Avoiding and Breaking out of Capacity Prison

Jake Welch

jawelch@microsoft.com

@jaketwelch

Full PowerPoint with animations at https://aka.ms/H1yylp

Microsoft

Why should I care?

- Affects Customer experience
  - Latency
  - Availability
- Operational toil
- Business
  - Cost to purchase capacity
  - Inability to expand customer usage

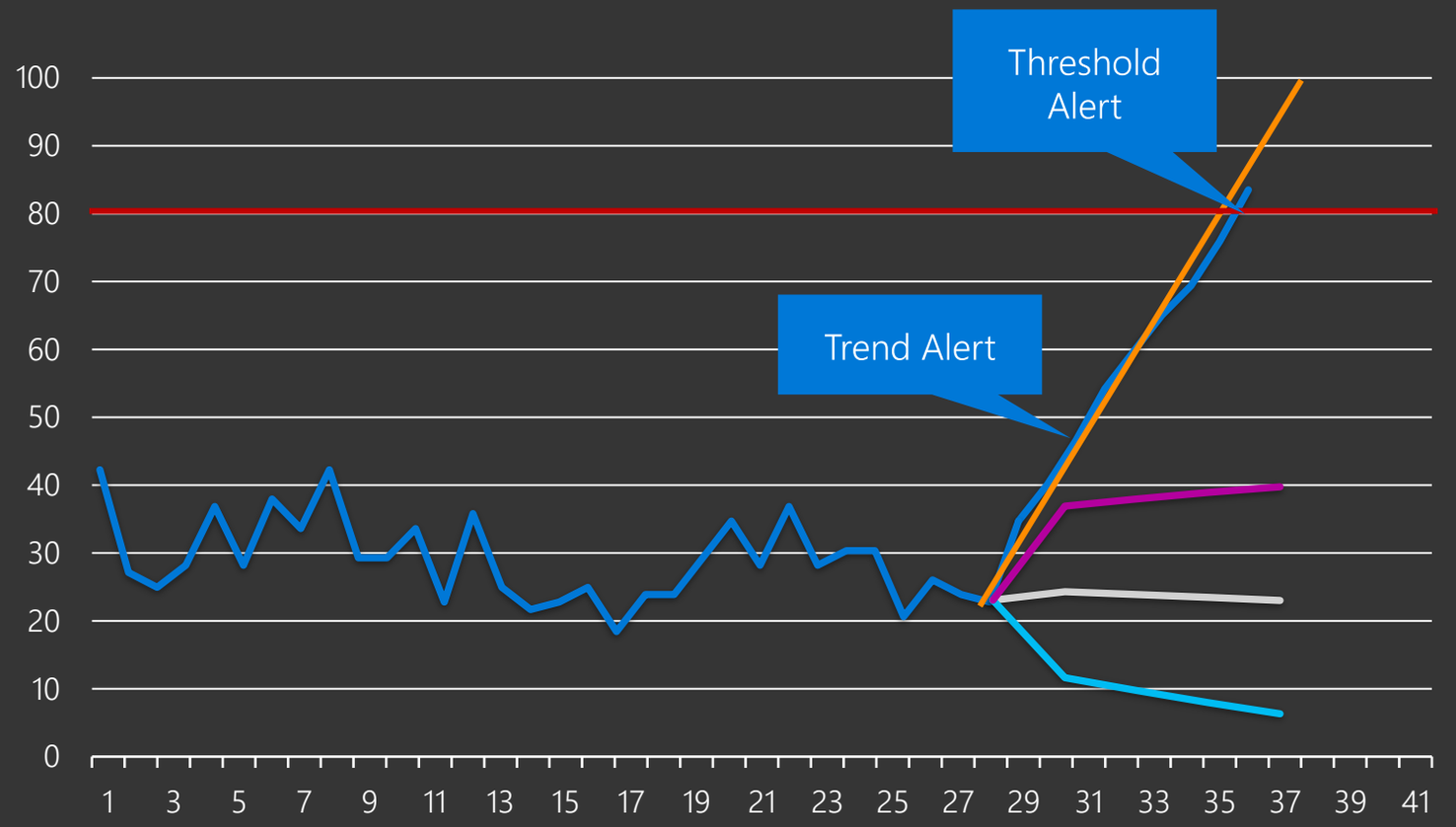
Identify

- Physical Resources
  - *CPU, memory, TCAM, disk, bandwidth, cooling system, …*
- Logical Resources
  - *Threads, file descriptors, application limits, ports, …*
- ‘Hard’ & ‘Soft’ limits
  - SKUs have varying capabilities
  - Don’t expect linear scaling!
  - May change over time
  - Limits are often inter-related

Monitor

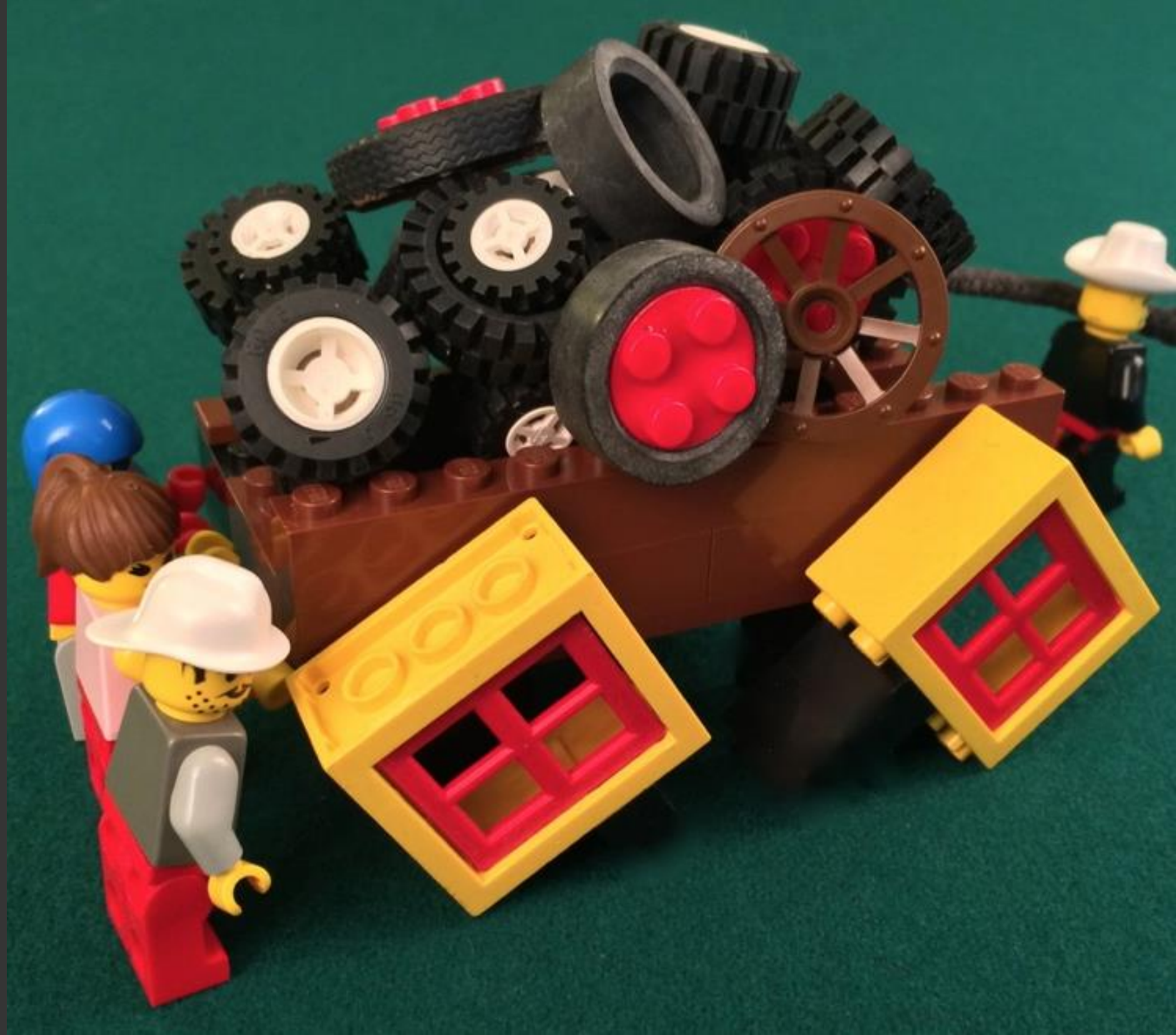
If you're not monitoring capacity, you're not monitoring your service.

- Monitor usage & efficiency
- Time Series for each limit
- Threshold Alerts
- Trend Alerts



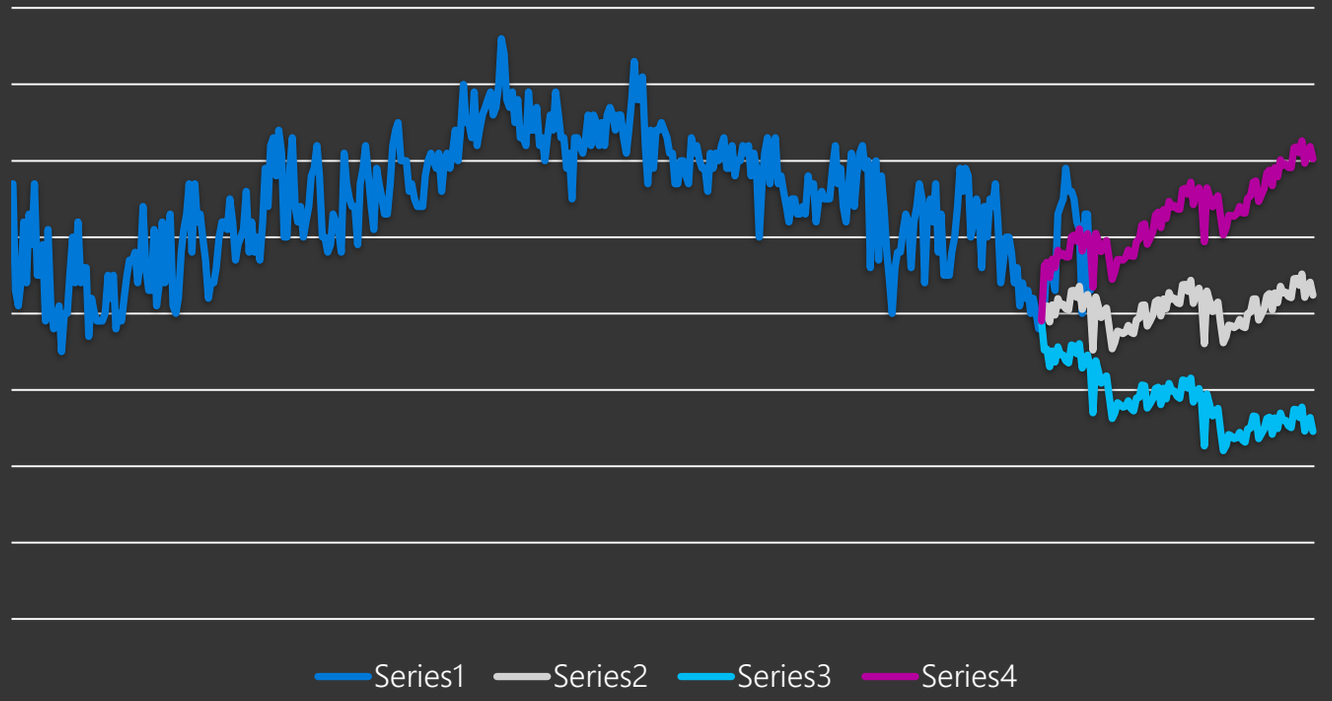
Efficiency

- Look for…
  - Deviations over time
  - Change in trends
- Set target ranges
- Compare versions
- Not linear
- Examples:
  - TPS per instance
  - Cost per X transactions
  - Data compression ratio



Forecast

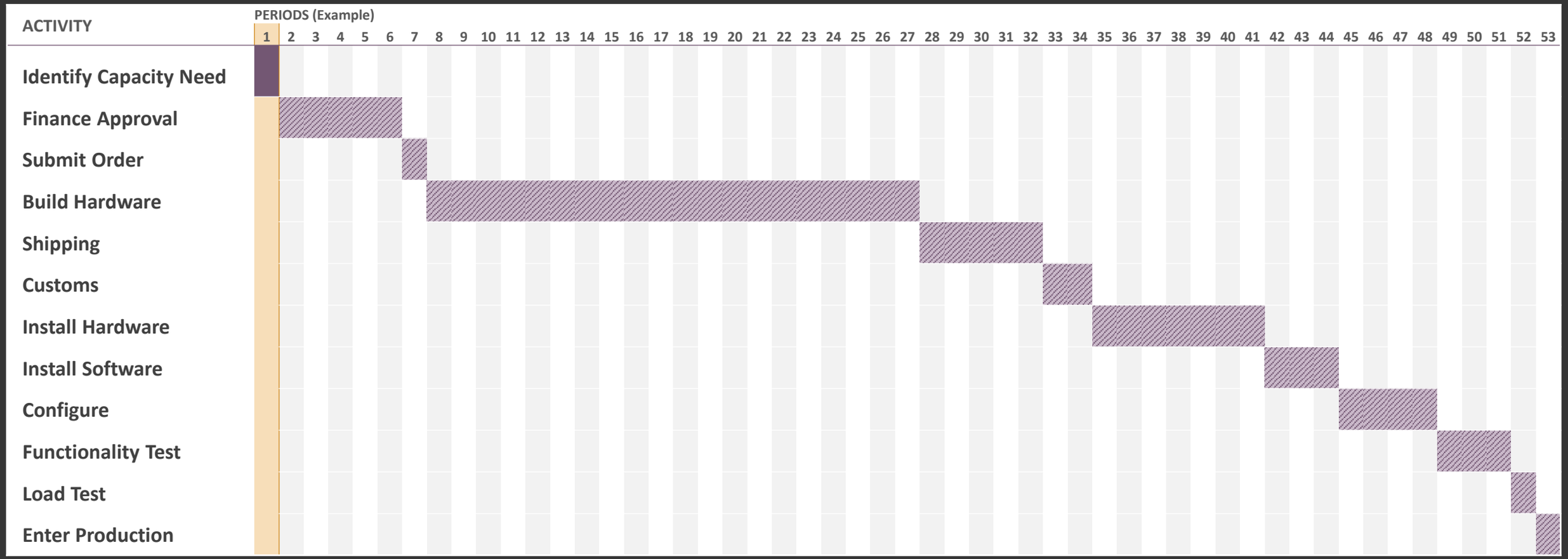
- Forecast key limits
  - Short, medium, long term
  - Consider seasonality
  - Consider promotions, new features
  - Organic & inorganic growth
- Normalize your SKUs
- Validate regularly w/ actual
- Plan ahead:
  - Hardware end of life
  - Time to get and setup new servers?
  - Time to get new datacenter space?



Normalize:

SKU	CPU	Transactions/sec	Disks	IOPS / GB
V1	4 x CPU Type D	123	1TB vendor 1	1
V2	4 x CPU Type E	217	8TB vendor 1	2
V3	8 x CPU Type F	643	8TB vendor 2	3

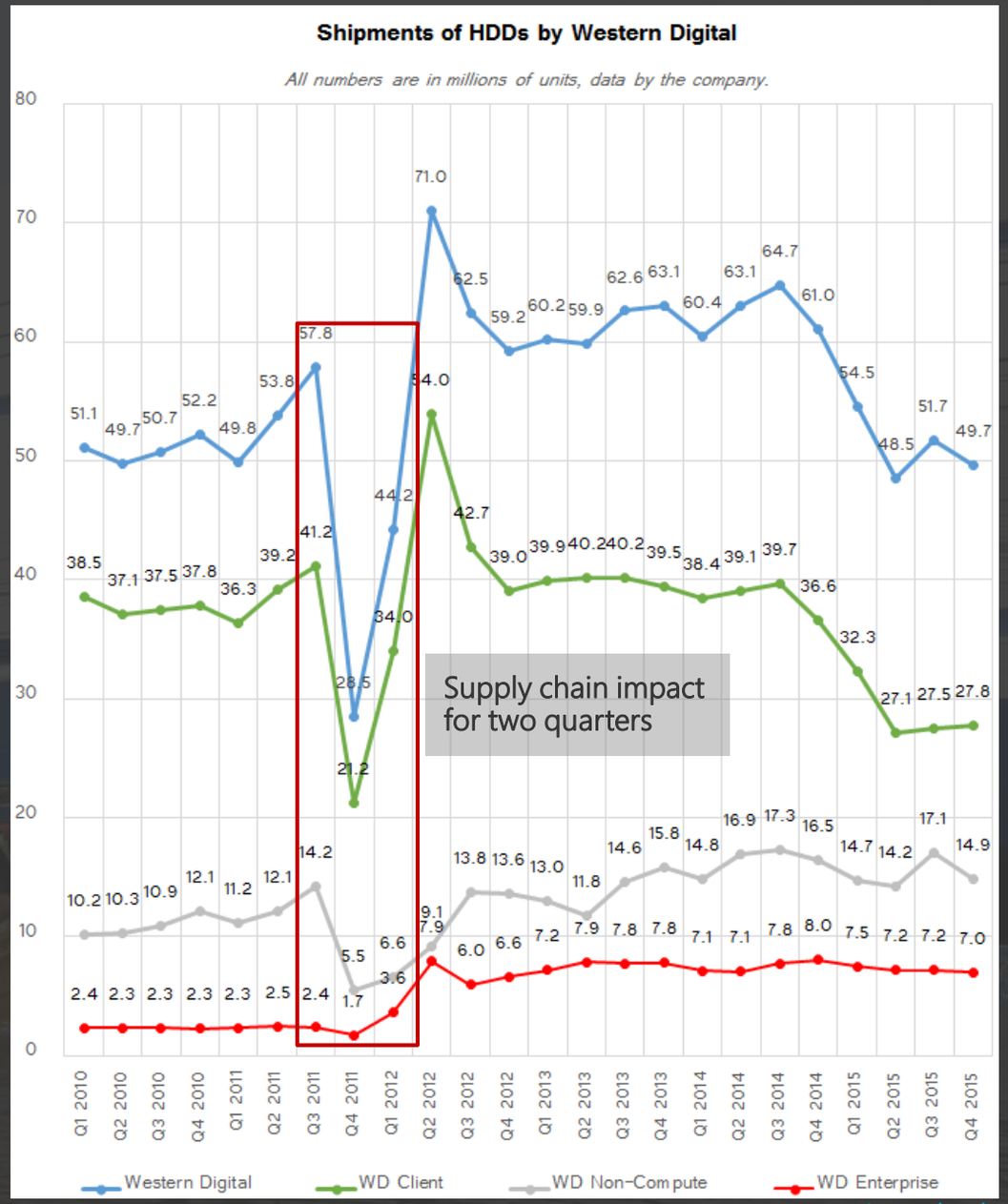
Getting Hardware



Prepare

What if…

- You can’t get new hardware for six months?
- Hardware fails rapidly?
- Higher than expected usage?
- You run out?



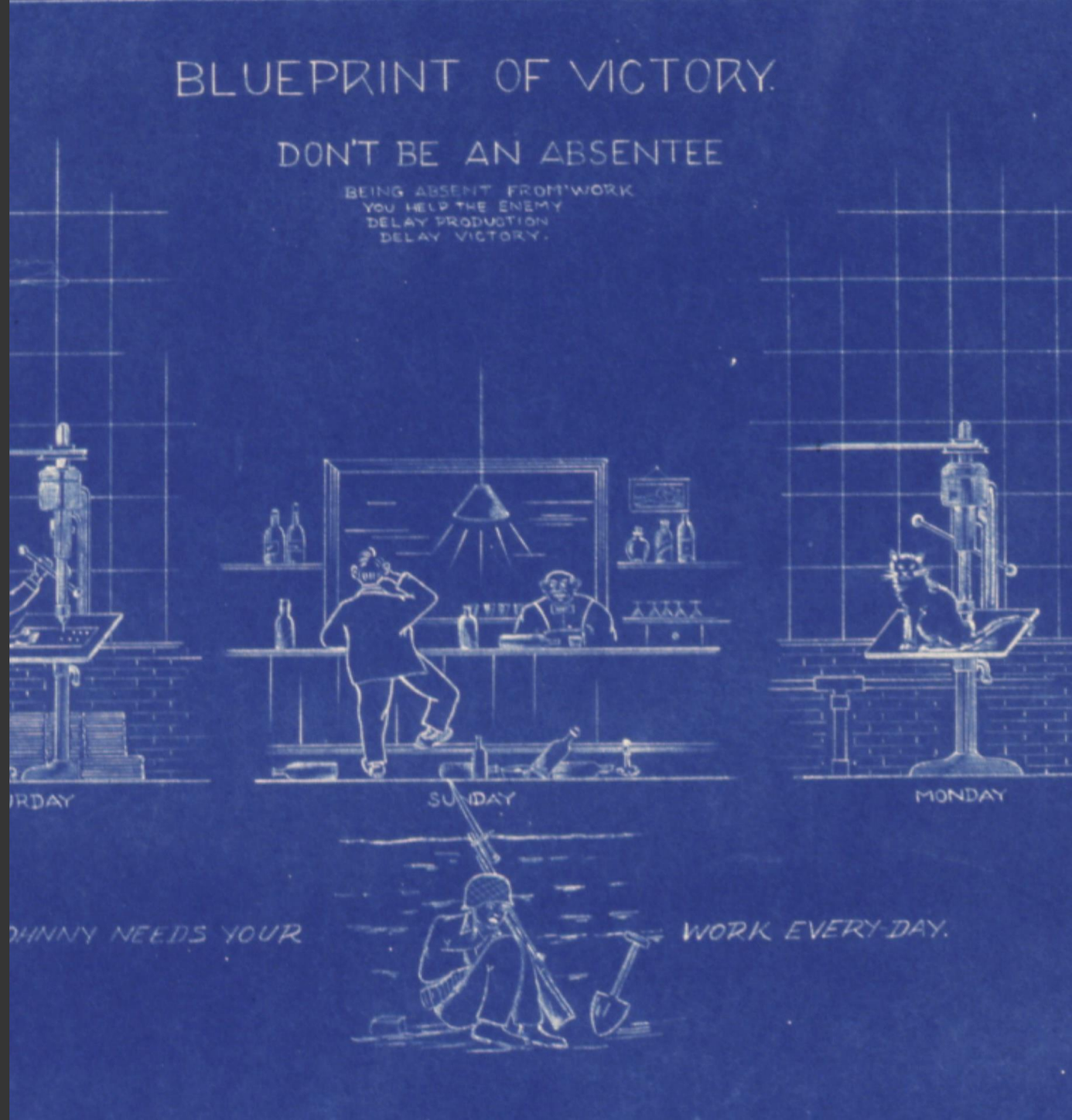
[Anandtech](#)

Image: New York Times

Levers

- Cost / benefit decision
- Implement as playbooks
- Proactively identify, test
- ‘business as usual’, not ‘exceptional’

- Examples
  - Shed or shift load to other regions
  - Decrease redundancy
  - Use safety buffer
  - Sales or marketing incentives, advertising
  - Investigate fraudulent usage
  - Repair failed hardware
  - Decrease retention policy
  - Use decommissioned capacity
  - Compress data, dedupe, erasure code

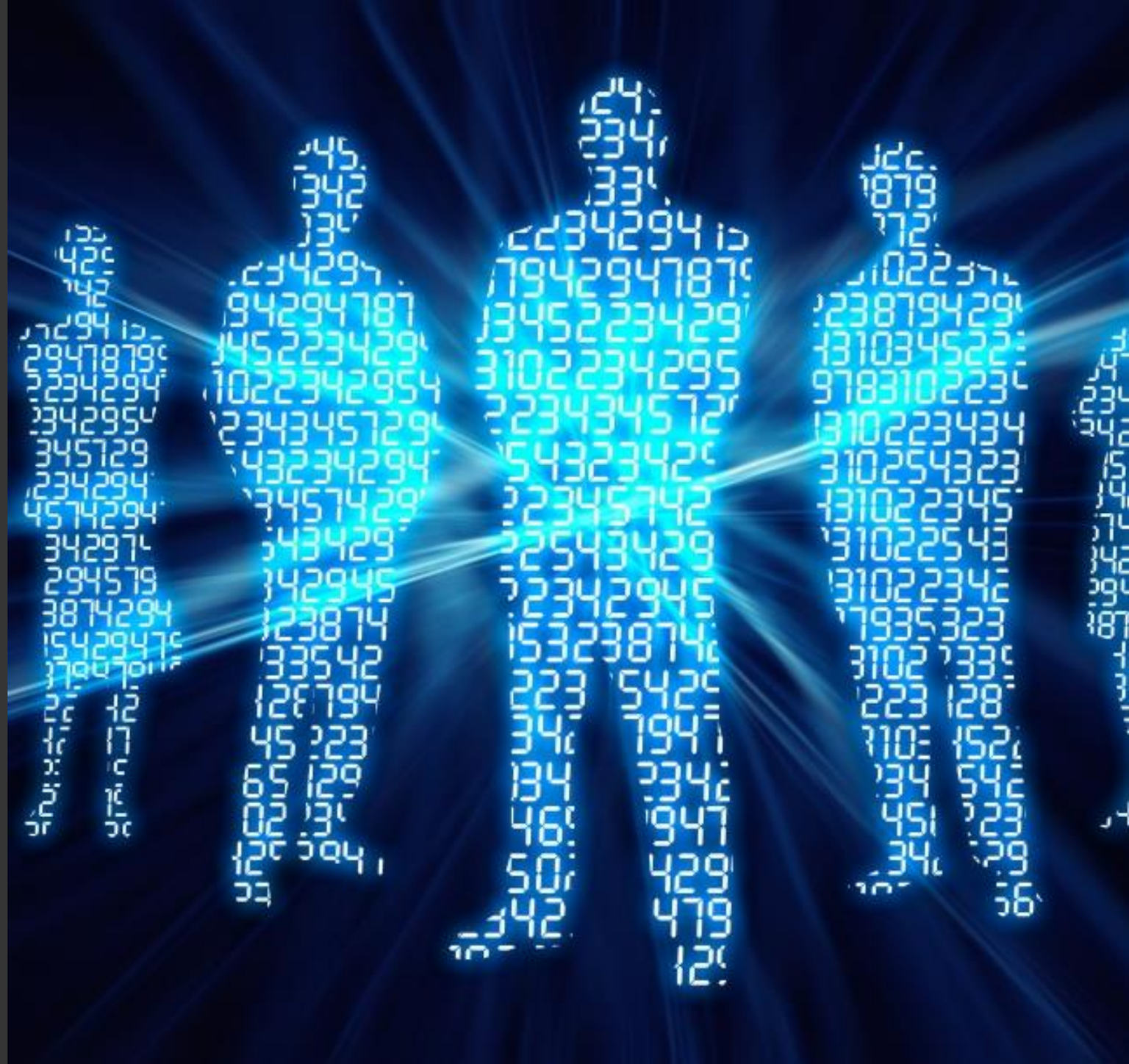


Playbook

Issue	Couch out of capacity
Cause	Incorrect orientation
Validation	Observe couch orientation
Lever Name	Correct couch orientation
Risk	Medium, risk of physical injury
Cost	Loss of floor space, physical exertion
Steps	1. Measure couch, floor space, wall clearance 2. Use PlacementEstimator script from repo ‘couch’ to verify space and determine optimal placement & tipping velocity 3. Gather the team (at least 3 people) 4. Move couch to optimal position 5. Slowly tip couch to correct orientation

Virtual Team

- Product Engineering
  - Efficiency
  - SKU testing
  - Feature release plans
- Hardware Engineering
  - Hardware design, capabilities
- Capacity Planning
  - Forecasting (data scientists)
  - Data center space allocation
  - Work with marketing on incentives
  - Track non-organic growth asks
- SRE
  - Playbooks/levers
  - Response + automation
  - Monitoring
- Finance
  - Keep us honest



Building Levers

Problem

High toil, difficult to scale

Goal

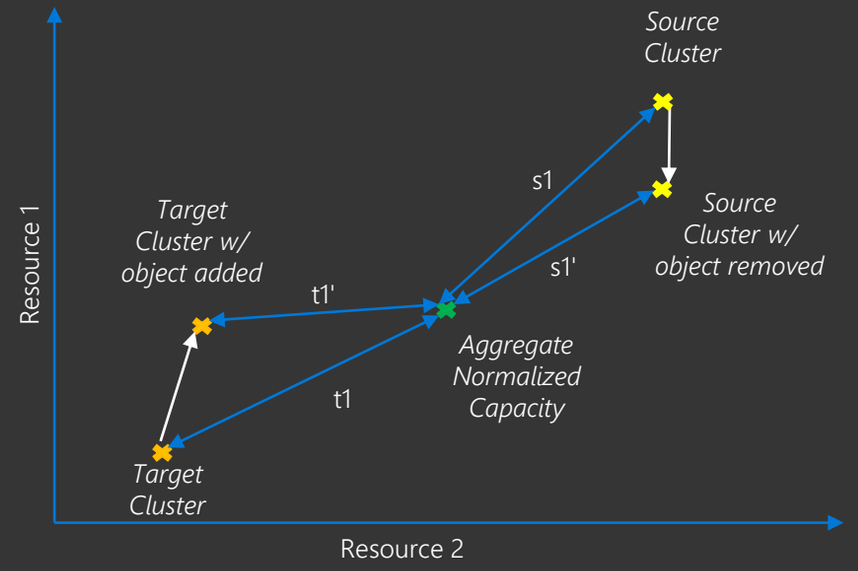
Reduce toil and improve scalability

Solution

Automated migration of capacity across clusters

Vector Bin Packing 101

- Efficient, predictable placement of many objects
- Works well with many dimensions
- Overview:
  - For each object and source/destination, calculate dot product before/after
  - Sort by pairs that bring distance closest to zero
    - Attempt to place largest ‘benefit’ first
    - Validate within target, soft and hard constraints
    - Try until all objects are placed or clusters within target

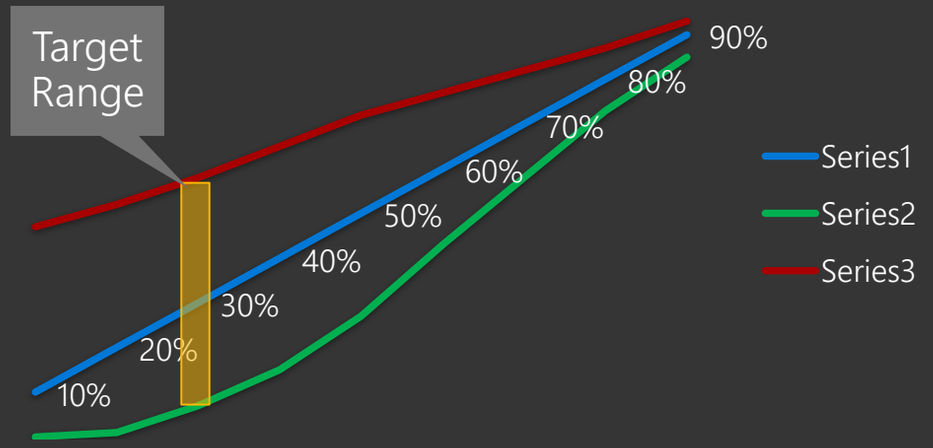


Automation Steps

1. Aggregate normalized capacity

Cluster	SKU	TPS.Current	TPS.Limit	Percent
A	V1	4,900	5,000	98%
B	V2	12,000	15,000	80%
C	V2	1,000	15,000	7%
D	V3	2,000	50,000	4%
Aggregate		19,900	85,000	23%

2. Set min, max target ranges



Source clusters are above max

3. Identify candidates to move

Candidate	TPS	Cluster Move	Percent of Source	Percent of Destination
1	4,000	A → C	80%	27%
1	4,000	A → D	80%	8%
2	100	A → C	2%	0.7%
2	100	A → D	2%	0.2%
3	8,000	B → C	53%	53%
3	8,000	B → D	53%	16%

4. Allocate w/ Vector Bin Packing

Candidate	TPS	Cluster	Source %	Dest %
1	4,000	A → C	98% -> 18%	7% -> 34%
3	8,000	B → D	80% -> 27%	4% -> 20%
2	100	A	No move, within limits	

*Assumptions

- Candidates have varied, positive resource usage
- Not all candidates must be placed

Constraints

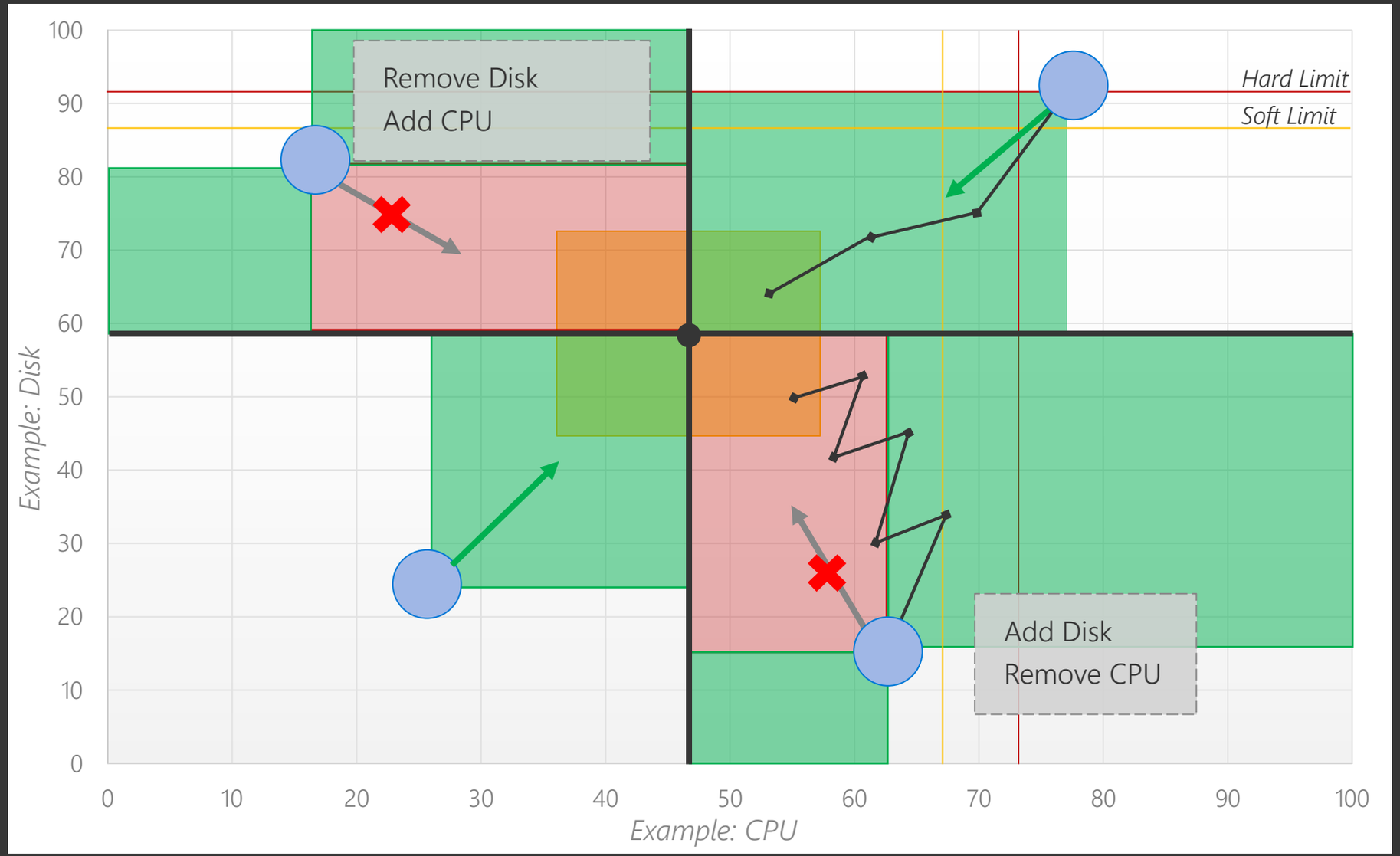
- Protects the system from harming itself
- Allows enforcing hard limits
- Can dynamically adjust criteria

- Examples:
  - Never schedule more than 20 outbound migrations at a time
  - Never migrate candidates with more than xPB of data
  - Never exceed 500gbps of ingress for a cluster
  - Never migrate if CPU is more than X%

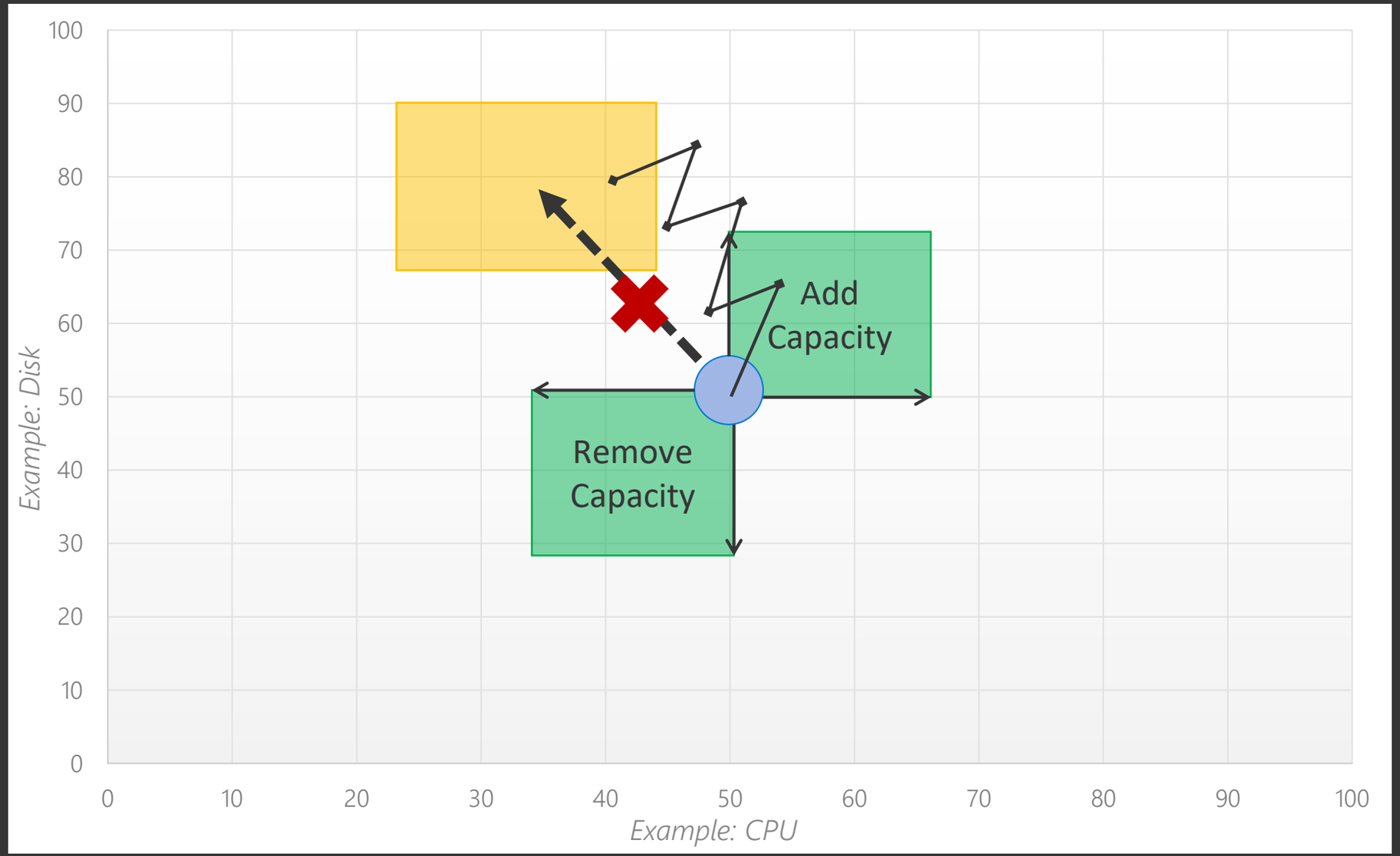
Challenges

- Deadlock if resources are imbalanced
- Migrations are not instantaneous
- Uses resources: Need to balance cost and benefit

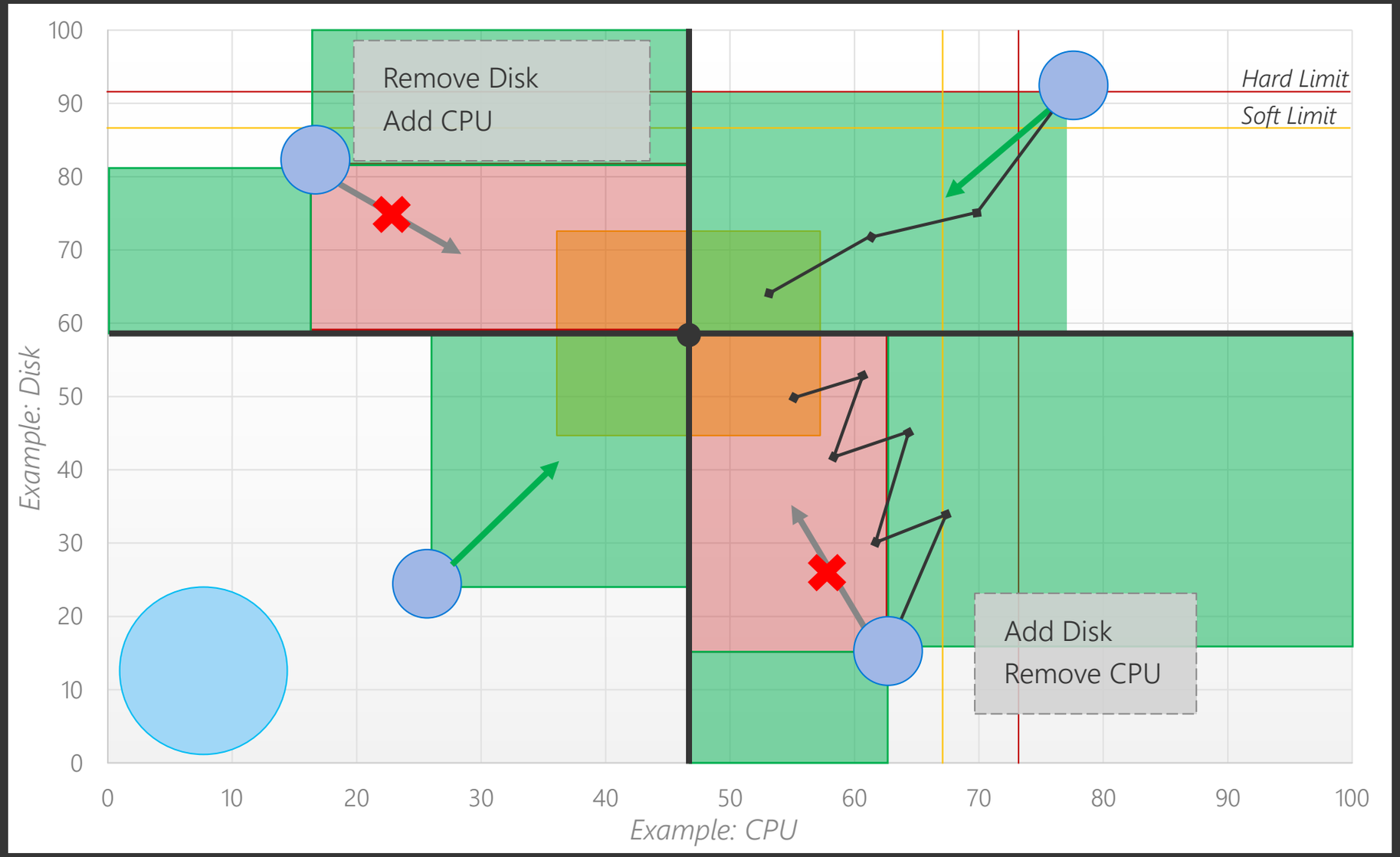
2-Dimensional Balancing



Legal Movement



2-Dimensional Balancing



Result

Goal: Reduce toil and improve scalability

- Improved customer experience
  - Latency more consistent
  - Availability & reliability improved
- Reduced capacity imbalance
- More ‘runway’ to respond to issues
- Toil now near zero
  - More time to solve other problems
- Scales w/ growth

Capacity Management

- ✓ Identify physical and logical limits
- ✓ Monitor and alert on resource growth & efficiency
- ✓ Forecast resource usage
- ✓ Prepare for the unexpected
- ✓ Create and test playbooks of levers
- ✓ Automate common tasks

Questions?

jawelch@microsoft.com

@jaketwelch

Microsoft