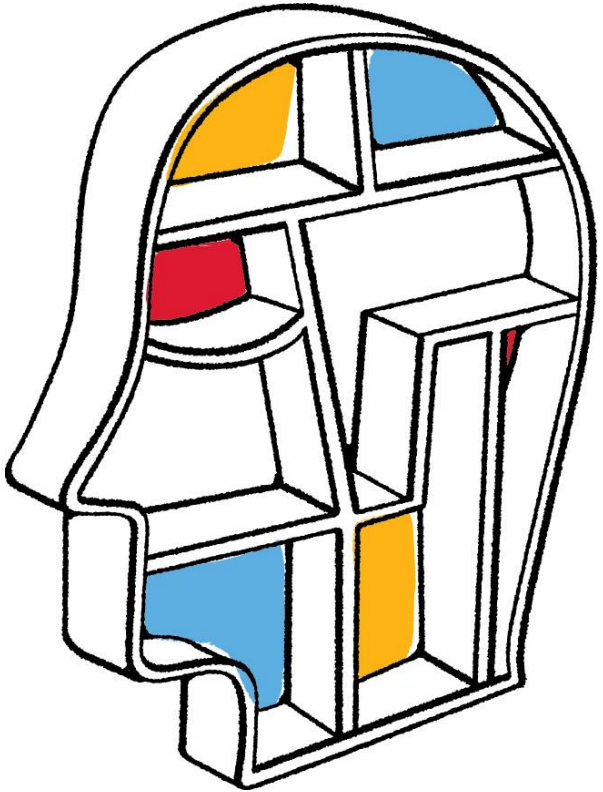
NetApp

Go further, faster®

Anode

Empirical Detection of Performance Problems in Storage Systems



Vipul Mathur, Cijo George, Jayanta Basak
Advanced Technology Group, NetApp India

Motivation

- Hard to detect, diagnose and fix performance problems in computing systems
  - Storage systems are no different
  - Addressing component failure is easy in comparison!

- Affects user satisfaction
  - Data unavailability/ downtime
  - Typical complaint: “system is slow”
  - Performance issues take 10x longer to close than others

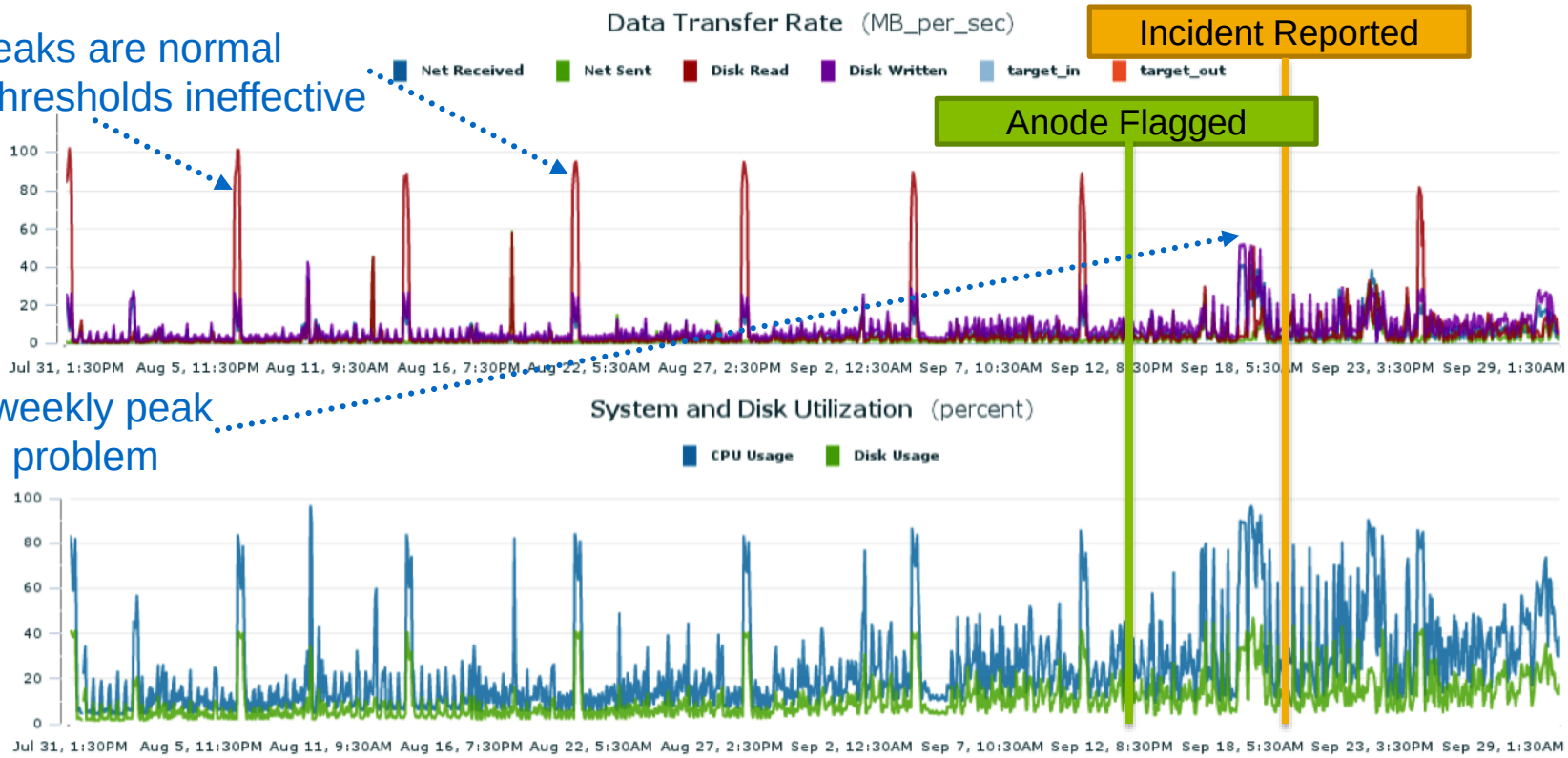
Challenges

- Is there a problem?
  - Thousands of metrics to gather and analyze
  - Systems and workloads are unique: no universal thresholds
- Where is the problem?
  - Larger the system, harder it becomes to pinpoint affected parts
- Exactly when does the problem manifest?
  - Multiple workloads and differing activity cycles
  - Performance problems can be intermittent

Sample Metrics from Actual Incident

Weekly peaks are normal
➔ static thresholds ineffective

Missing weekly peak indicates problem

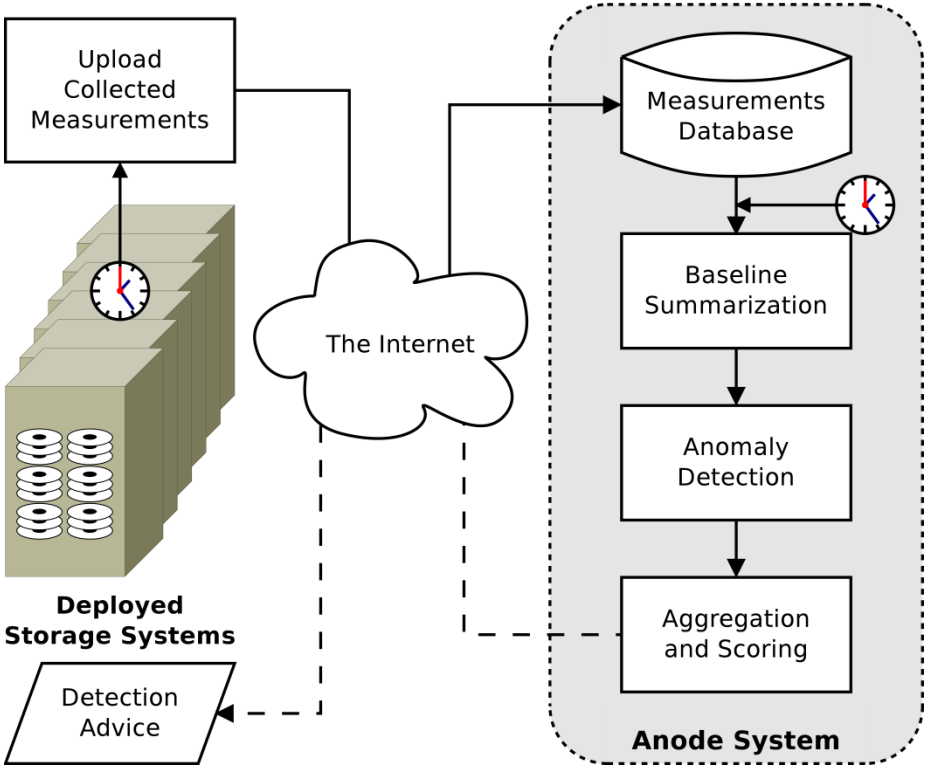


Anode Approach

- Improve productivity and effectiveness of experts
- Do not try to replace them!

- Use time-series analysis to process metrics
- Detect anomalies based on past behavior
- Pin-point affected parts
- Identify time-periods when impact is felt
- Find top symptoms experienced

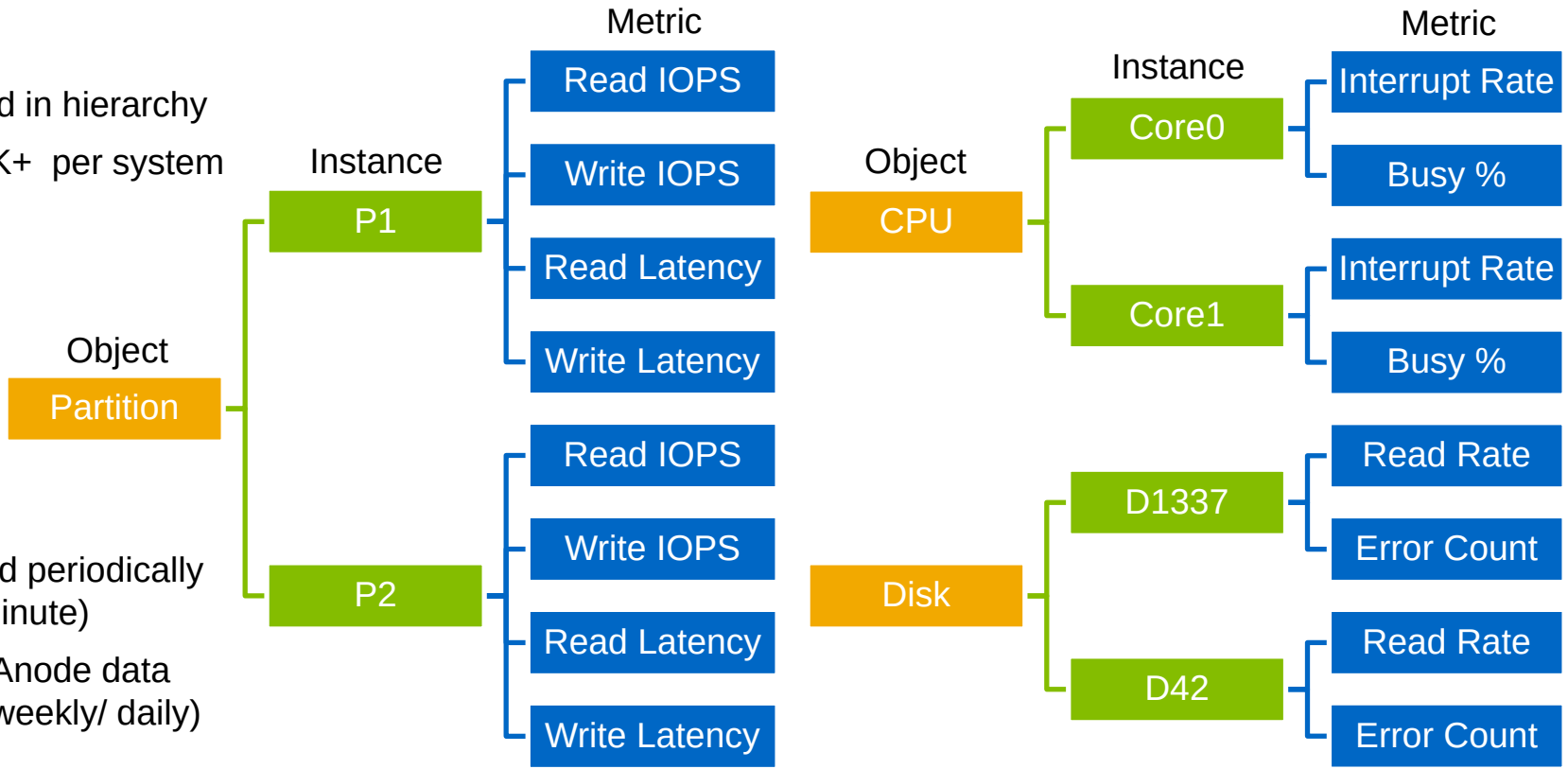
Anode System



- Metrics collected internally by storage systems deployed in field data centers
- Measurement data gathered in Anode data center
- Analyzed in batch mode
- Results made available to admins/ support personnel

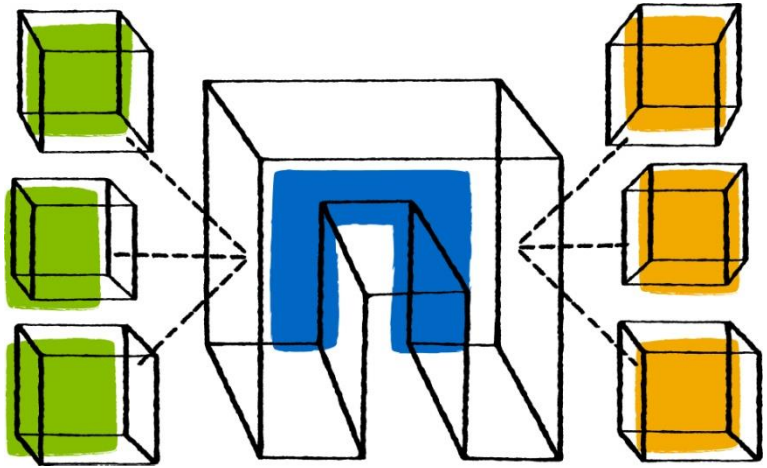
Metrics from a Storage System

- Arranged in hierarchy
- 3K—50K+ per system
- Collected periodically (hour/ minute)
- Sent to Anode data center (weekly/ daily)



Anode Methodology

Our solution



Overview

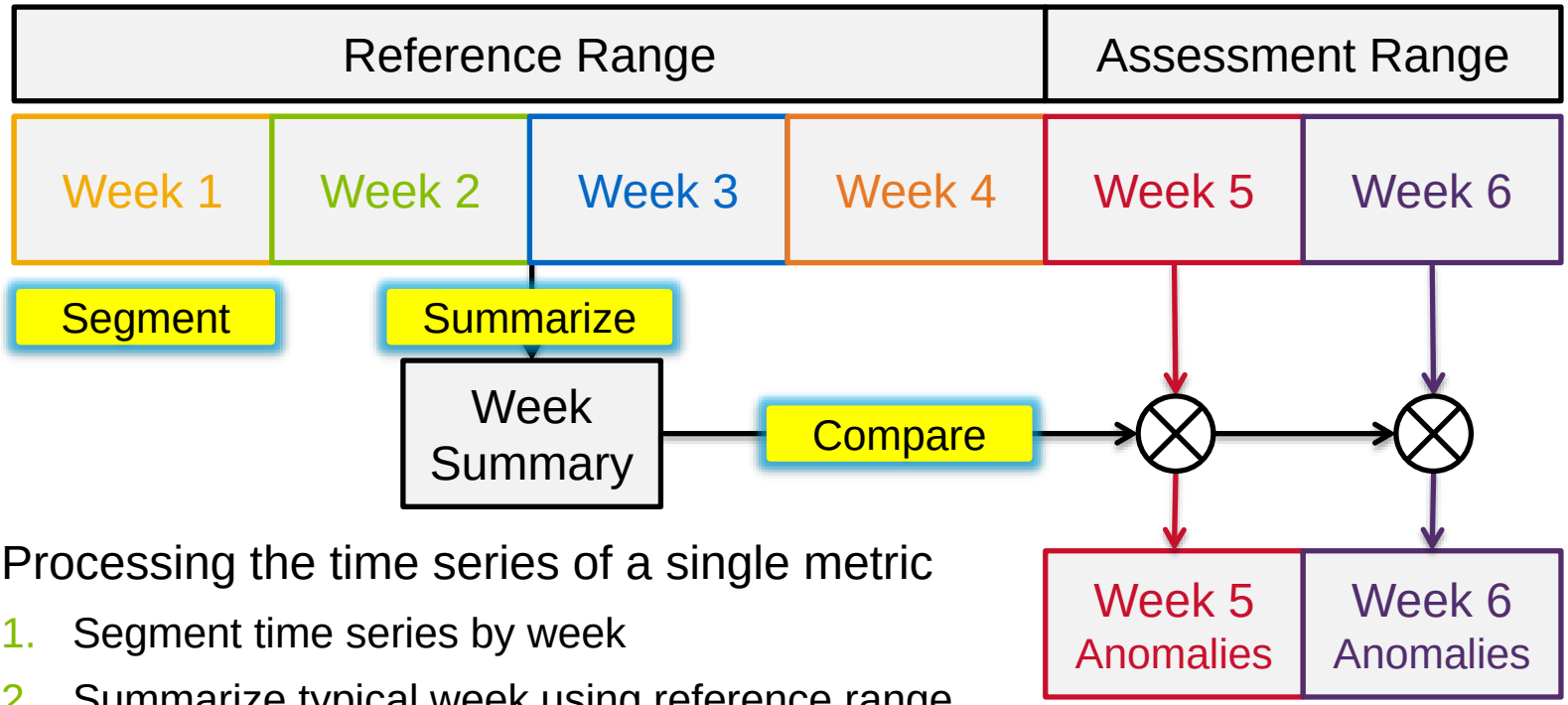
- Key Observation: Metrics repeat with weekly periodicity
  - Driven by commonly observed daily user load fluctuations

1. Baseline Summarization: Extract range of expected values for each hour of the week based on historical values

2. Anomaly Detection: Use the baseline summary to detect anomalies in individual metrics

3. Aggregation and Scoring: Use combinations of several metrics to make a robust assessment of performance

Anomaly Detection Overview

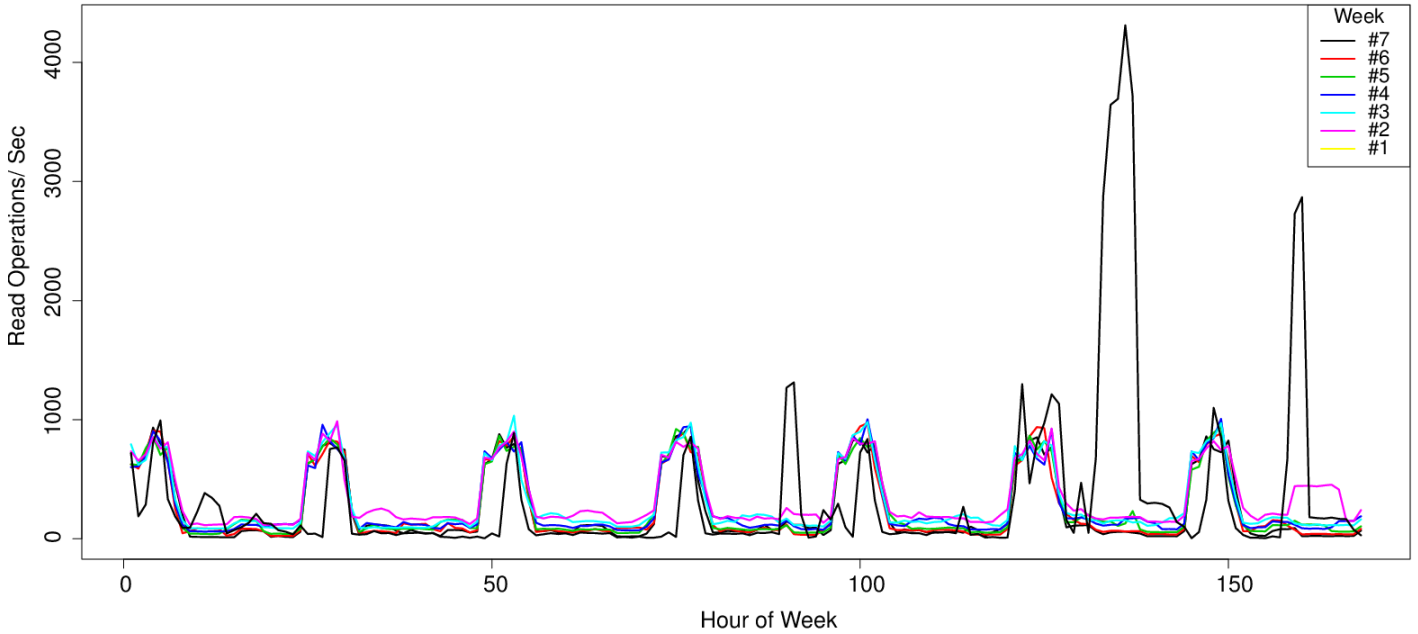


Processing the time series of a single metric

1. Segment time series by week
2. Summarize typical week using reference range
3. Compare with assessment range to flag anomalies

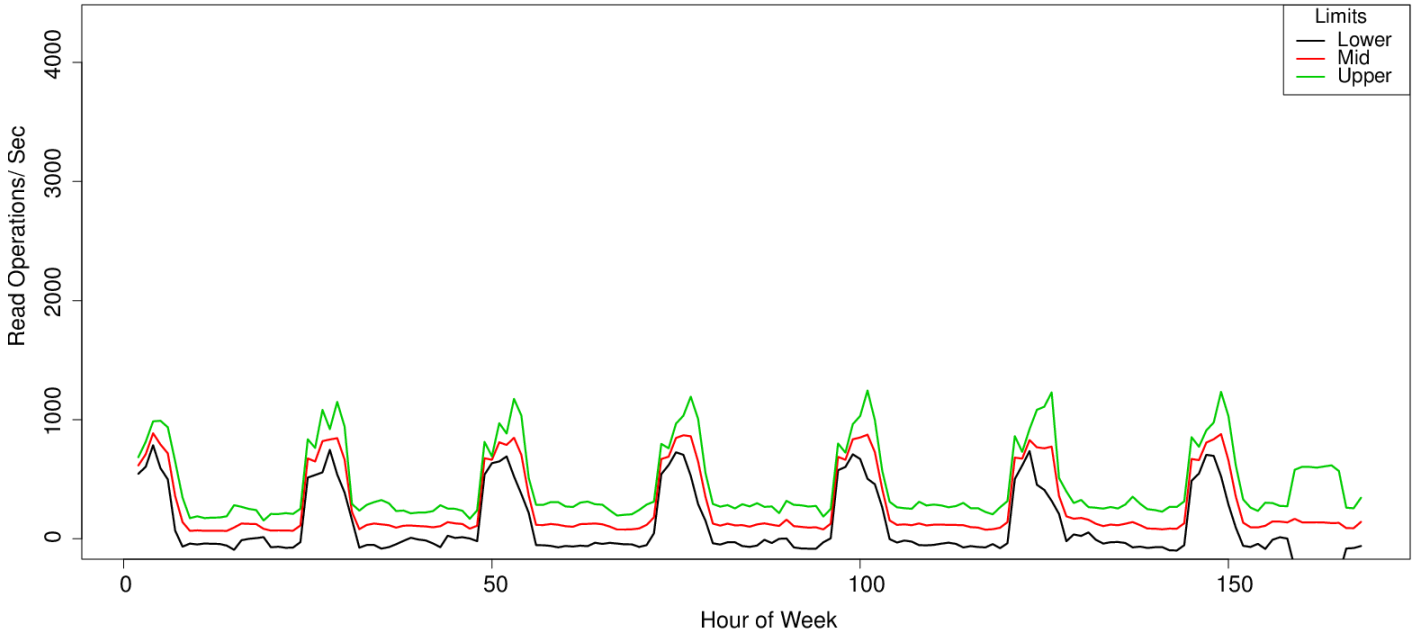
Anomaly Detection Sample

Weeks #1 to #7 segmented and stacked



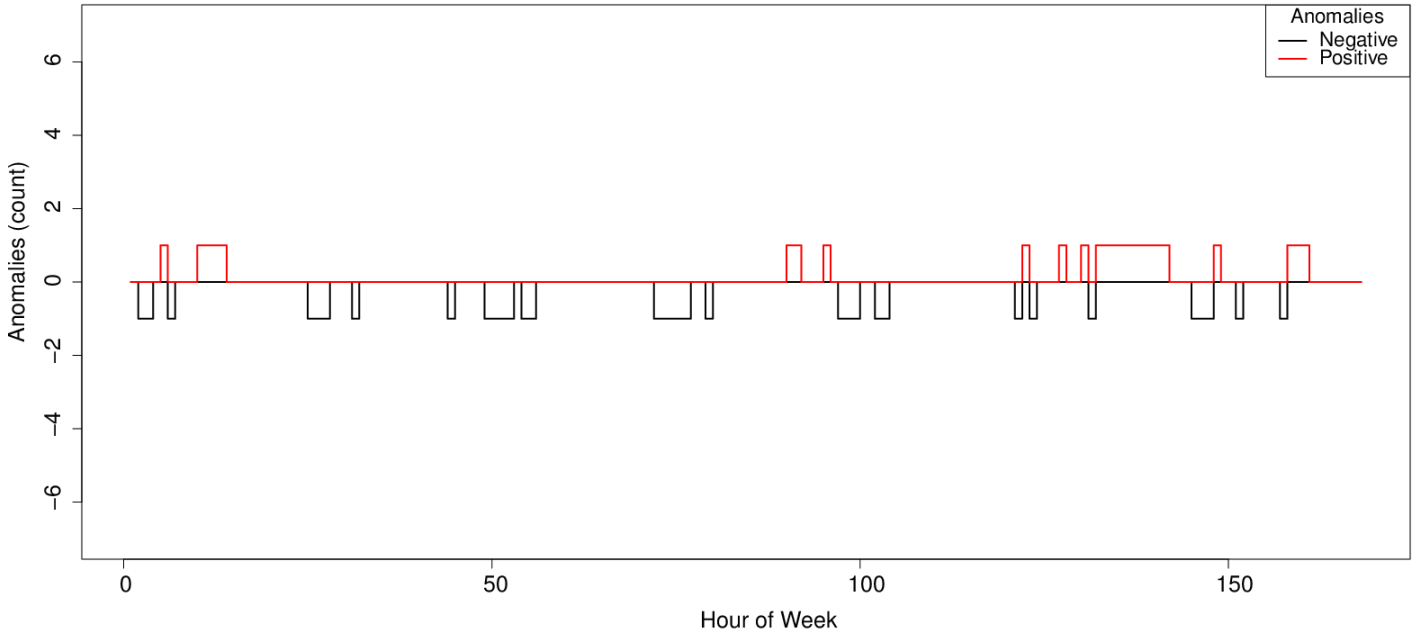
Anomaly Detection Sample

Week summary based on weeks #1 to #4



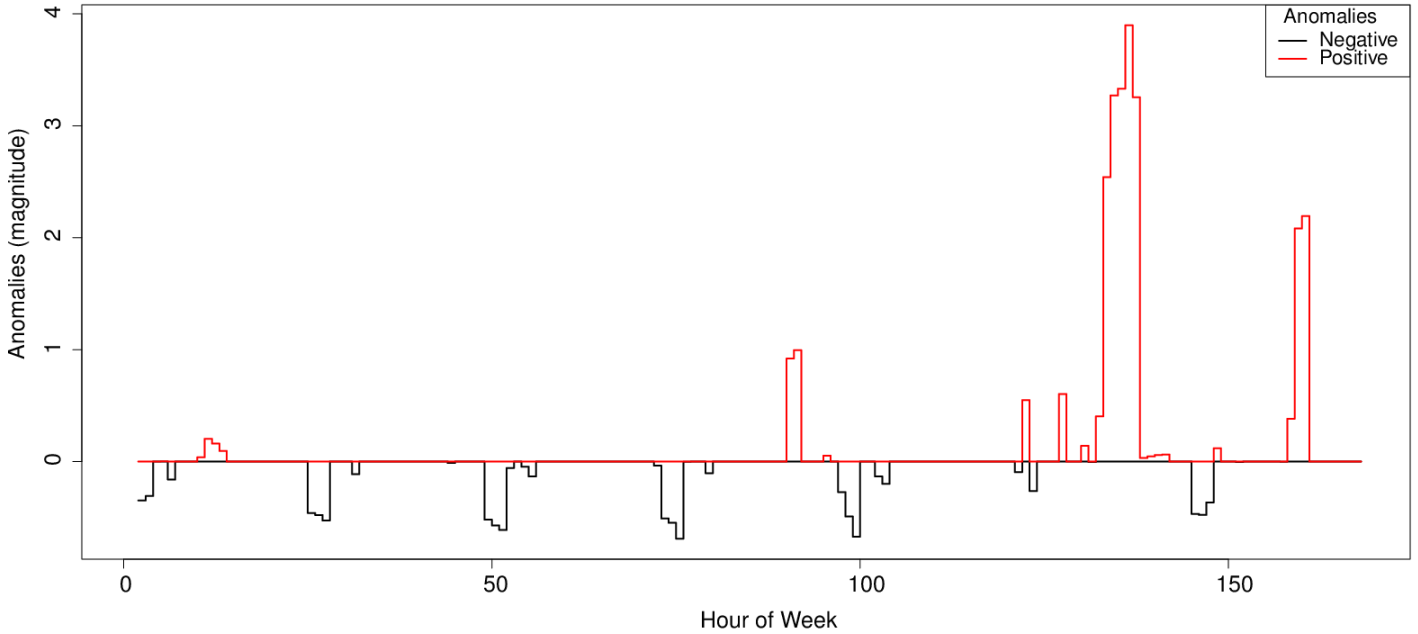
Anomaly Detection Sample

Anomaly flags in week #7



Anomaly Detection Sample

Anomaly magnitudes in week #7



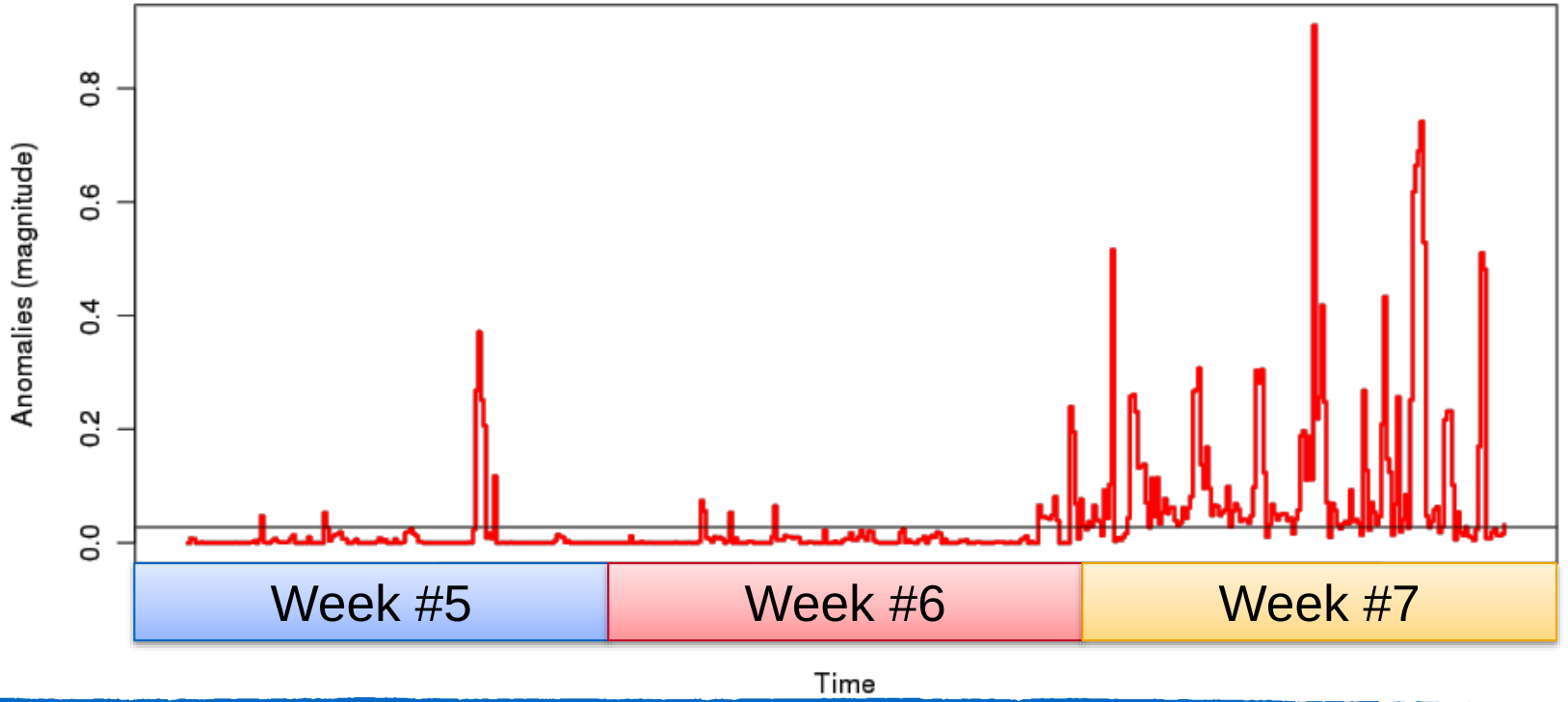
Aggregation

- Single metrics can have random spikes/ noise
  - spurious alerts/ false positives
- Add robustness: combine anomalies across metrics
- Typically need to assess object/ instance; not each metric

- Aggregation Sets: sets of metrics to aggregate together
  - e.g. CPU:#:* or system:system:*
- Aggregation Method: combine anomaly flags & magnitudes
  - mean, median, weighted sum, OR, AND, …
- Percentile thresholds on combined magnitude

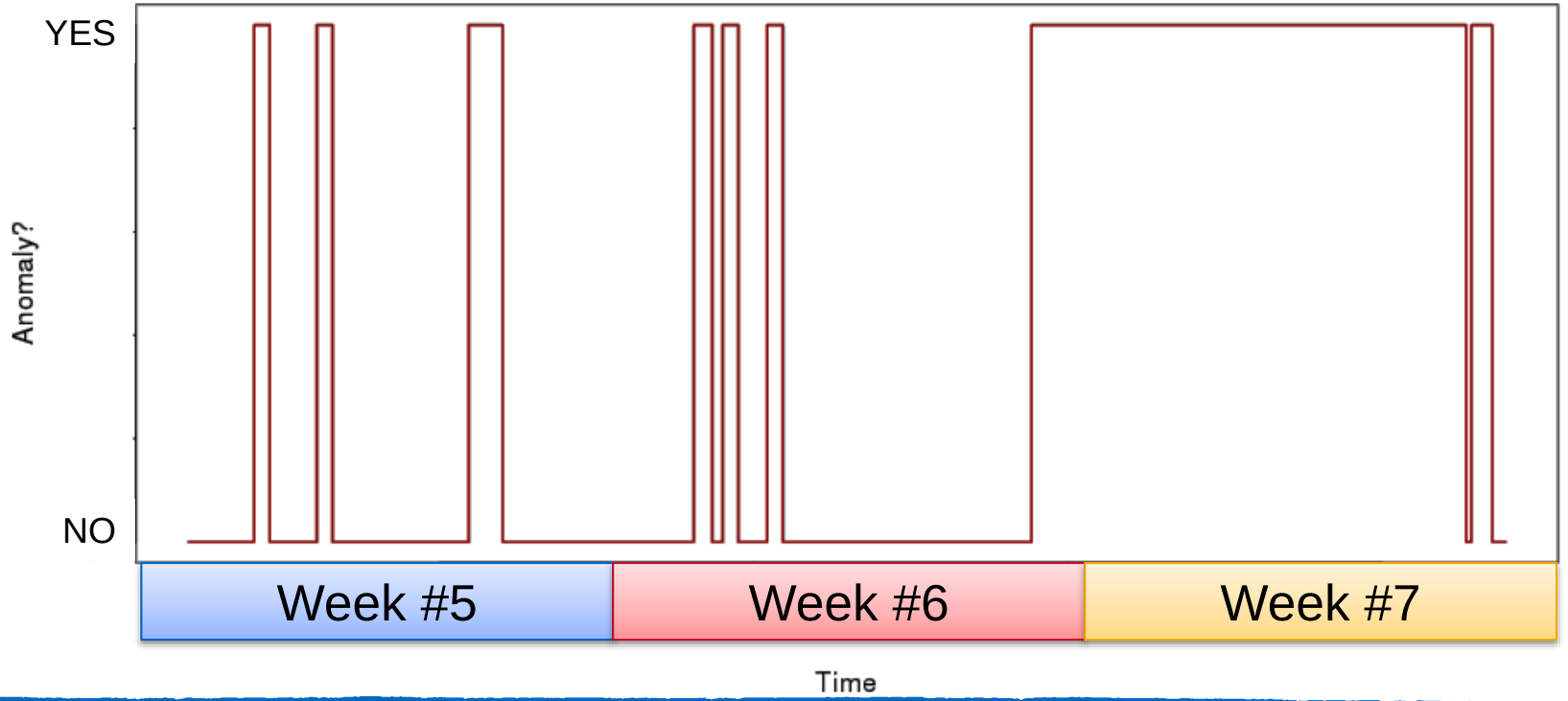
Anomaly Detection Sample

Aggregated anomaly magnitude across all system-level counters



Anomaly Detection Sample

Aggregated anomaly flags across all system-level counters



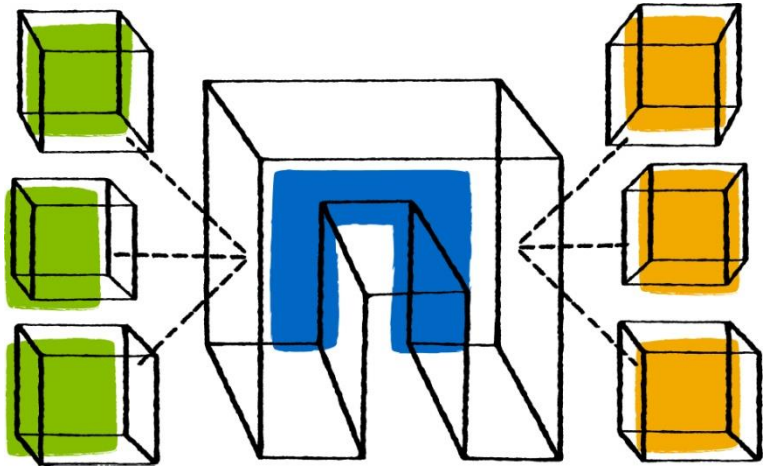
Scoring and Ranking

- Anomaly magnitudes are normalized
  - Comparable across metrics/ aggregation sets/ nodes
- Assign numeric score to each anomaly assessment
  - Anomaly duration; Cumulative magnitude; Avg. count
- Sort by score to get rank
  - Per metric ➔ “top symptoms”
    - E.g. system-wide cache hit rate and partition X read latency showing highest anomalies ➔ maybe workload on X changed to less cacheable
  - Per instance aggregation set ➔ find “most affected” parts

Laboratory Validation

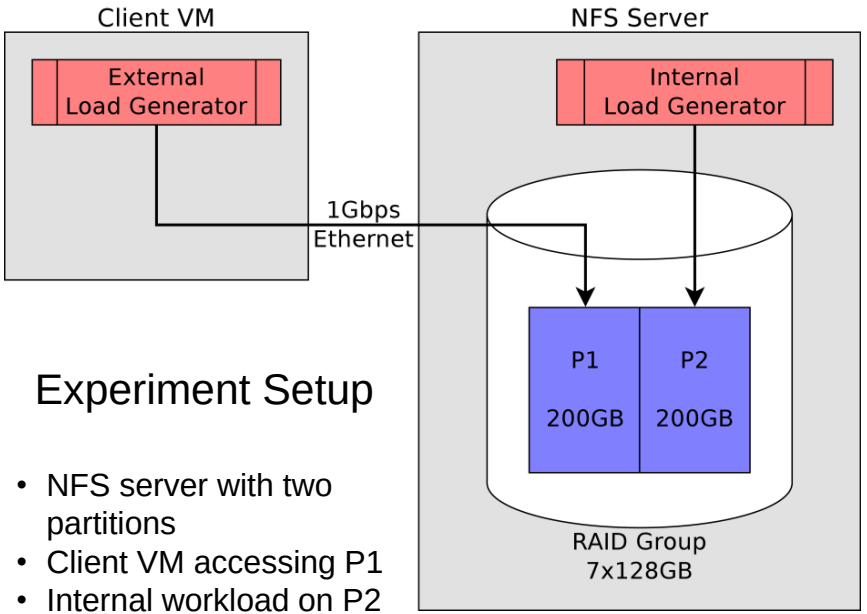
Experiments conducted in

a controlled environment



Lab Experiment Setup

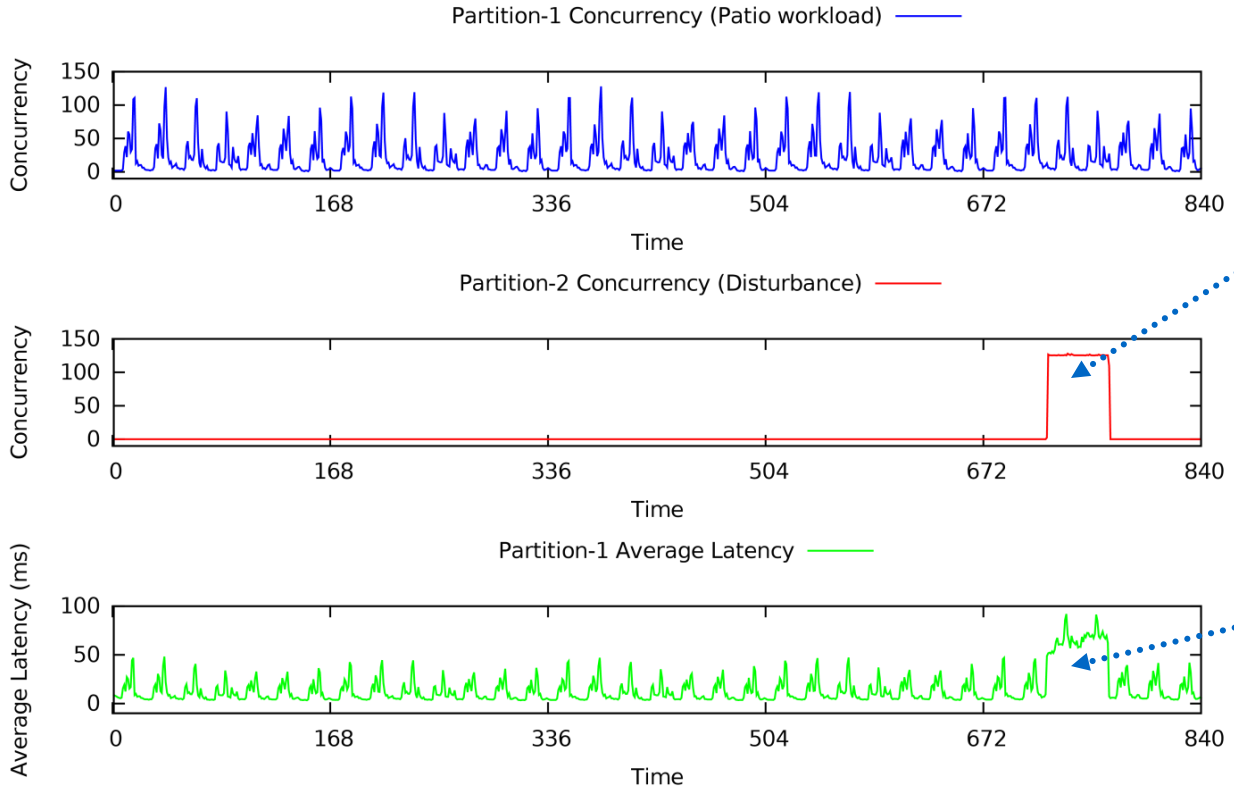
- Client load generator emulates concurrency patterns derived from actual deployed systems
- Trigger several types of disruptions to create performance anomalies
  - Internal workload
  - Failed disk: degraded RAID
  - RAID reconstruction
- Measure impact on client



Experiment Setup

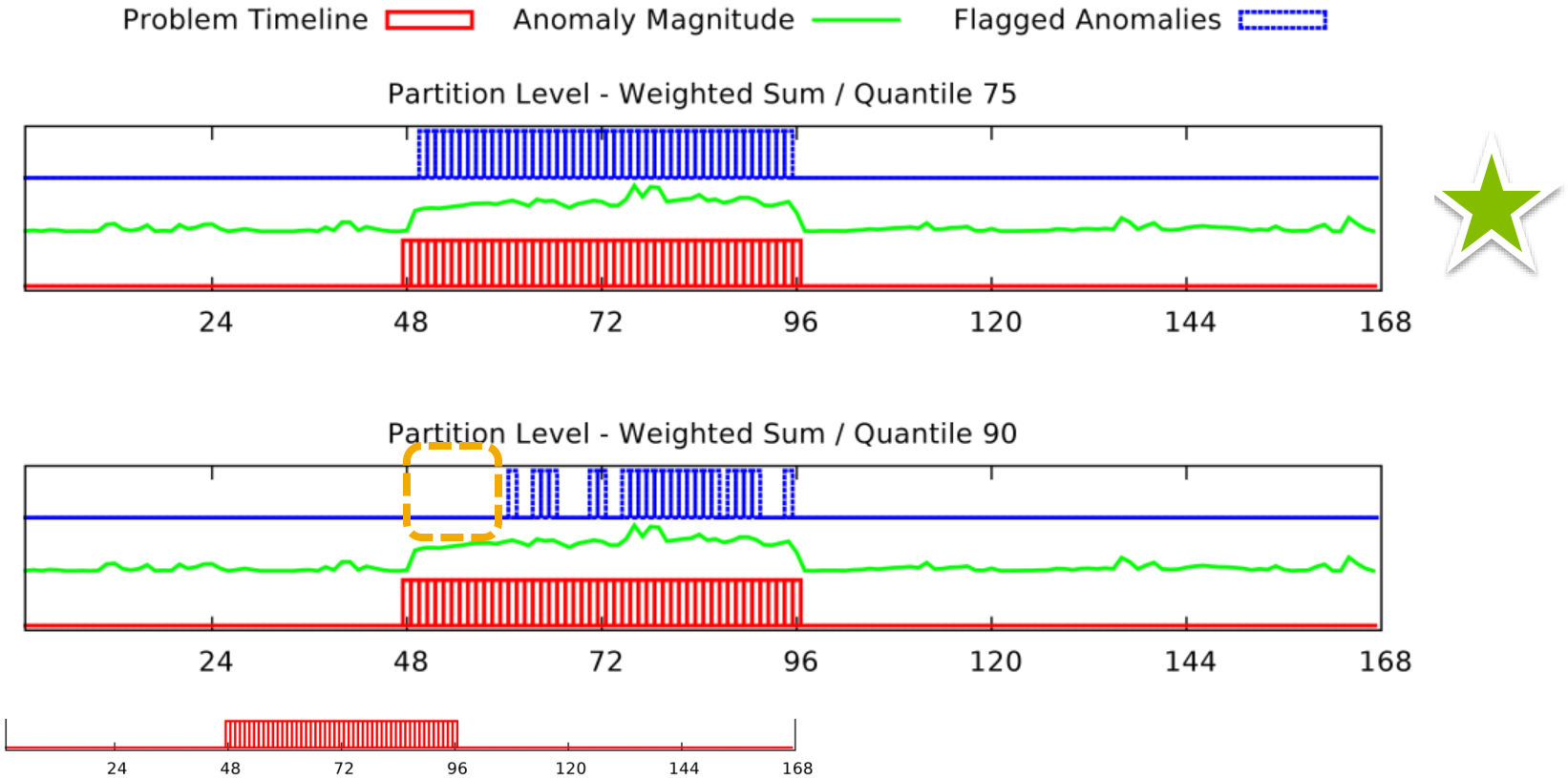
- NFS server with two partitions
- Client VM accessing P1
- Internal workload on P2

Lab Experiment: Sample Run



- 5 weeks worth of metrics shown
- Last week has an internal workload that disrupts client workload (P1)
- I/O on P2 causes latency anomaly in P1

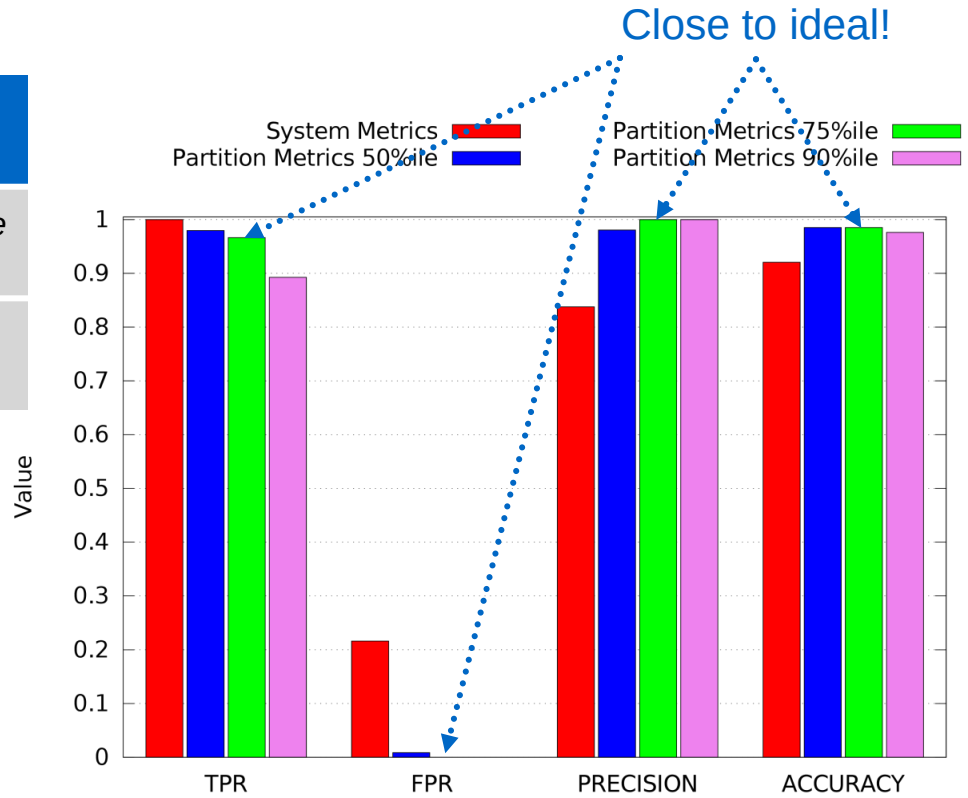
Lab Experiment: Anomaly Detection



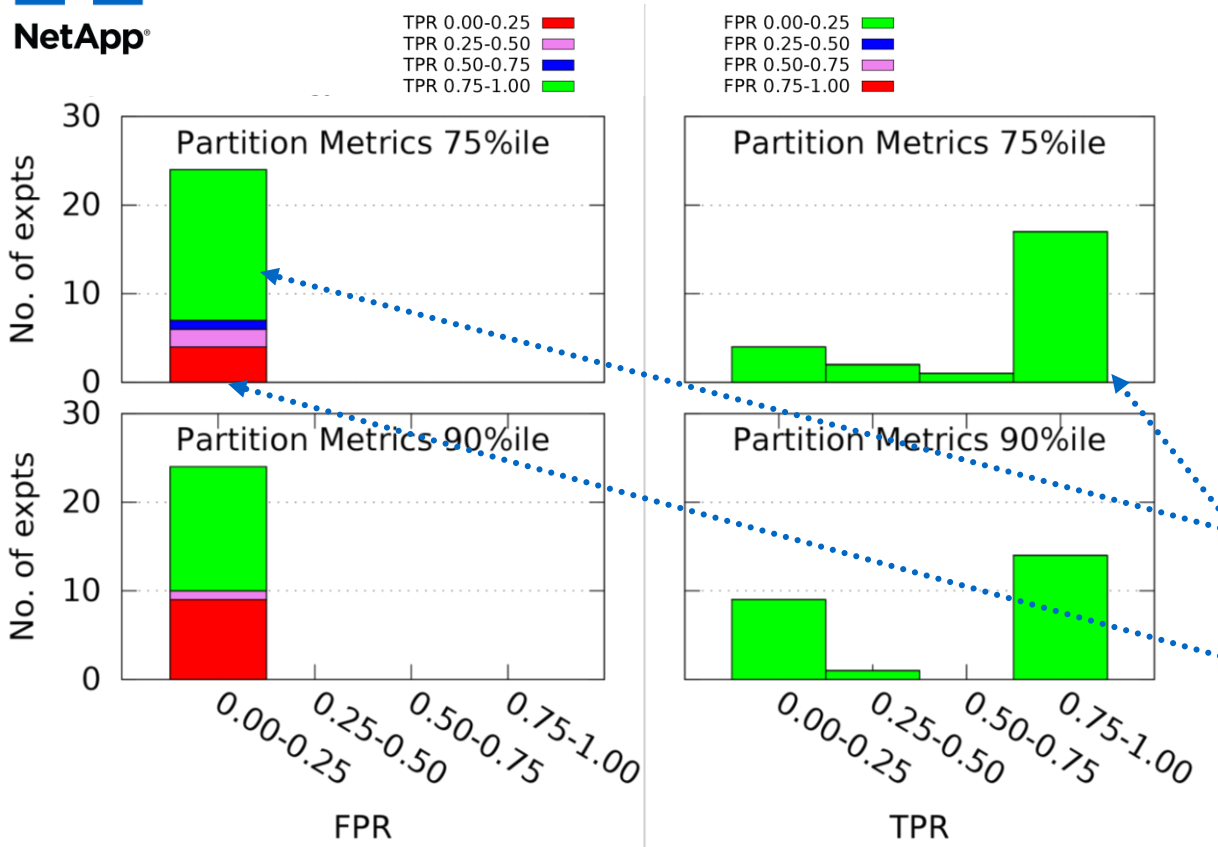
Lab Validation: Summary Stats

Performance Problem	Exists	Doesn't Exist
Flagged	TP	FP
Not Flagged	FN	TN

- True Positive Rate (TPR)
  - Ideally 1
- False Positive Rate (FPR)
  - Ideally 0
- Precision and Accuracy
  - Ideally 1



Lab Validation: TPR & FPR Distribution



- Reminder: stats are for hour-by-hour assessment of 24 exp

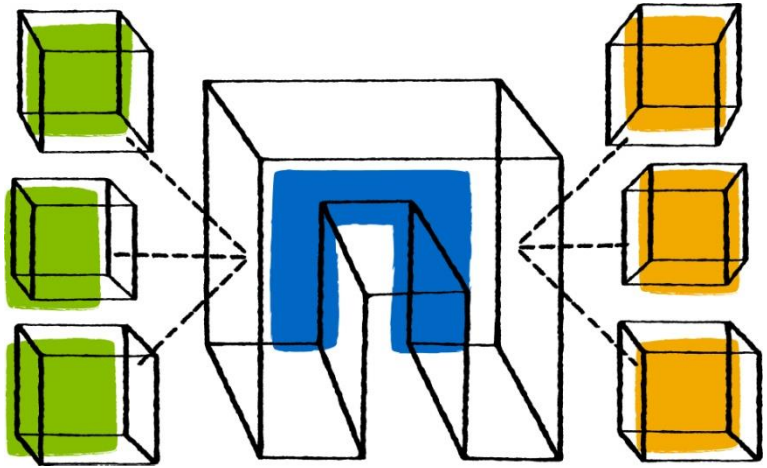
Chosen Assessment

Partition-level Weighted Sum with 75th Percentile Threshold

- TPR is high in most experiments
- FPR is low across all experiments
  - No FPR > 0.25

Field Validation

Analysis of actual customer-reported performance issues

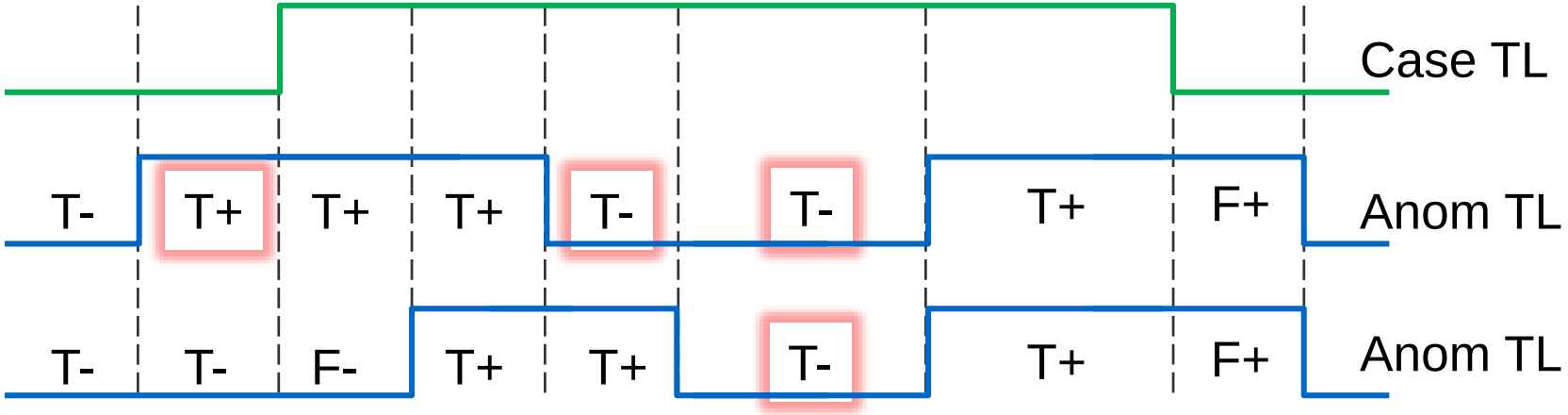


“Ground-Truth” for Comparison?

- Anode assesses performance impact on hourly basis but reported cases only have open and close date
  - How do we compare the two?

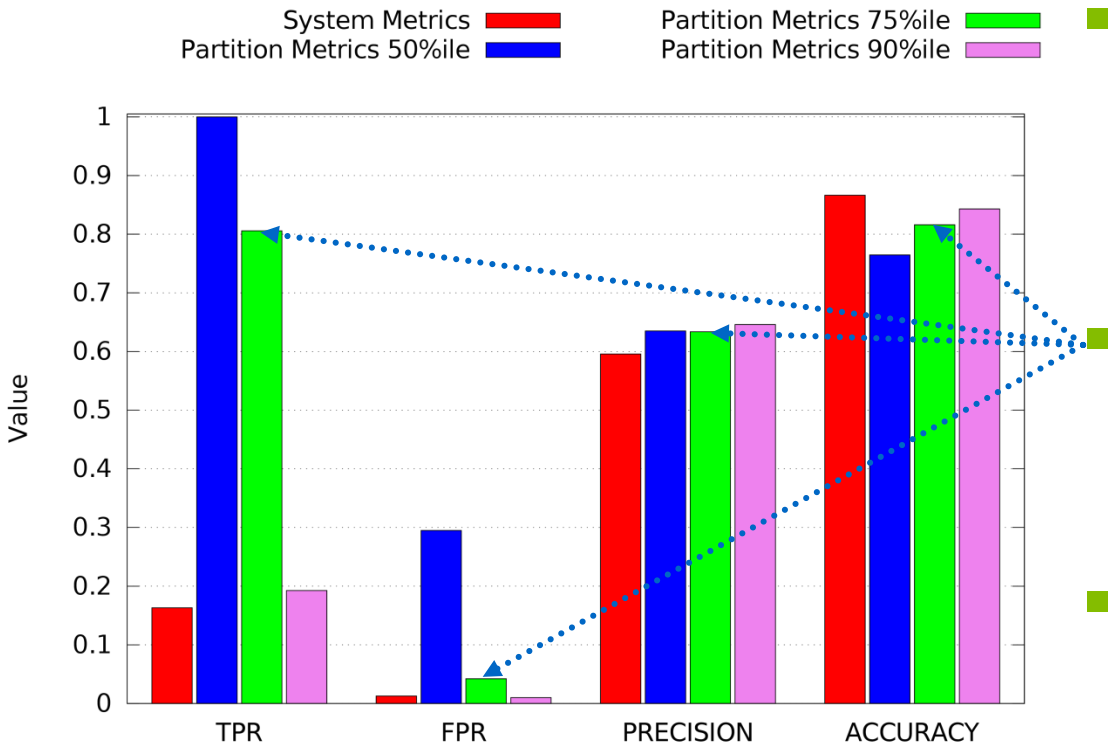
- Performance impact may
  - start before case is opened (usually does)
  - be intermittent, not continuous while case is open
  - stop before case is closed (fix done)

Modified Assessment



- F- before start of anomalies remain F-
- F- after start of anomalies become T-
- F+ after close of case remain F+
- F+ before start of case become T+

Field Validation: Summary Stats



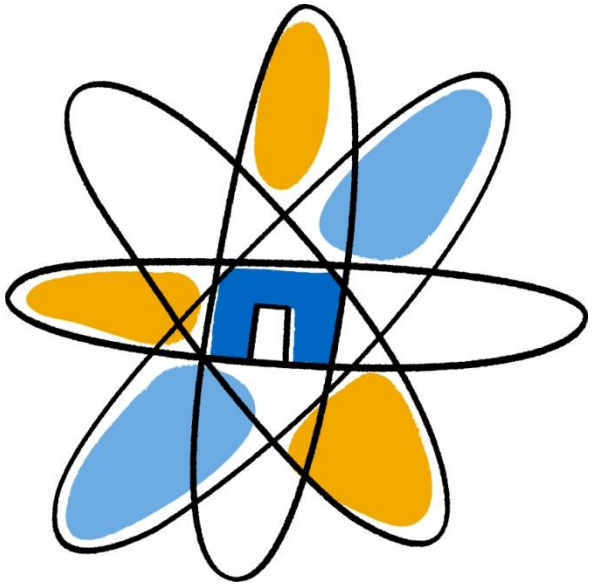
- Reminder: These are median values across 423 actual reported cases
- Chosen assessment performs well in field validation too
- Drill-down available to support personnel

Summary

- We designed a time-series data analysis pipeline to speed up detection and initial triage of performance problems
- Anode gives accurate indications of when and where a performance problem occurred in a storage system
- The core technique is generic and may be extended to any similar system
- Paves the way for quicker diagnosis and fixing of performance problems

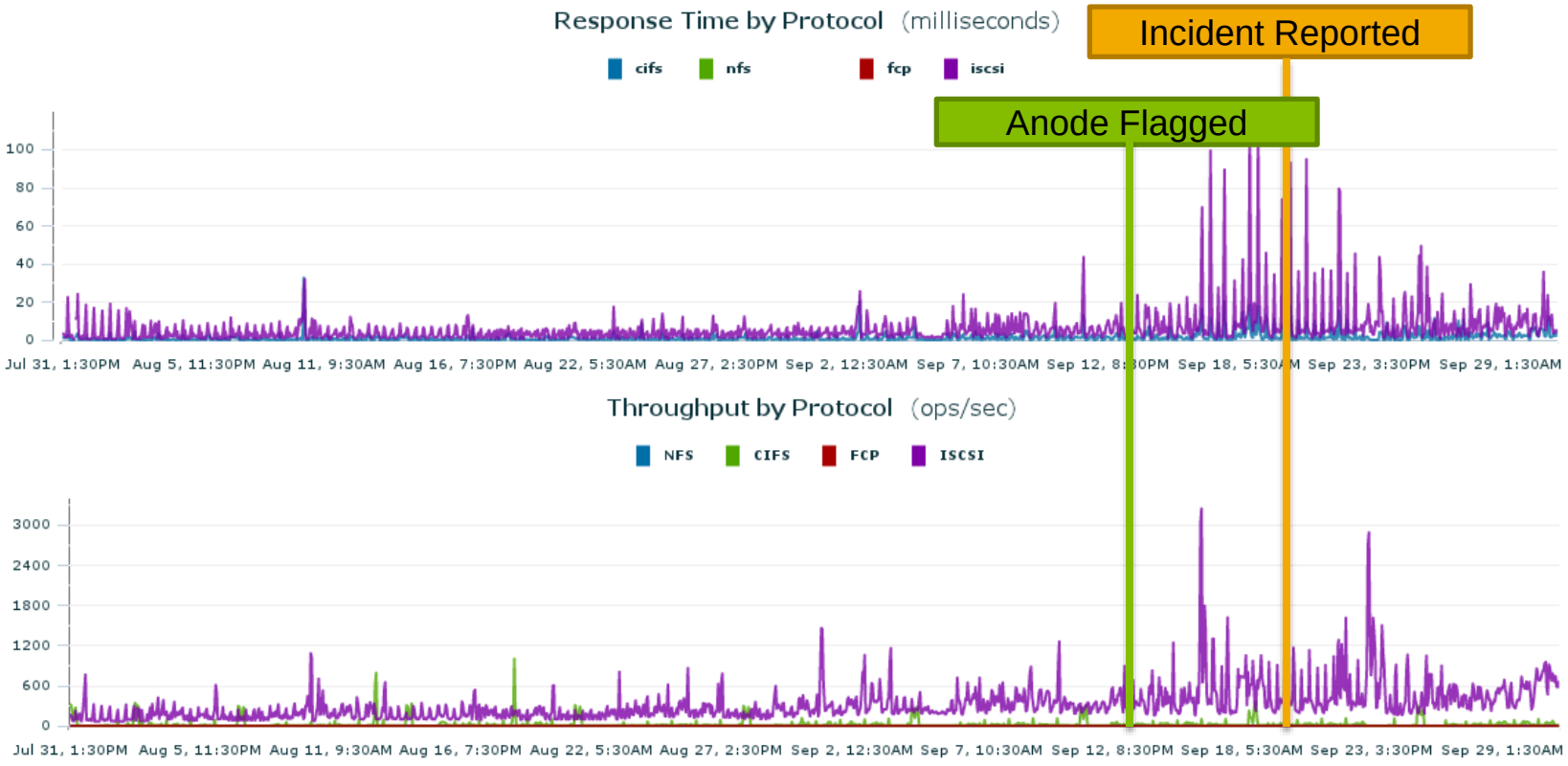
NetApp

Thank you



Vipul.Mathur@netapp.com

Sample Metrics from Actual Incident



Field Validation: TPR & FPR Distribution