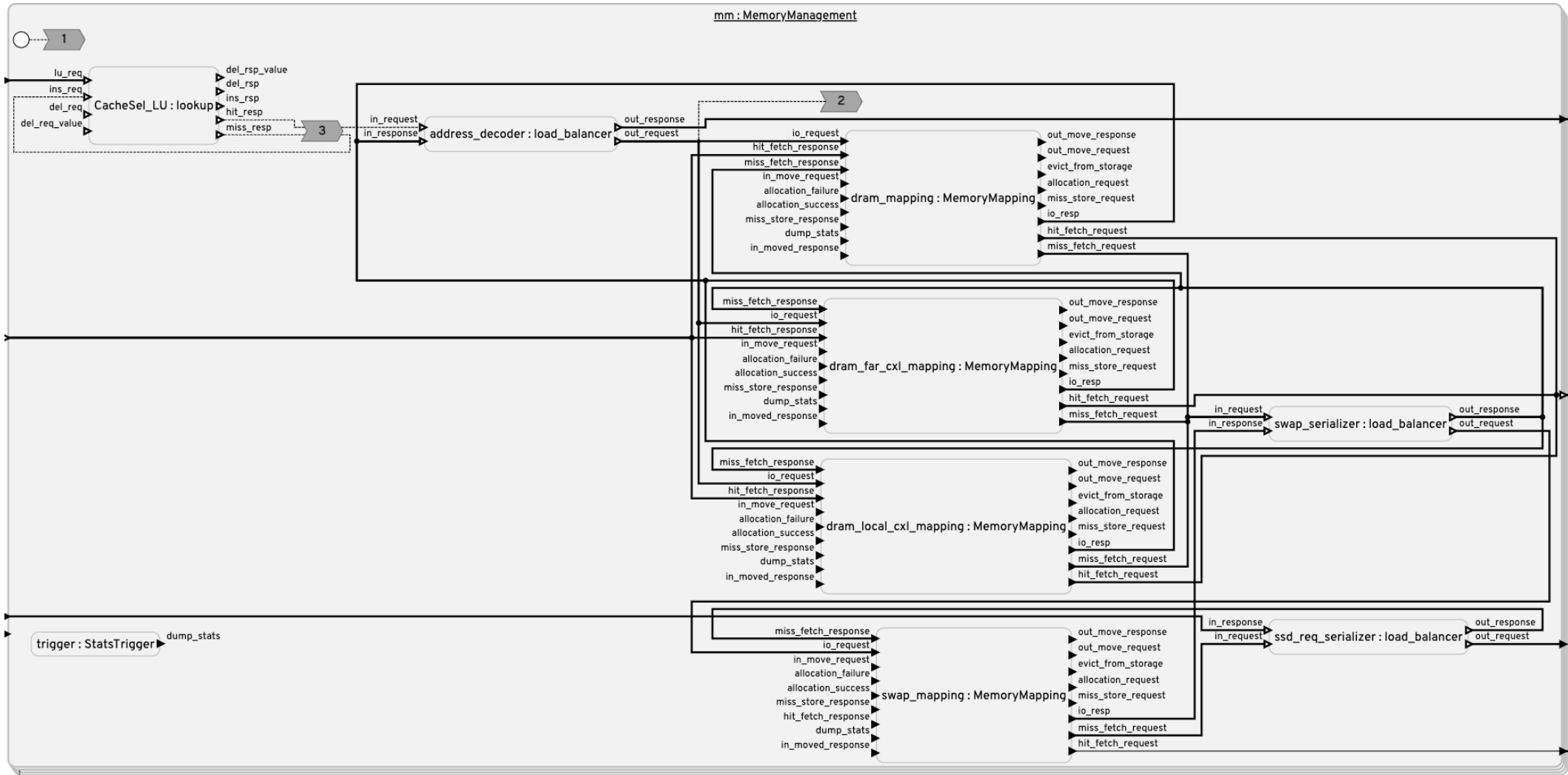
\* Can model network bandwidth as a function of cache misses from each tier

# Simulation Code

- UI generated from code
- Code simulates component

```
reaction (address_decoder.out_request) {=  
    for (int i = 0; i < self->n_DRAMs; ++i) {  
        if (address_decoder.out_request[i]->is_present) {  
            LOG_TRACE ( self->log_level, "(%lld, %u) physical_time:%lld "  
                "%s sending out address_decoder.out_request at port:%d",  
                lf_time_logical_elapsed(), lf_tag().microstep, lf_time_physical_elapsed(),  
                self->name, i  
            );  
        }  
    }  
}=}  
  
reaction (CacheSel_LU.hit_resp, CacheSel_LU.miss_resp) -> address_decoder.in_request, CacheSel_LU.ins_req {=  
    if (CacheSel_LU.hit_resp[0]->is_present) {  
        cxl_request_t *req = &CacheSel_LU.hit_resp[0]->value;  
        LOG_TRACE ( self->log_level, "(%lld, %u) physical_time:%lld "  
            "%s LU found req_id:%u lba:%lu obj_size:%u ts:%llu lb_hint:%d",  
            lf_time_logical_elapsed(), lf_tag().microstep, lf_time_physical_elapsed(),  
            self->name, req->req_id, req->lba, req->obj_size, req->ts, req->lb_hint  
        );  
        lf_set (address_decoder.in_request[0], *req);  
    } else if (CacheSel_LU.miss_resp[0]->is_present) {  
        cxl_request_t *req = &CacheSel_LU.miss_resp[0]->value;  
        LOG_TRACE ( self->log_level, "(%lld, %u) physical_time:%lld "  
            "%s LU not found req_id:%u lba:%lu obj_size:%u ts:%llu",  
            lf_time_logical_elapsed(), lf_tag().microstep, lf_time_physical_elapsed(),  
            self->name, req->req_id, req->lba, req->obj_size, req->ts  
        );  
        lf_set (address_decoder.in_request[0], *req);  
        lf_set (CacheSel_LU.ins_req[0], *req);  
    }  
}=}  
  
dram_mapping.hit_fetch_request, dram_local_cxl_mapping.hit_fetch_request, dram_far_cxl_mapping.hit_fetch_request -> fetch_from_mem;  
rsp_from_mem -> dram_mapping.hit_fetch_response, dram_local_cxl_mapping.hit_fetch_response, dram_far_cxl_mapping.hit_fetch_response;  
  
io_request -> CacheSel_LU.lu_req;  
address_decoder.out_request -> dram_mapping.io_request, dram_local_cxl_mapping.io_request, dram_far_cxl_mapping.io_request;  
dram_mapping.io_resp, dram_local_cxl_mapping.io_resp, dram_far_cxl_mapping.io_resp -> address_decoder.in_response;  
address_decoder.out_response -> io_response;  
  
dram_mapping.miss_fetch_request, dram_local_cxl_mapping.miss_fetch_request, dram_far_cxl_mapping.miss_fetch_request -> swap_serializer.in_request;  
swap_serializer.out_request -> swap_mapping.io_request;  
swap_mapping.io_resp -> swap_serializer.in_response;  
swap_serializer.out_response -> dram_mapping.miss_fetch_response, dram_local_cxl_mapping.miss_fetch_response, dram_far_cxl_mapping.miss_fetch_response;  
  
swap_mapping.miss_fetch_request -> ssd_req_serializer.in_request;  
ssd_req_serializer.out_request -> miss_fetch_request;  
miss_fetch_response -> ssd_req_serializer.in_response;  
ssd_req_serializer.out_response -> swap_mapping.miss_fetch_response;  
  
swap_mapping.hit_fetch_request -> fetch_from_swap;
```

# Graphical representation



# Variables

```
reactor MemoryManagement ( bank_index:int = 0, name:string = "MMU", dram_size:uint64_t = 4096, swap_size:uint64_t = 4096, dram_page_size:uint64_t = 4096,
                           swap_page_size:uint64_t = 4096, n_DRAMs:int = 3, n_Swaps:int = 1, log_level:int = {=LOG_DEBUG_LEVEL=} ) {
    input io_request:cxl_request_t;
    output io_response:cxl_response_t;

    output miss_fetch_request:cxl_request_t;
    input miss_fetch_response:cxl_response_t;

    output [3] fetch_from_mem:cxl_request_t;
    input [3] rsp_from_mem:cxl_response_t;

    output fetch_from_swap:cxl_request_t;
    input rsp_from_swap:cxl_response_t;

    CacheSel_LU = new lookup<cxl_request_t, cxl_request_t, cxl_request_t> (name = name, n_ports = 1, log_level = log_level);

    dram_mapping = new
    MemoryMapping(warm_up_epochs = 0, name = "DRAM_MAPPING",
                 store_on_miss = false, store_from_origin = true,
                 move_on_hit_source = false, move_on_hit_sink = false,
                 move_on_hit_count = 1, ttl_time_sec = 3600,
                 ttl_schedule_time_sec = -1, evict_methods = {=&lru_eviction_methods=},
                 eviction_type = {=LRU=}, n_ports = 1, cache_size = dram_size, max_cachable_obj = 10485760,
                 page_size = dram_page_size, log_level = log_level);

    dram_local_cxl_mapping = new
    MemoryMapping(warm_up_epochs = 0, name = "DRAM_LOCAL_CXL_MAPPING",
                 store_on_miss = false, store_from_origin = true,
                 move_on_hit_source = false, move_on_hit_sink = false,
                 move_on_hit_count = 1, ttl_time_sec = 3600,
                 ttl_schedule_time_sec = -1, evict_methods = {=&lru_eviction_methods=},
                 eviction_type = {=LRU=}, n_ports = 1, cache_size = dram_size, max_cachable_obj = 10485760,
                 page_size = dram_page_size, log_level = log_level);

    dram_far_cxl_mapping = new
    MemoryMapping(warm_up_epochs = 0, name = "DRAM_FAR_CXL_MAPPING",
                 store_on_miss = false, store_from_origin = true,
                 move_on_hit_source = false, move_on_hit_sink = false,
                 move_on_hit_count = 1, ttl_time_sec = 3600,
                 ttl_schedule_time_sec = -1, evict_methods = {=&lru_eviction_methods=},
                 eviction_type = {=LRU=}, n_ports = 1, cache_size = dram_size, max_cachable_obj = 10485760,
                 page_size = dram_page_size, log_level = log_level);

    swap_mapping = new
    MemoryMapping(warm_up_epochs = 0, name = "SWAP_MAPPING",
                 store_on_miss = false, store_from_origin = true,
                 move_on_hit_source = false, move_on_hit_sink = false,
                 ttl_time_sec = 3600, ttl_schedule_time_sec = -1,
                 evict_methods = {=&lru_eviction_methods=}, eviction_type = {=LRU=}, n_ports = 1,
                 cache_size = swap_size, max_cachable_obj = 10485760, page_size = swap_page_size, log_level = log_level);

    address_decoder = new
    load_balancer<cxl_request_t, cxl_response_t> (name = "ADDRESS_DECODER",
        selection_methods = {=&url_hint_lb_methods=}, selection = {=USR_DEF_SELECTION=}, n_inputs = 1,
        n_outputs = 3, def_output = 0, log_level = log_level);

    trigger = new StatsTrigger (name = name, epoch_time = 3600, epoch_unit = {=IN_SECONDS=}, log_level = log_level);
```

# Variables

```
79 storage_capacity:
80   value: [549755813888]
81 cache_size_ratio:
82   value: [0.001, 0.005, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5]
83 store_on_miss:
84   value: ["false"]
85 store_from_origin:
86   value: ["true"]
87 move_on_hit_source:
88   value: ["false"]
89 move_on_hit_sink:
90   value: ["false"]
91 move_on_hit_count:
92   value: [1]
93 eviction_types:
94   value: [FIFO, SIEVE, CLOCK, LRU]
95 eviction_methods:
96   value: ["&lru_eviction_methods"]
97 page_size:
98   value: [4096]
99 l2_server_lb:
100  name: L2_SERVER_LOADBALANCER
101 selection_types:
102   value: [USR_DEF_SELECTION]
103 selection_methods:
104   value: ["&url_hash_lb_methods"]
105 l2_servers:
106   value: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30]
107   name: L2_SERVER
108   switch:
109     value: [enable]
110   storage_media:
111     value: [ssd_lite]
112   storage_capacity:
113     value: [549755813888]
114 store_on_miss:
115   value: ["false"]
116 store_from_origin:
117   value: ["true"]
118 move_on_hit_source:
119   value: ["false"]
120 move_on_hit_sink:
121   value: ["false"]
122 eviction_types:
123   value: [FIFO, SIEVE, CLOCK, LRU]
124 eviction_methods:
125   value: ["&lru_eviction_methods"]
126 page_size:
127   value: [4096]
128 pop_serializer:
129   name: POP_SERIALIZER
130 data_mover:
131   name: POP_L1_TO_L2_MOVER
132   switch:
133     value: [disable]
```

Wiki  
Topology

```
79 storage_capacity:
80   value: [549755813888]
81 cache_size_ratio:
82   value: [0.001, 0.005, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5]
83 store_on_miss:
84   value: ["false"]
85 store_from_origin:
86   value: ["true"]
87 move_on_hit_source:
88   value: ["true"]
89 move_on_hit_sink:
90   value: ["false"]
91 move_on_hit_count:
92   value: [1, 2, 3, 4]
93 eviction_types:
94   value: [FIFO, SIEVE, CLOCK, LRU]
95 eviction_methods:
96   value: ["&lru_eviction_methods"]
97 page_size:
98   value: [4096]
99 l2_server_lb:
100  name: L2_SERVER_LOADBALANCER
101 selection_types:
102   value: [LB_1_ON_1_WIRING_DEFAULT]
103 selection_methods:
104   value: ["&url_hash_lb_methods"]
105 l2_servers:
106   value: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30]
107   name: L2_SERVER
108   switch:
109     value: [enable]
110   storage_media:
111     value: [ssd_lite]
112   storage_capacity:
113     value: [549755813888]
114 store_on_miss:
115   value: ["false"]
116 store_from_origin:
117   value: ["false"]
118 move_on_hit_source:
119   value: ["false"]
120 move_on_hit_sink:
121   value: ["true"]
122 eviction_types:
123   value: [FIFO, SIEVE, CLOCK, LRU]
124 eviction_methods:
125   value: ["&lru_eviction_methods"]
126 page_size:
127   value: [4096]
128 pop_serializer:
129   name: POP_SERIALIZER
130 data_mover:
131   name: POP_L1_TO_L2_MOVER
132   switch:
133     value: [enable]
```

Cloudflare  
Topology

# Variables – Drive Duty Cycle

```
capacity_goal_in_pb      120 PB
empty_rack_cost.         $1100
hdd_esp_per_gb_per_year 0.012
ssd_esp_per_gb_per_year 0.045
power_unit_efficiency_factor 1.15
hdd_capacity_purchasing_multiplier 1.2
hdd_deployment_term     5
hdd_duty_cycle.         [ 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 ]
ssd_capacity_purchasing_multiplier 1.2
ssd_deployment_term     7
ssd_name                 Solidigm 128TB E3.s on 17G", "Solidigm 64TB E3.S on 17G", "Solidigm
128TB U.2 on F910", "Nearline 24TB on A300", "Nearline 24TB on F910"
```

# Variables – The Costs

```

capacity_goal_in_pb      120 PB
empty_rack_cost.        $1100
hdd_esp_per_gb_per_year 0.012
ssd_esp_per_gb_per_year 0.045
power_unit_efficiency_factor 1.15
hdd_capacity_purchasing_multiplier 1.2
hdd_deployment_term    5
hdd_duty_cycle.        [ 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 ]
ssd_capacity_purchasing_multiplier 1.2
ssd_deployment_term    7
ssd_name                Solidigm 128TB E3.s on 17G", "Solidigm 64TB E3.S on 17G", "Solidigm
128TB U.2 on F910", "Nearline 24TB on A300", "Nearline 24TB on F910"
hdd_name                "Seagate 24TB SATA HDD", "Seagate 20TB SATA HDD on ebay server"
    
```

Capacity	Avg Active Power	Idle Power
24,000GB	8.4W	5.5W
20,000GB	8.4W	5.5W
18,000GB	8.4W	5.5W
30,000GB	9.5W	5.5W
24,000GB	9.5W	5.5W
24,000GB	9.5W	5.5W
24,000GB	9.5W	5.5W
24,000GB	9.5W	5.5W
30,000GB	9.5W	5.5W
14,000GB	9.5W	5.5W

Capacity	Avg Active Power	Idle Power
128,000GB	15.0W	3.5W
64,000GB	8.5W	3.5W
128,000GB	15.0W	3.5W
24,000GB	4.0W	1.5W
24,000GB	4.0W	1.5W
122,880GB	17.0W	5.0W
122,880GB	17.0W	5.0W

# Variables – The Total Costs

X(Config): solidigm\_demo.hdd\_duty\_cycle

Y(Stat):  
config-id  
time  
solidigm\_demo.INTERVAL.HDD\_TCO  
solidigm\_demo.INTERVAL.SSD\_TCO

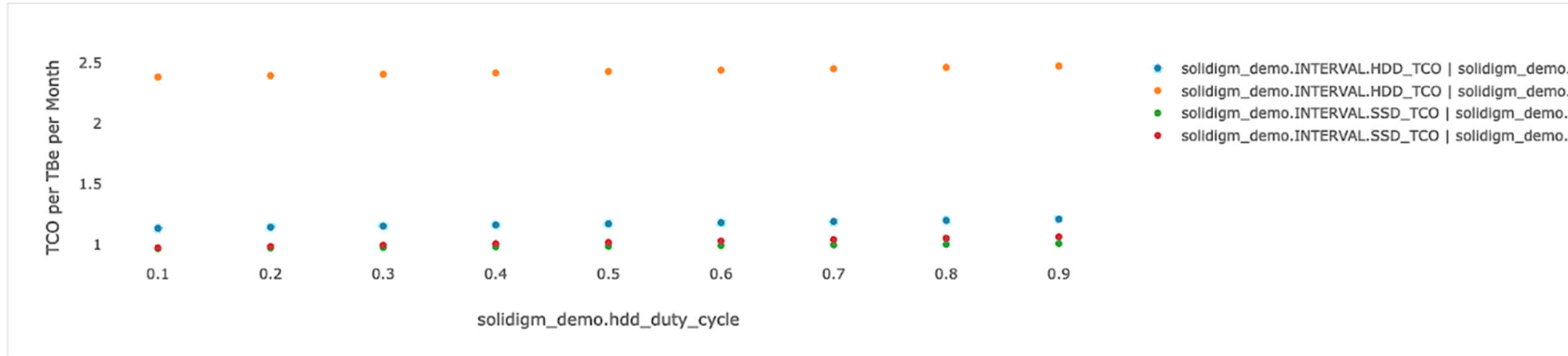
Group by: solidigm\_demo.hdd\_name

Filter: solidigm\_demo.ssd\_name

Solidigm 128TB E3.s on 17G  
Solidigm 64TB E3.S on 17G  
Solidigm 128TB U.2 on F910  
Nearline 24TB on A300

Plot

TCO per TBe per Month:



Paying attention to the wrong thing:

Design decisions are made based on too small a set of data

Paying attention to the wrong thing:

Design decisions are made based on too small a set of data





- ▶ “Choose from these three columns” is not adequate



Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

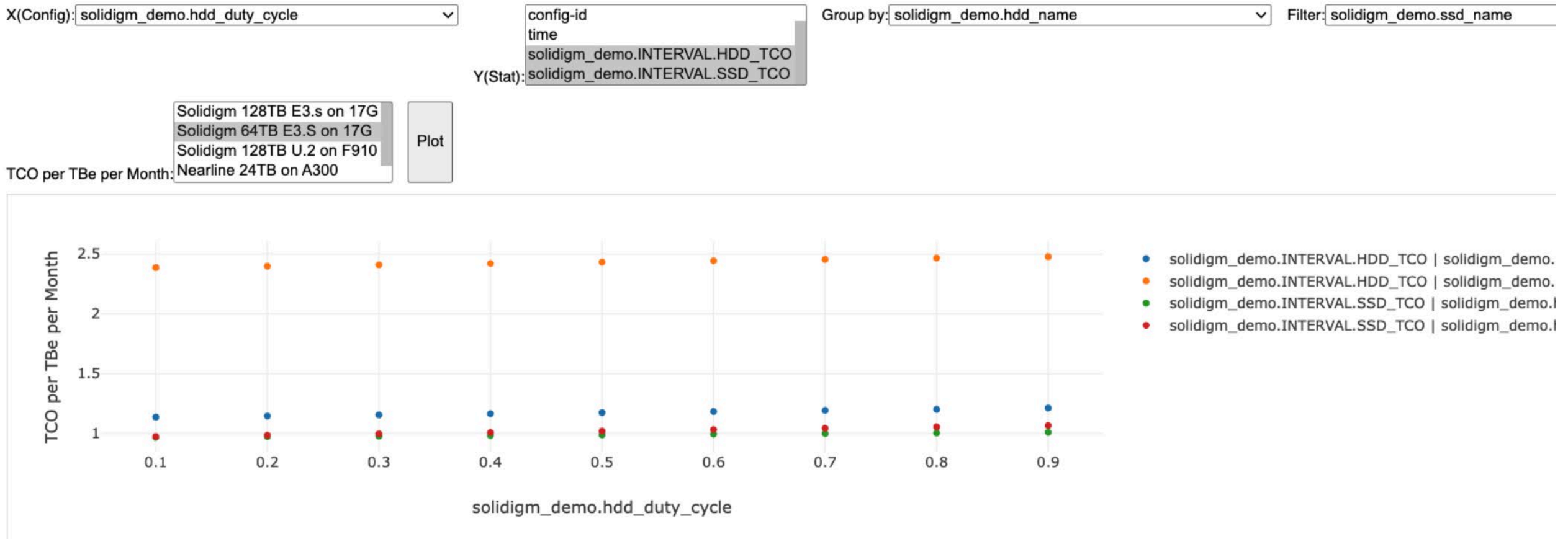
- “Choose from these three columns” is not adequate

<b>Start Fast</b>			
			
<b>Developer System</b>	<b>Small</b>	<b>Medium</b>	<b>Large</b>
2 NVIDIA H100 NVL GPU's	4 or 8 NVIDIA L40S GPUs	8 or 16 NVIDIA L40S GPUs	16 or 32 NVIDIA H100 NVL GPUs
32 TB Integrated	109 TB	217 TB	670 TB
Customer Network	100GbE NVIDIA Networking	200GbE NVIDIA Networking	400GbE NVIDIA Networking
Up to 2.2 kW	up to 8 kW rack	up to 17.7 kW	up to 16.5 kW x 2

Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

➤ Experimentation space should be defined and refined



Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

➤ Experimentation space should be defined and refined

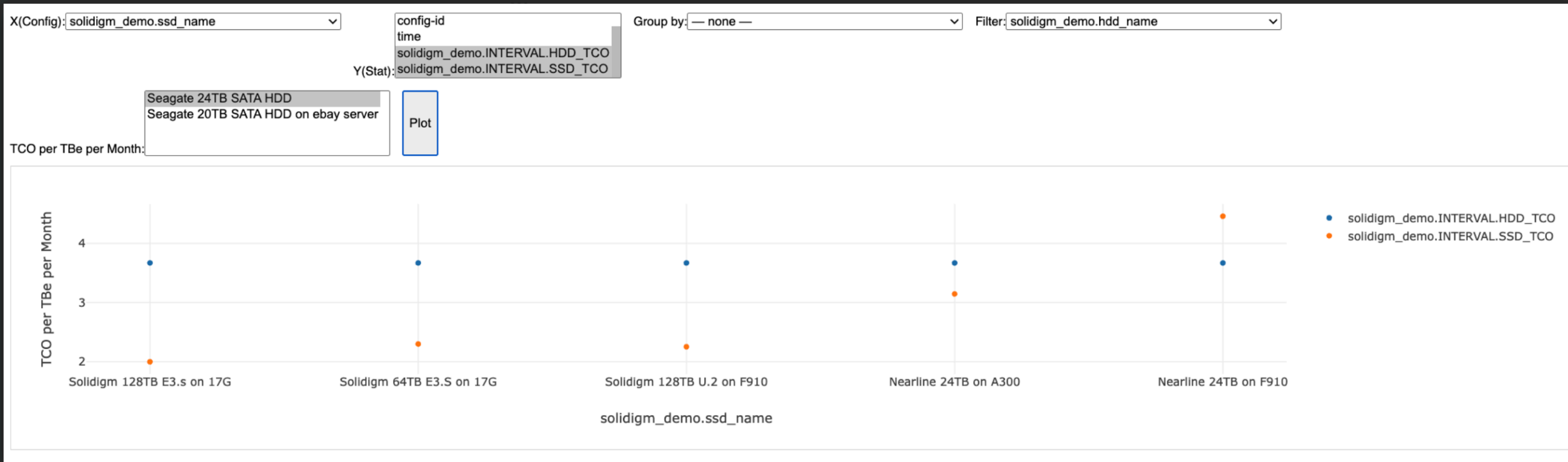
The screenshot shows the Magnition Orchestrator interface. On the left, there is a sidebar with 'APP STRUCTURE' and 'CONFIGS'. The 'CONFIGS' section lists 'base-test', 'csp-test', 'high-test' (selected with a green dot), and 'low-test'. The main area displays 'Configuration: high-test' under the heading 'Homogeneous Configuration'. Below this is a table with three columns: 'Parameter', 'Type', and 'Value'.

Parameter	Type	Value
solidigm_demo.capacity_goal_in_pb	double	120.000000
solidigm_demo.empty_rack_cost	int	1100
solidigm_demo.hdd_asp_per_gb_per_year	double	0.012000
solidigm_demo.ssd_asp_per_gb_per_year	double	0.045000
solidigm_demo.power_unit_efficiency_factor	double	1.150000
solidigm_demo.hdd_inline_data_compression	double	1.0
solidigm_demo.hdd_capacity_utilization	double	0.8
solidigm_demo.hdd_hw_redundancy	double	3
solidigm_demo.hdd_deployment_term	double	4
solidigm_demo.hdd_duty_cycle	double	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
solidigm_demo.ssd_inline_data_compression	double	3
solidigm_demo.ssd_capacity_utilization	double	0.95
solidigm_demo.ssd_hw_redundancy	double	1.14
solidigm_demo.ssd_deployment_term	double	7
solidigm_demo.ssd_name	std::string	"Solidigm 128TB E3.s on 17G", "Solidigm 64TB E3.S on 17G", "Solidigm"

Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

- Experimentation space should be defined and refined



Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

- Experimentation space should be defined and refined

```
79 storage_capacity:
80   value: [549755813888]
81 cache_size_ratio:
82   value: [0.001, 0.005, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5]
83 store_on_miss:
84   value: ["false"]
85 store_from_origin:
86   value: ["true"]
87 move_on_hit_source:
88   value: ["false"]
89 move_on_hit_sink:
90   value: ["false"]
91 move_on_hit_count:
92   value: [1]
93 eviction_types:
94   value: [FIFO, SIEVE, CLOCK, LRU]
95 eviction_methods:
96   value: ["&lru_eviction_methods"]
97 page_size:
98   value: [4096]
99 l2_server_lb:
100  name: L2_SERVER_LOADBALANCER
101 selection_types:
102   value: [USR_DEF_SELECTION]
103 selection_methods:
104   value: ["&url_hash_lb_methods"]
105 l2_servers:
106   value: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30]
107 name: L2_SERVER
108 switch:
109   value: [enable]
110 storage_media:
111   value: [ssd_lite]
112 storage_capacity:
113   value: [549755813888]
114 store_on_miss:
115   value: ["false"]
116 store_from_origin:
117   value: ["true"]
118 move_on_hit_source:
119   value: ["false"]
120 move_on_hit_sink:
121   value: ["false"]
122 eviction_types:
123   value: [FIFO, SIEVE, CLOCK, LRU]
124 eviction_methods:
125   value: ["&lru_eviction_methods"]
126 page_size:
127   value: [4096]
128 pop_serializer:
129   name: POP_SERIALIZER
130 data_mover:
131   name: POP_L1_TO_L2_MOVER
132   switch:
133     value: [disable]
```

Wiki  
Topology

```
79 storage_capacity:
80   value: [549755813888]
81 cache_size_ratio:
82   value: [0.001, 0.005, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5]
83 store_on_miss:
84   value: ["false"]
85 store_from_origin:
86   value: ["true"]
87 move_on_hit_source:
88   value: ["true"]
89 move_on_hit_sink:
90   value: ["false"]
91 move_on_hit_count:
92   value: [1, 2, 3, 4]
93 eviction_types:
94   value: [FIFO, SIEVE, CLOCK, LRU]
95 eviction_methods:
96   value: ["&lru_eviction_methods"]
97 page_size:
98   value: [4096]
99 l2_server_lb:
100  name: L2_SERVER_LOADBALANCER
101 selection_types:
102   value: [LB_1_ON_1_WIRING_DEFAULT]
103 selection_methods:
104   value: ["&url_hash_lb_methods"]
105 l2_servers:
106   value: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 30]
107 name: L2_SERVER
108 switch:
109   value: [enable]
110 storage_media:
111   value: [ssd_lite]
112 storage_capacity:
113   value: [549755813888]
114 store_on_miss:
115   value: ["false"]
116 store_from_origin:
117   value: ["false"]
118 move_on_hit_source:
119   value: ["false"]
120 move_on_hit_sink:
121   value: ["true"]
122 eviction_types:
123   value: [FIFO, SIEVE, CLOCK, LRU]
124 eviction_methods:
125   value: ["&lru_eviction_methods"]
126 page_size:
127   value: [4096]
128 pop_serializer:
129   name: POP_SERIALIZER
130 data_mover:
131   name: POP_L1_TO_L2_MOVER
132   switch:
133     value: [enable]
```

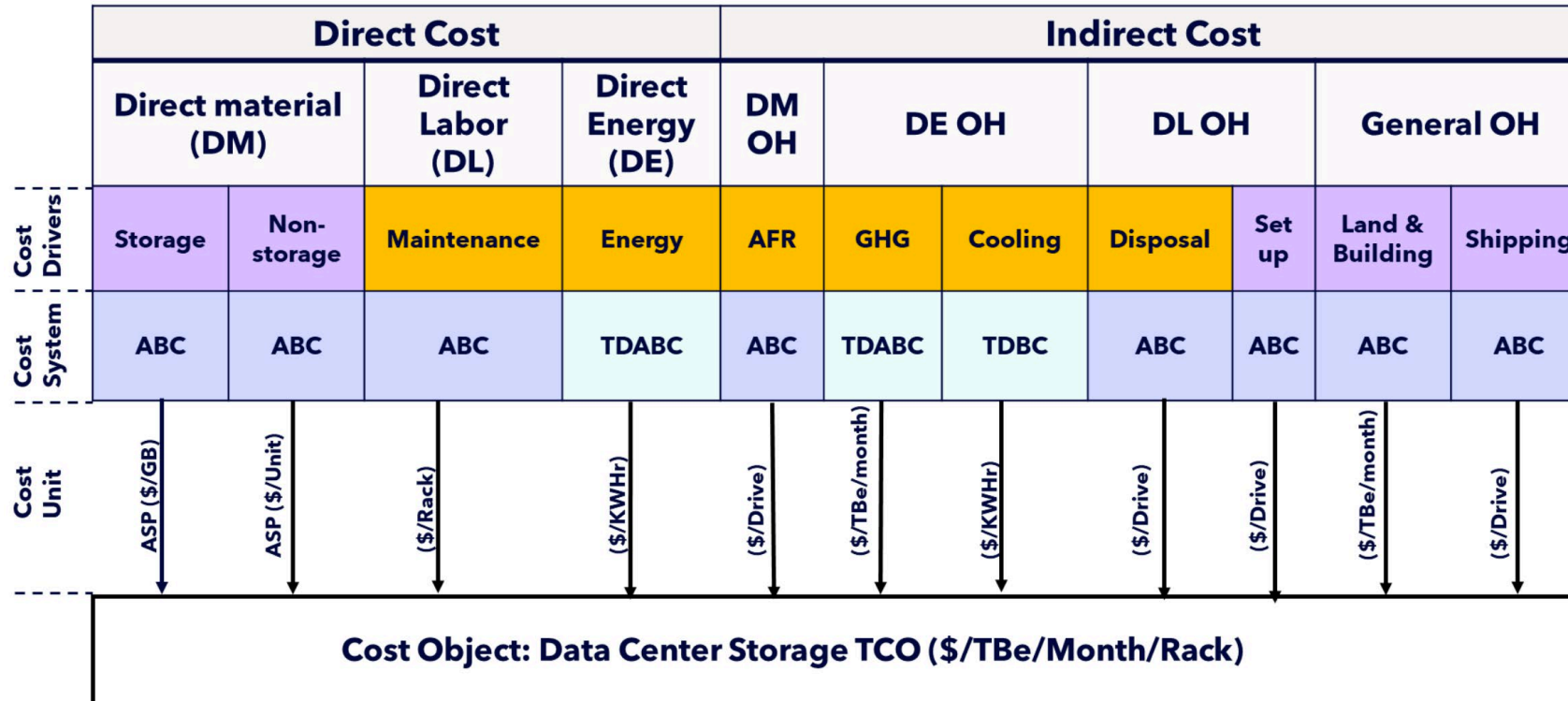
Cloudflare  
Topology

Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

- Experimentation space should be defined and refined

## Cost system in TCO model



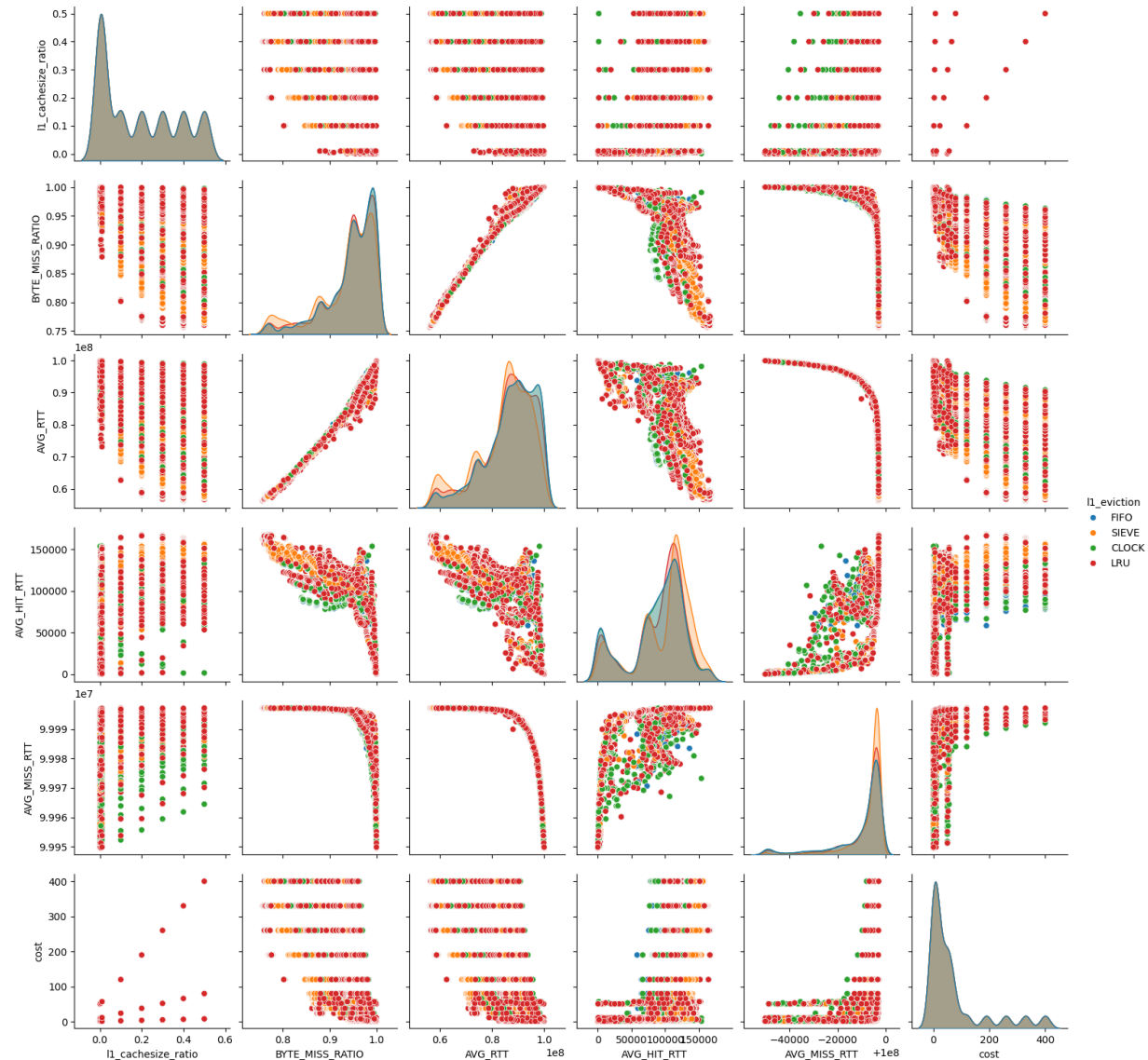
	CapEx
	OpEx
	ABC (Activity based cost)
	TDABC (Time Driven ABC)

© 2024 Solidigm. All rights reserved.

Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

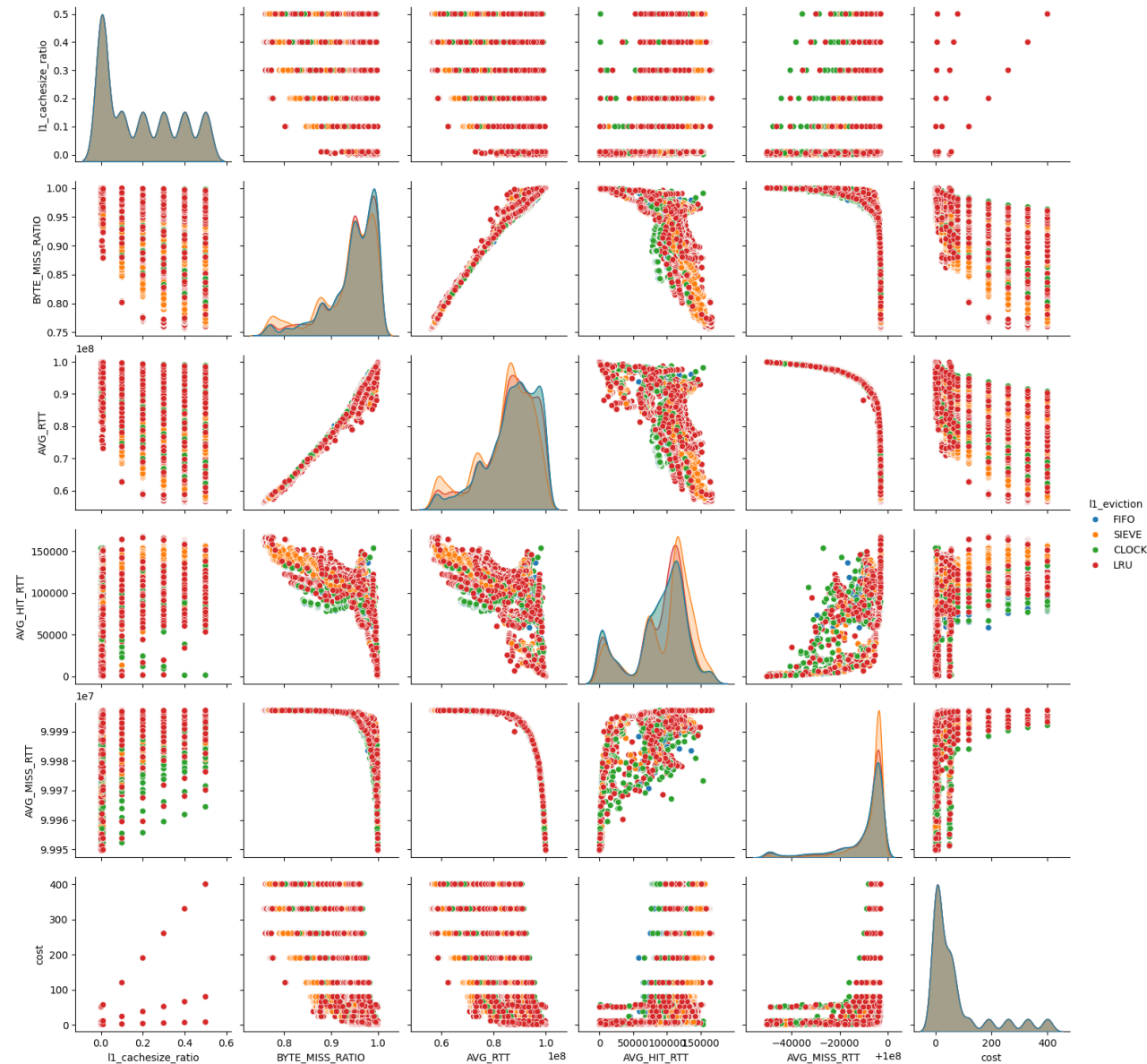
- Experimentation space should be defined and refined



Paying attention to the wrong thing:

# Design decisions are made based on too small a set of data

- Experimentation space should be defined and refined



Paying attention to the wrong thing:

## Design decisions are made based on too small a set of data

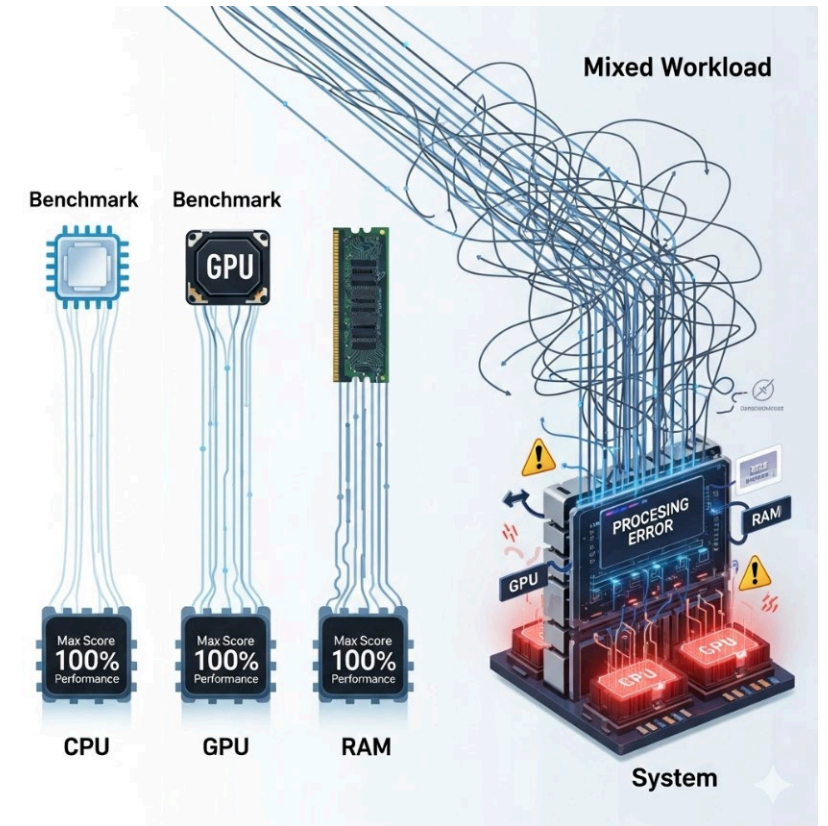
- This experimentation should be included in up-front costs
  - A normal part of due diligence
  - Adequate documentation to tell a vendor what you need rather than what you can buy



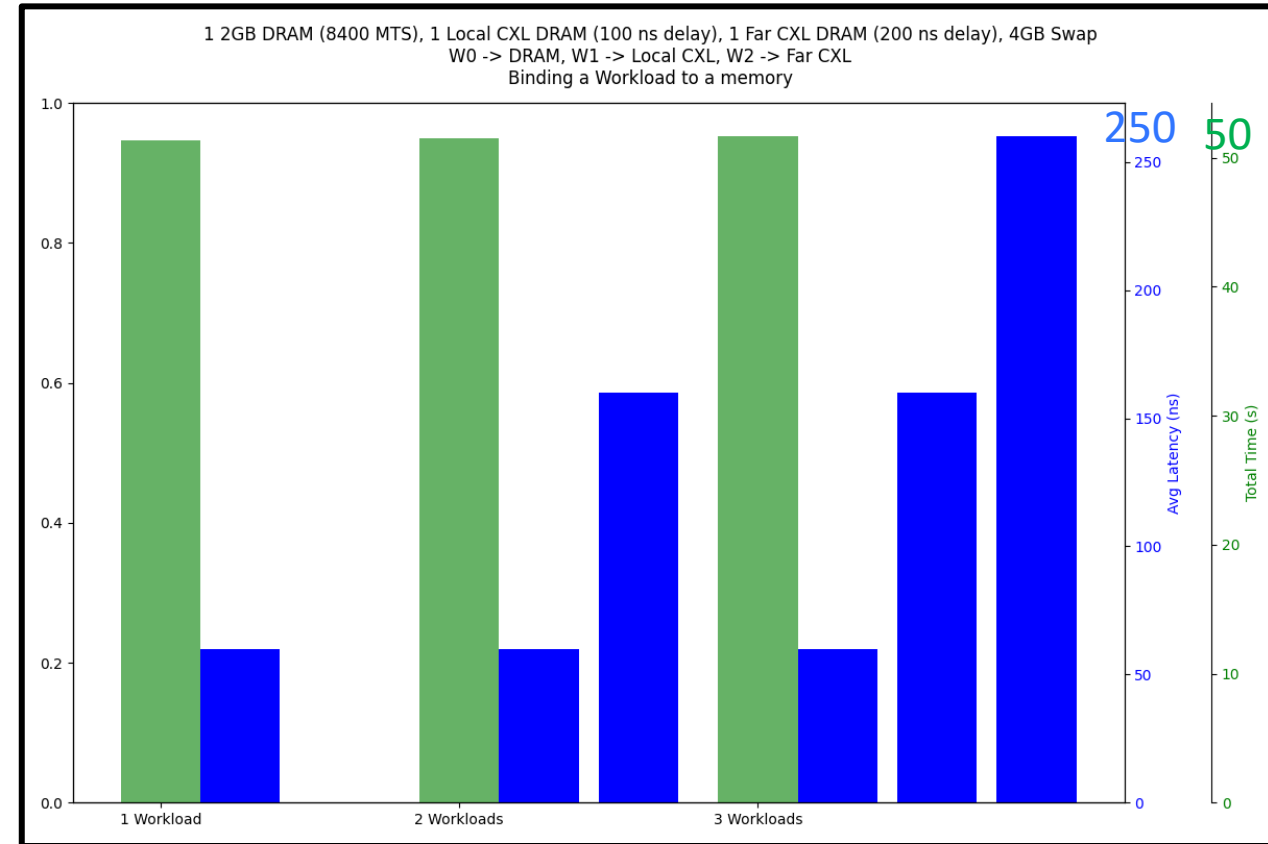
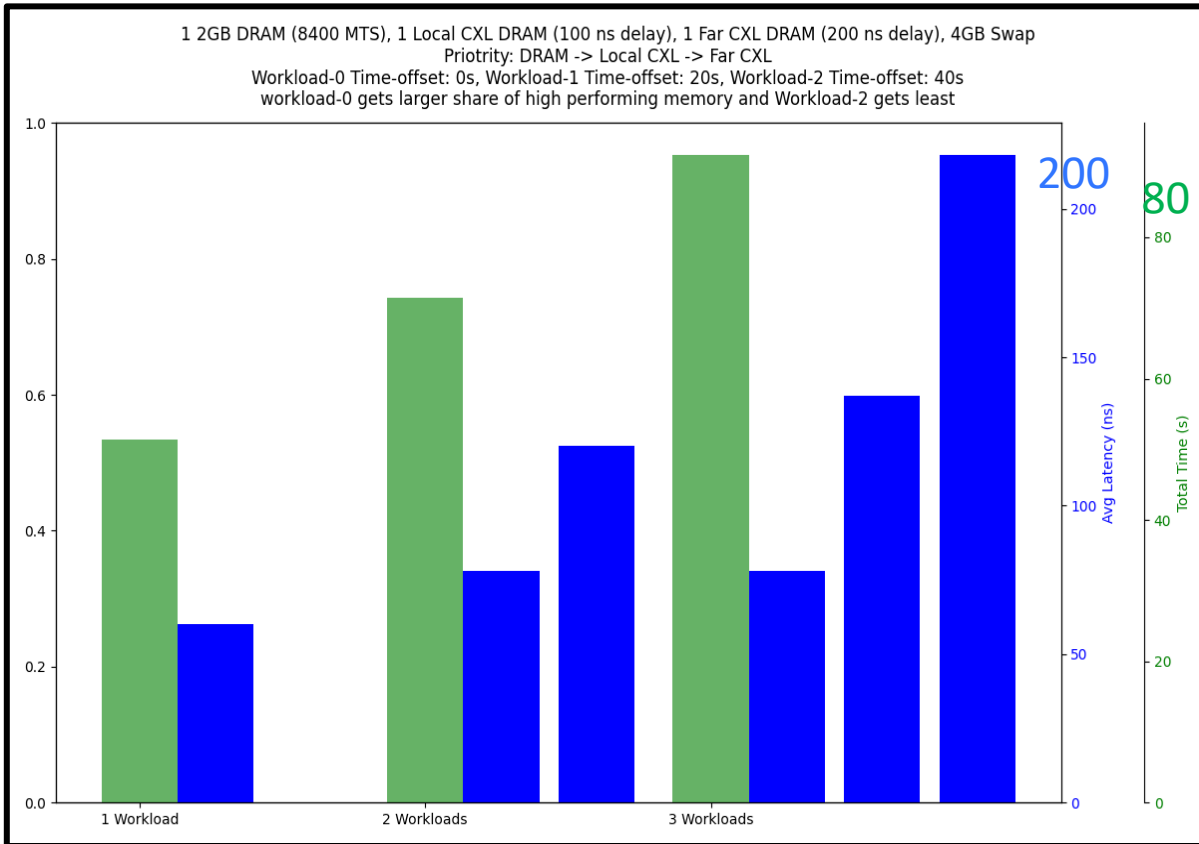
Paying attention to the wrong thing:

# Decisions made without taking applications into account

- Max performance of each component doesn't define the system performance
- Benchmarks don't adequately measure performance of specific applications
- Mixed workloads don't perform like the aggregate of the individual workloads



# Mixed, Unmanaged applications waste resources



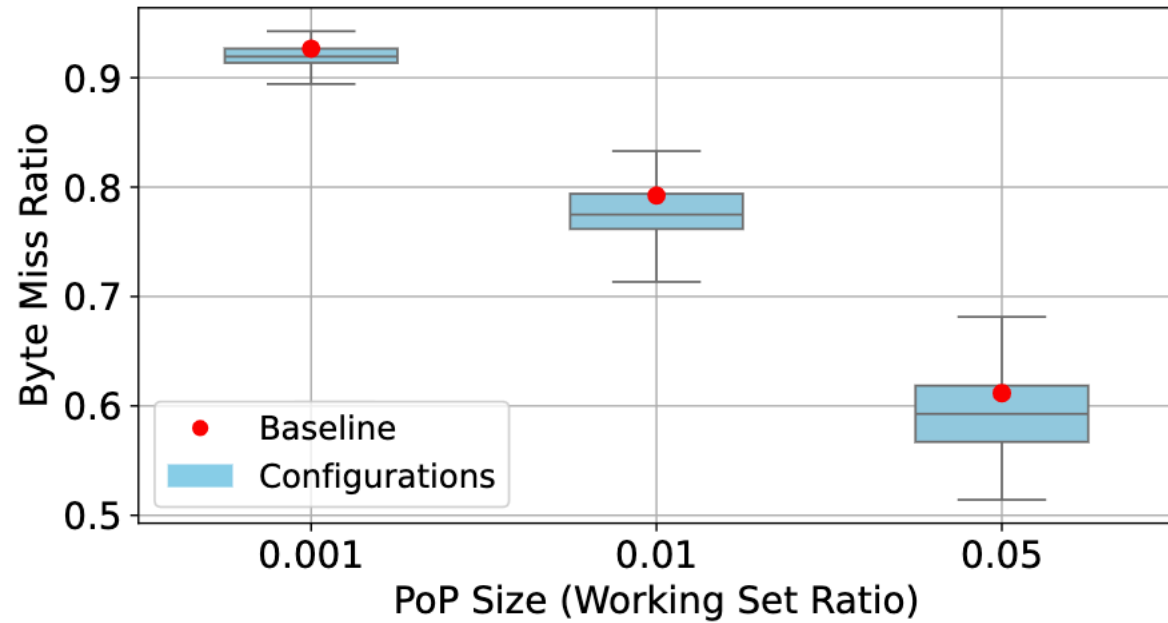
Paying attention to the wrong things:

# Decisions made without taking applications into account

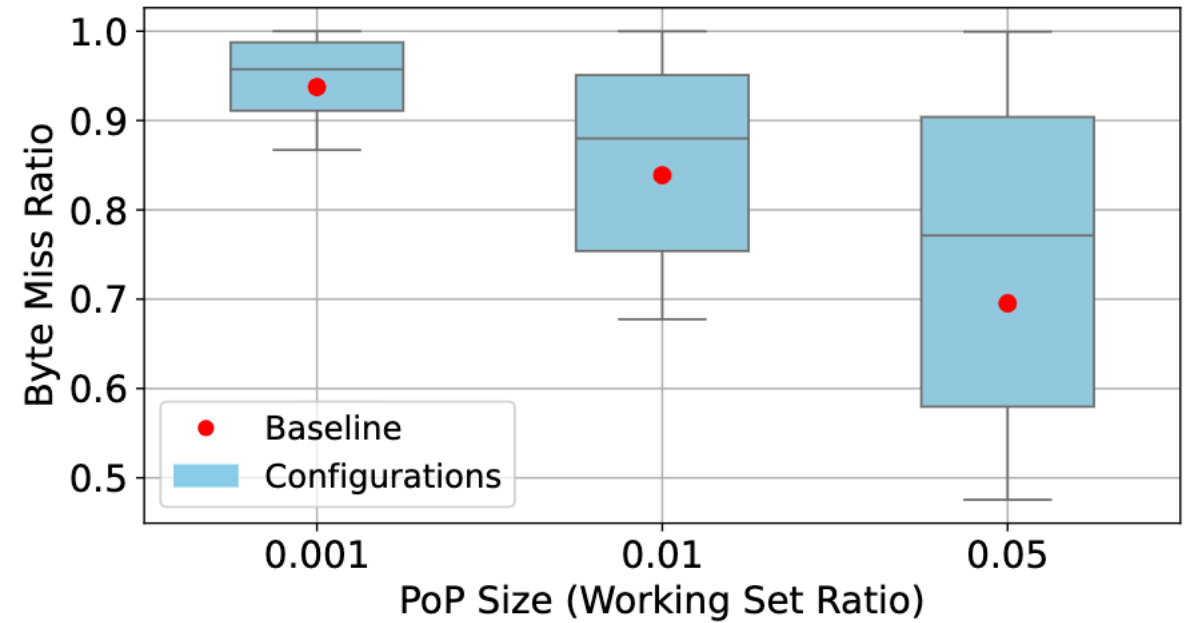
- Various workloads can be optimized multiple ways
  - Including different workloads with the same premise
- Optimization is multi-objective
  - Cost vs. performance is a simpler case
  - CapEx vs. TCO can monkey up the true cost
  - CapEx/OpEx arbitrage can be a factor



# Case Study Results

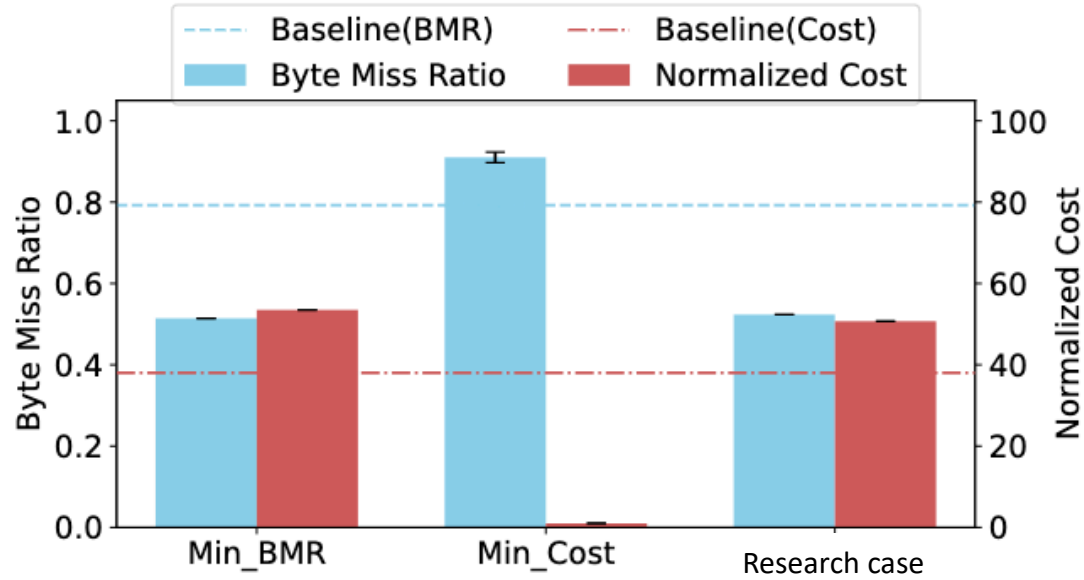


Wikimedia

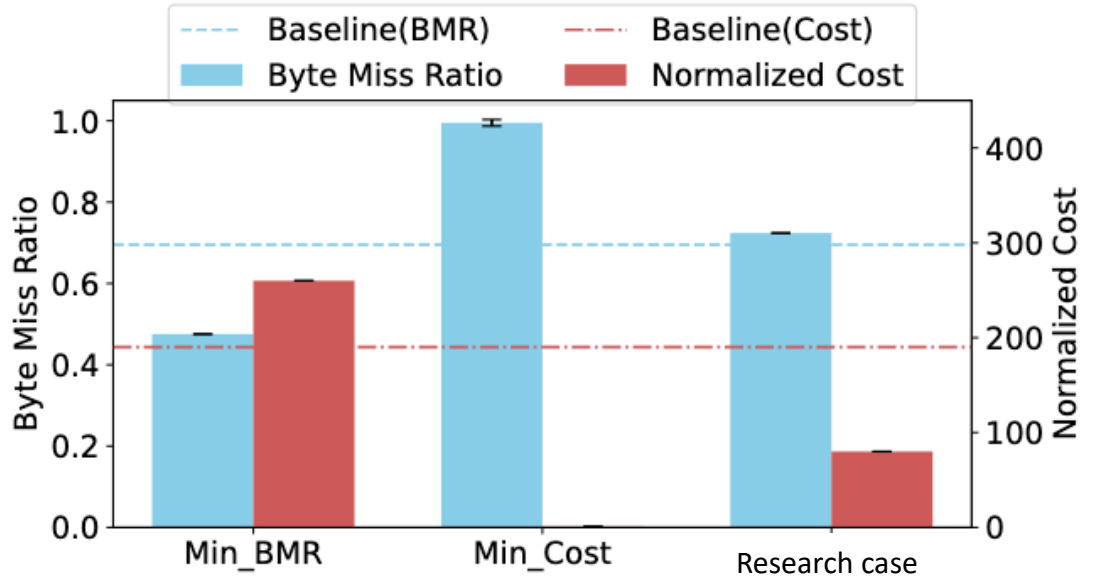


Cloudflare

# Case Study Results

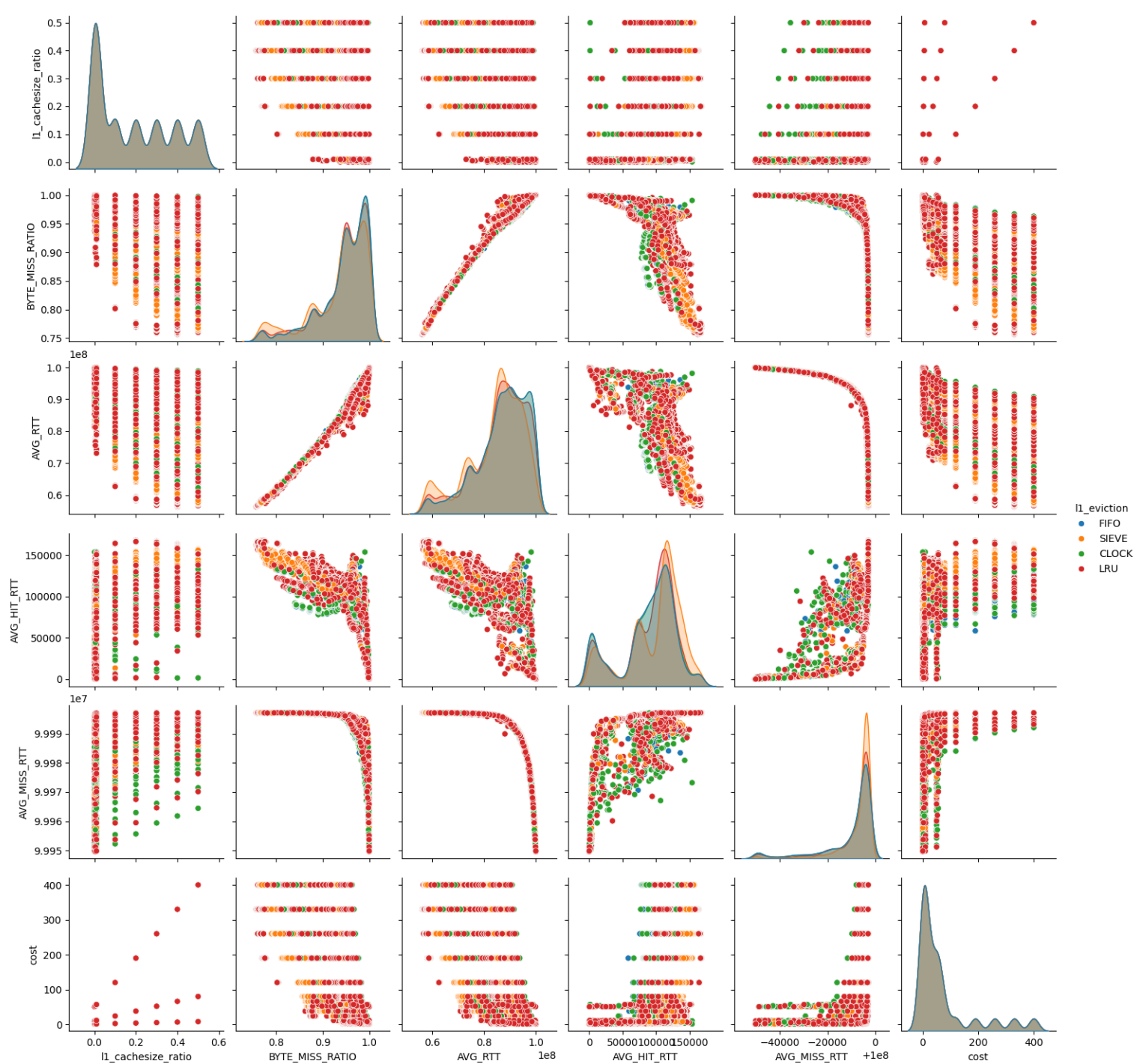


Wikimedia



Cloudflare

# Results



# I Learned Something Today

- Calculating costs help determine the scope of the entire price.
- Design starts before purchase, not after.
- Workloads always matter.
- Looking at my phone wasn't the most interesting thing I could have been doing.





# Thank you for attending!

Please remember to rate this session. You get access the presentations at  
<http://sniadeveloper.org/conference>