2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Big Data in Verification: Making Your Engineers Smarter

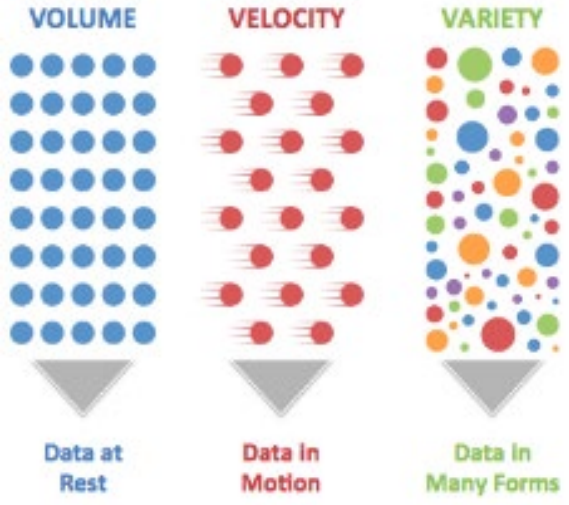
David Lacey, Michael McGrath, Alan Pippin, Ed Powell,
Ron Thurgood, and Alex Wilson

accellera SYSTEMS INITIATIVE

Hewlett Packard Enterprise

2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Big Data in Verification



- What is Big Data Analysis?
- How can you utilize Big Data in Verification?
- How can you do something meaningful with it?

accellera SYSTEMS INITIATIVE

So Much Data – So Many Questions

Compute Farm Data

- compute farm job data
- license usage data
- disk usage
- system load
- per site/business
- per user
- every 6 min

~400M rows/yr
~40 GB/yr for 10+ yrs

Performance Data

- test name
- traffic mixes
- bandwidth
- latency
- utilization
- swept variables and parameters
- per interface
- per topology

~20M rows/yr
~20 GB/yr for 10+ yrs

Test Results Data

- test metadata
- test definition
- tree revision
- pass/fail results
- error sigs
- durations
- memory usage
- cpu usage

~70M rows/yr
~30 GB/yr for 10+ yrs

Big Coverage Data

- per cover event
  - name
  - sim time
  - test id
  - value
- per test
  - name, user
  - cmdline, date
  - generator

~100B rows/mo
~1 TB/mo

Motivation

- Use this data to get our products to market sooner to meet customer needs faster while balancing resource costs by:
  - Driving more accurate forecasts for licenses, compute and engineering resources, schedules
  - Optimizing simulation throughput and performance to maximize our resource cost/cycle
  - Reacting to and fixing issues faster, in real time, as they happen
- Asking the right questions about the data we collect helps our team perform verification tasks smarter by:
  - Increasing simulation throughput and performance
  - Monitoring and reacting quickly to DUT performance issues
  - Improving test effectiveness through more advanced coverage analysis
  - Enabling us to find bugs faster, reducing our time to market

Case Study #1

Compute Farm Data

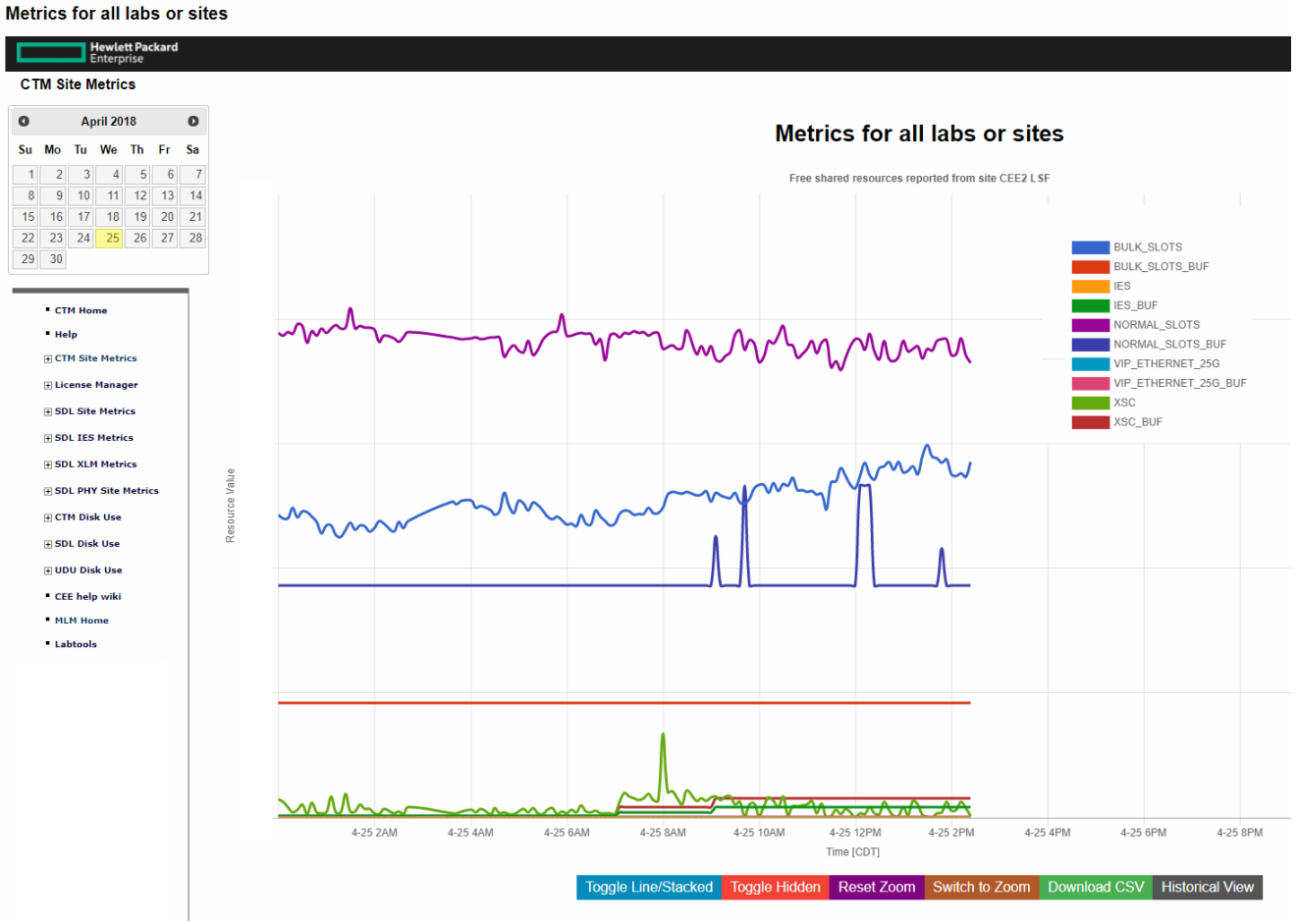
Case Study #1: Compute Farm Data

What specific problems has this big data helped us solve?

License Forecasting

License Utilization

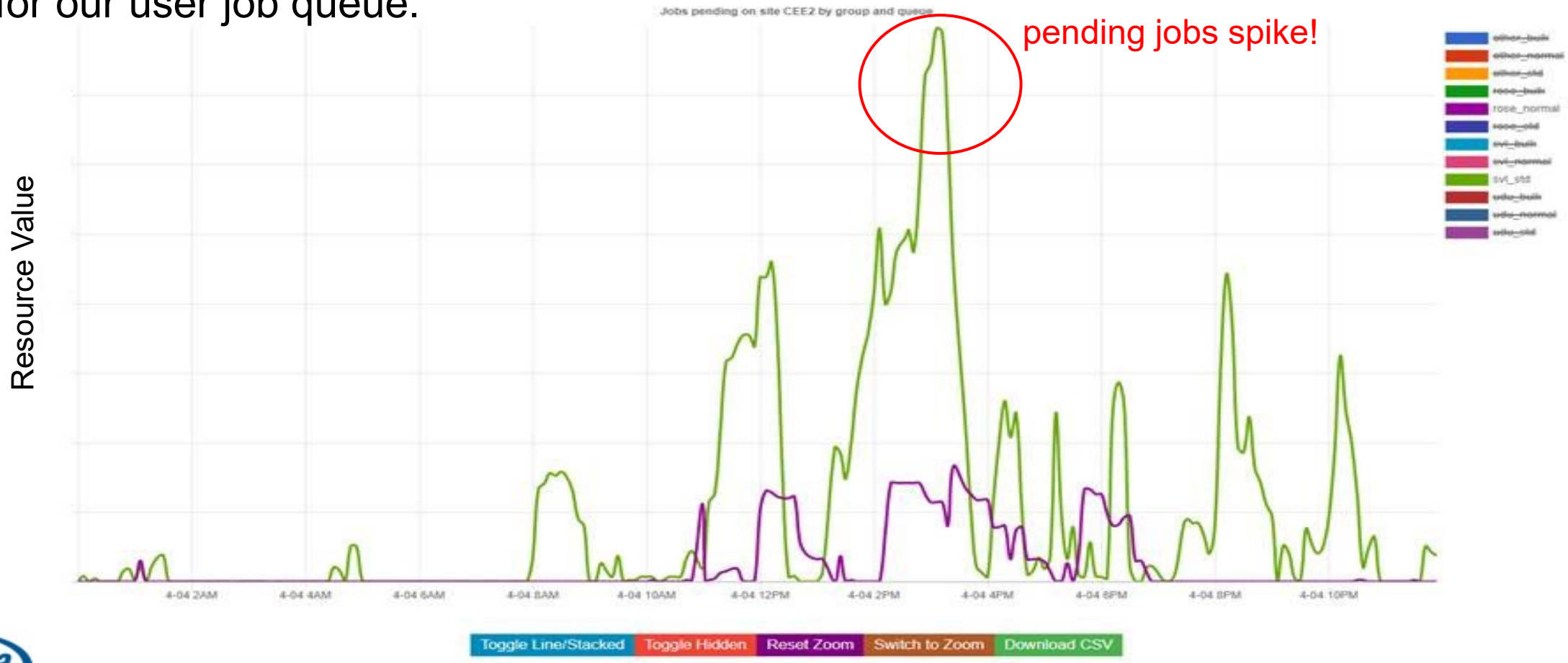
Compute Farm Queueing Issues



Case Study #1: Compute Farm Data

Why are my user jobs being starved for licenses?

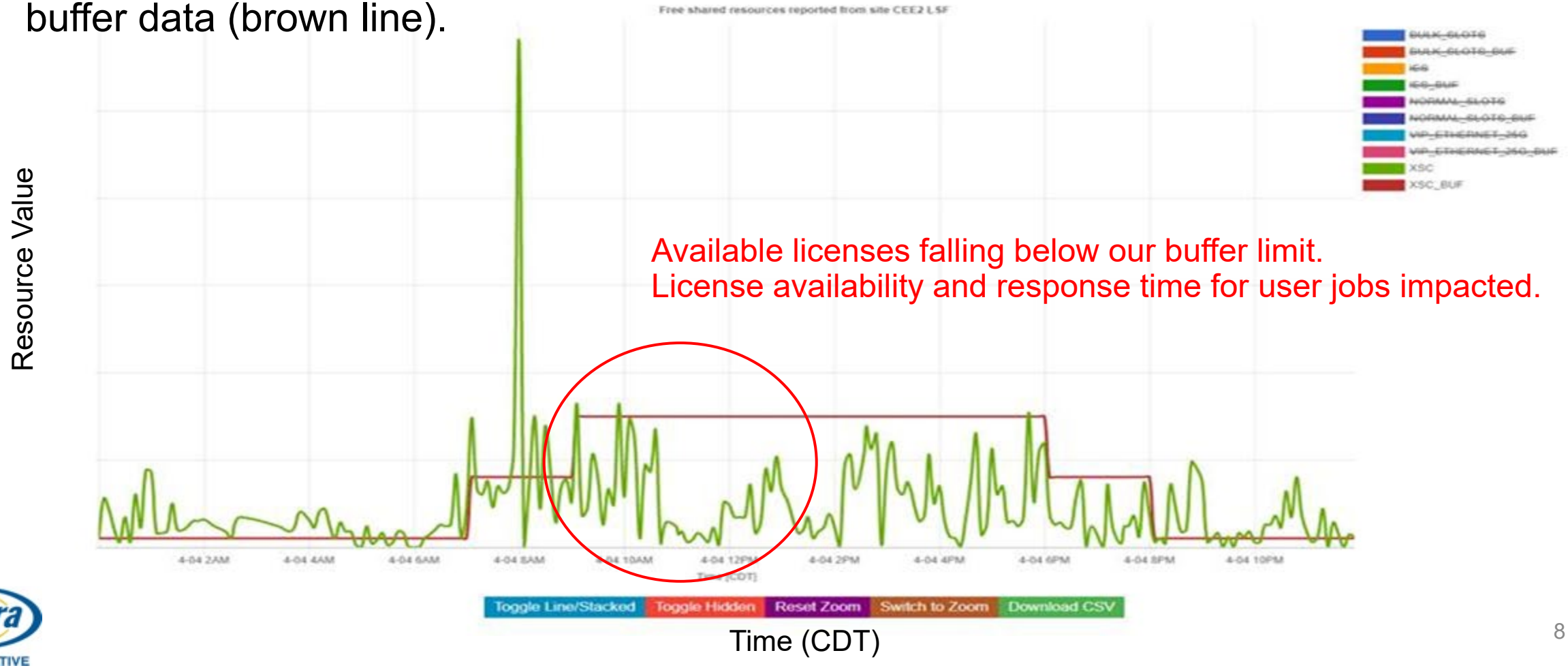
- We analyzed the number of pending jobs data (green line) on our compute server site for our user job queue.



Case Study #1: Compute Farm Data

Did our user jobs have the right priority compared to our automated jobs?

- We analyzed the available licenses data (green line) against our user min available buffer data (brown line).

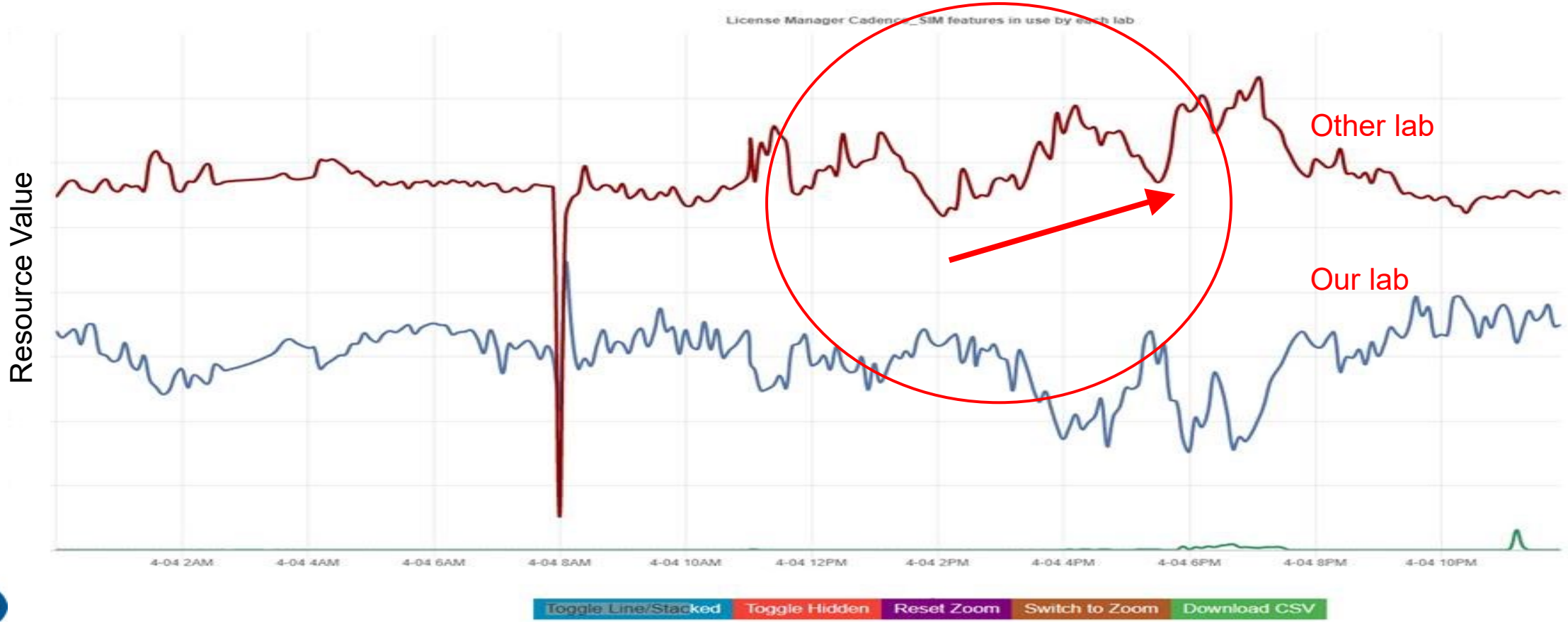


Case Study #1: Compute Farm Data

Which lab was utilizing the licenses during this time?

- We then analyzed our license server data. We discovered there were 2 labs in contention for them.

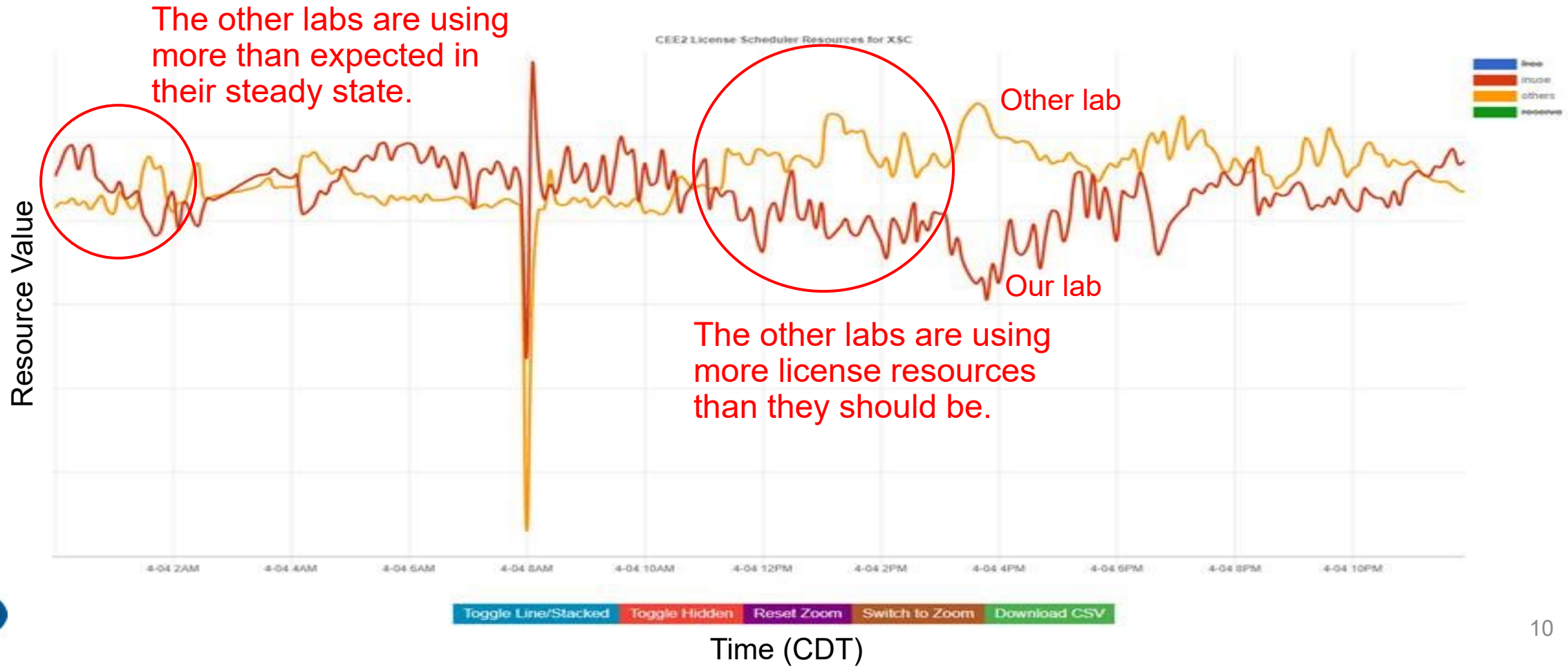
The other lab is steadily taking more licenses from our lab.



Case Study #1: Compute Farm Data

Who was using the resources?

- We analyzed our license scheduler resource data and discovered that another lab was using the licenses outside of our shared compute farm (yellow line) at a higher rate than normal.



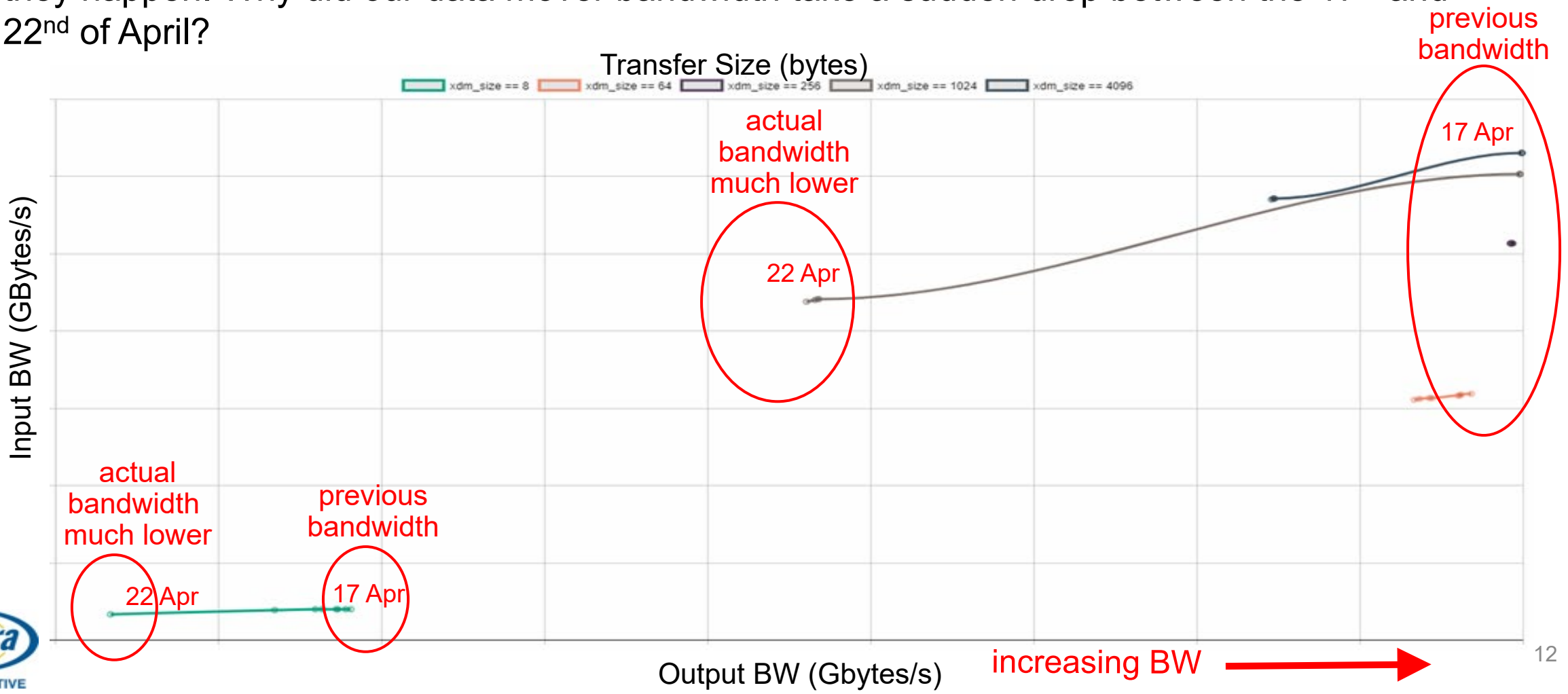
Case Study #2

Performance Data

Case Study #2: Performance Data

Data Mover Performance Anomaly

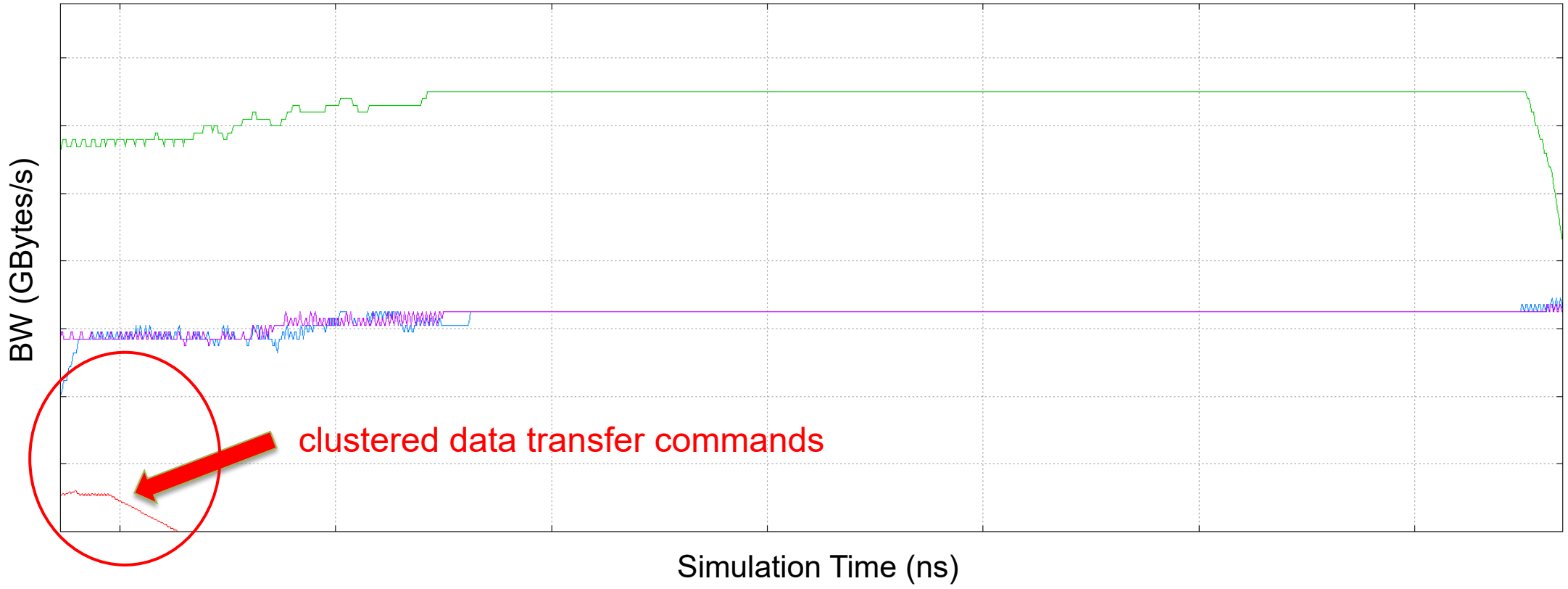
- Our performance data helps us spot performance anomalies in our design in real time as they happen. Why did our data mover bandwidth take a sudden drop between the 17th and 22nd of April?



Case Study #2: Performance Data

Data Mover Performance Anomaly

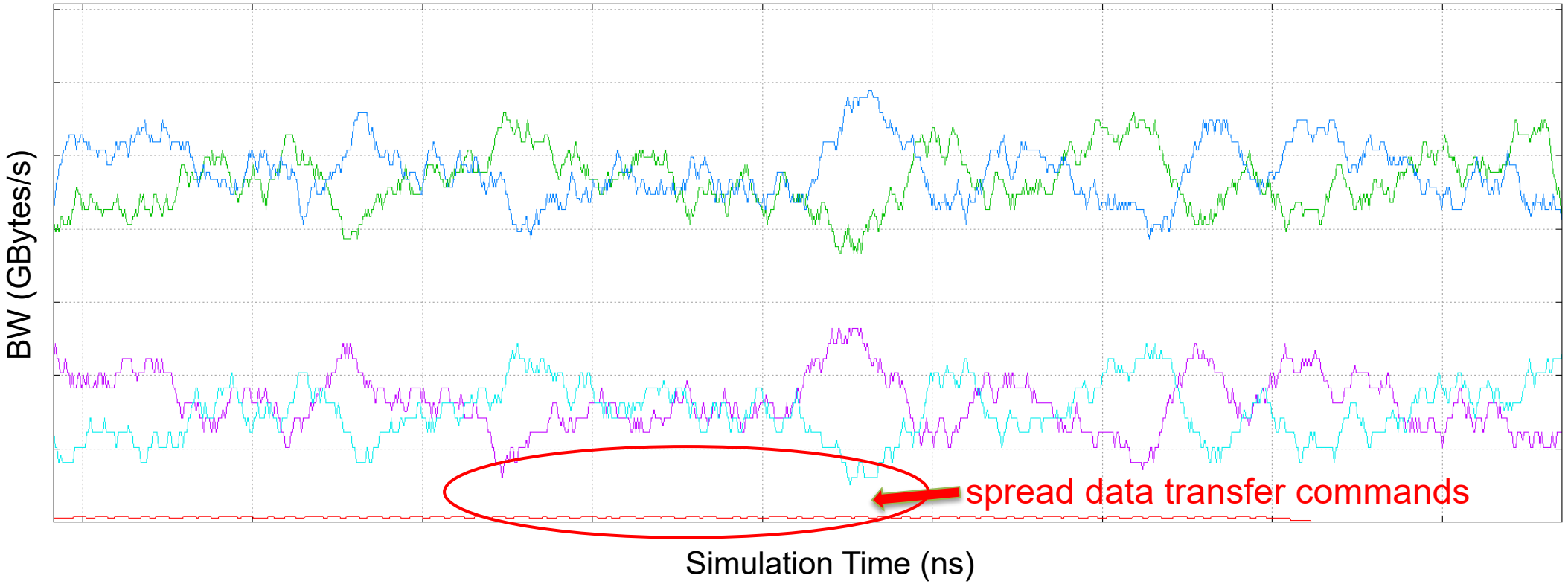
- The data from April 17th showed us that data transfer commands were all being read at the beginning of the simulation vs throughout the simulation, causing our steady state BW to be higher than was realistic.



Case Study #2: Performance Data

Data Mover Performance Anomaly

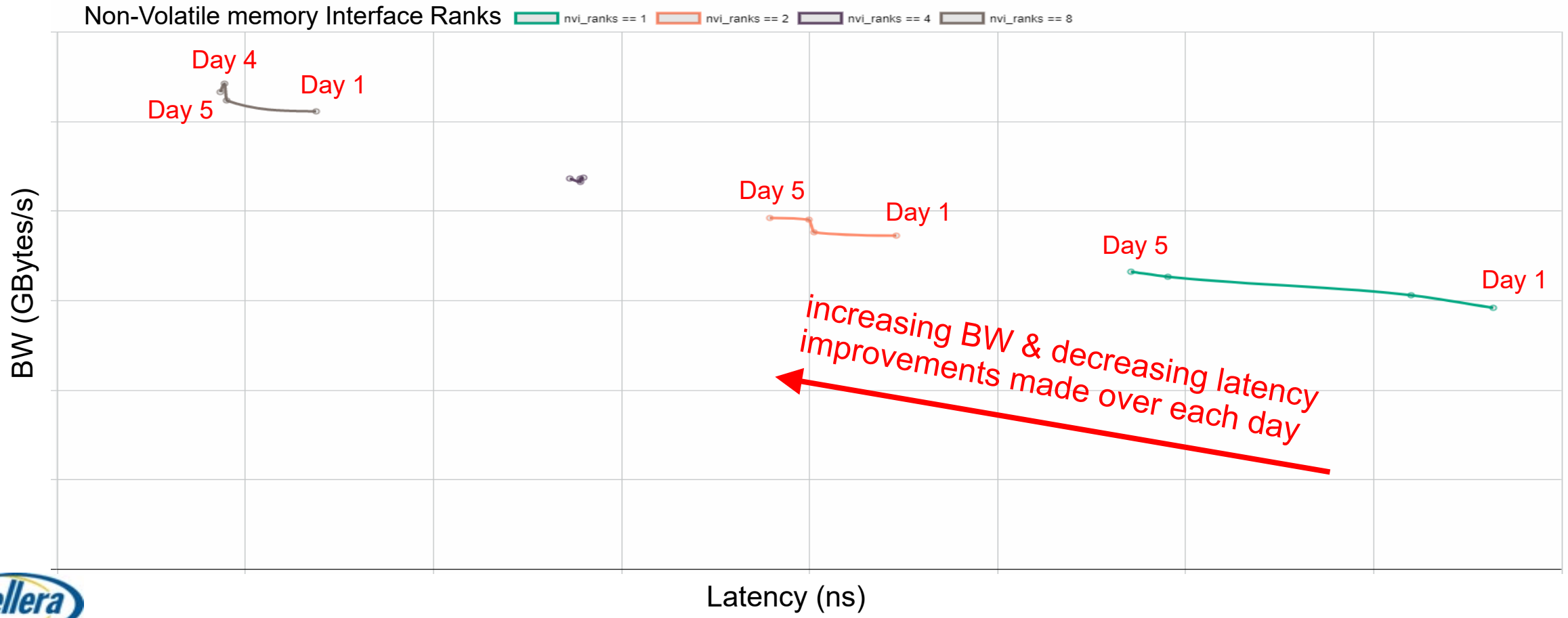
- The data from April 22nd showed us that data transfer commands were now being read throughout the simulation, causing our steady state BW to be lower, but closer to our realistic BW numbers.



Case Study #2: Performance Data

Visualizing the impact of Read BW vs NVI ranks over time

– Our performance data also helps us visualize and track performance improvements over time, in real time, enabling us to quickly identify design changes that adversely impact latency and BW.



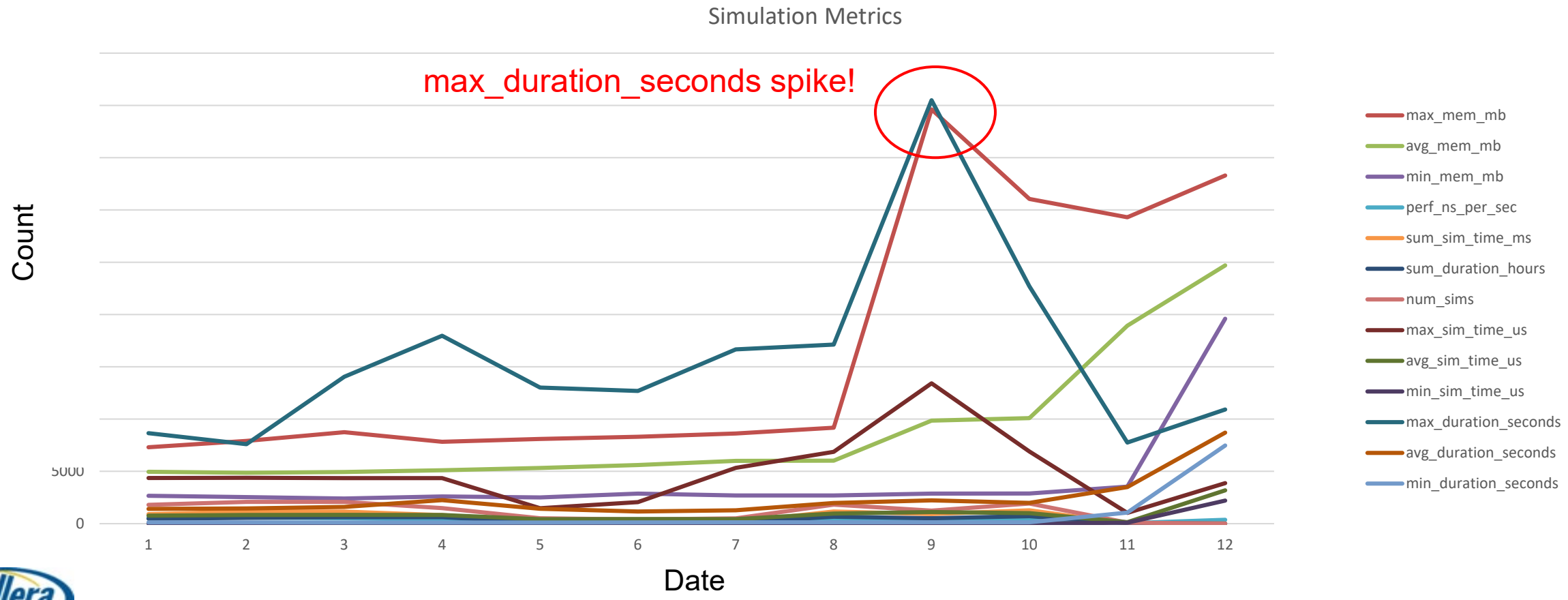
Case Study #3

Test Results Data

Case Study #3: Test Results Data

Why are my tests running slow and taking longer than expected?

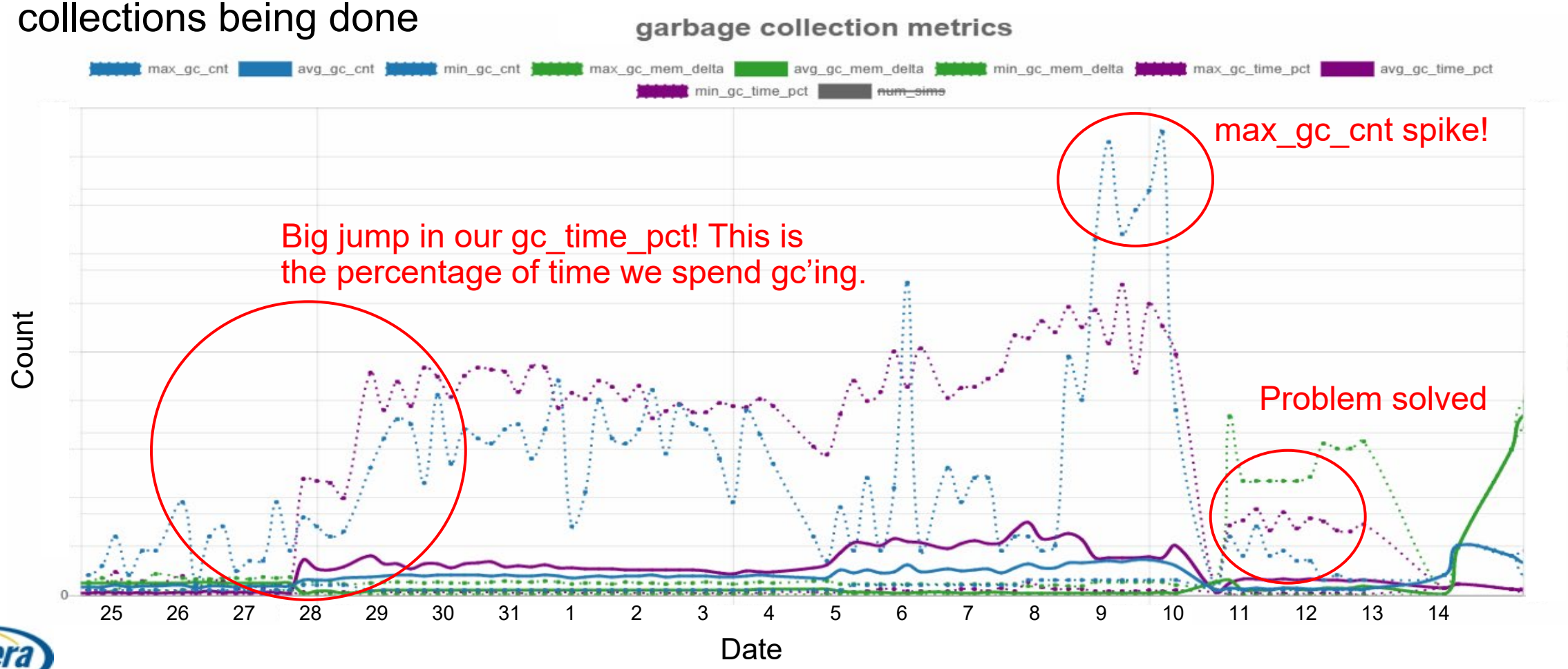
- We analyzed our test results data and found our simulations were running 6x longer than normal



Case Study #3: Test Results Data

Why are my tests running slow and taking longer than expected?

- We analyzed the data further and correlated a huge spike in the number of garbage collections being done



Case Study #3: Test Results Data

Other ways to use the data

- How many cycles are we running for each testbench?
- What are our pass and fail rates? What is our time to first failure?
  What tests are finding the most bugs? What types of bugs are we finding?
- How does the bug rate compare with verification cycles and cycles over time?
- My test distribution doesn't look right. Am I getting the right distribution of tests?
  Are any being starved or not running as often as they should be?
- Is my test simulation performance optimal? What can I do to make it better?
- I want to upgrade my compute farm. What configuration do I need?

Big Coverage Data

Big Coverage Data

Big Data Coverage Toolset: Recording user defined events over all of sim time

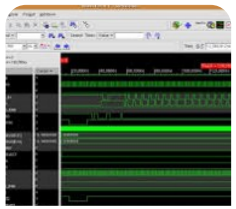
Record test events/activity throughout the simulation	Define functions that operate on the data	Bind functions to events/registers/ activity to create metrics
Use test filters to apply metrics to groups of tests	Visualize metrics across tests and regressions with charts	Incorporate metrics into other tools with a REST data service

Big Coverage Data

Recording test events/activity and Defining functions that operate on the data

Instrumentation

Testbench Code



Simulation

Events Captured

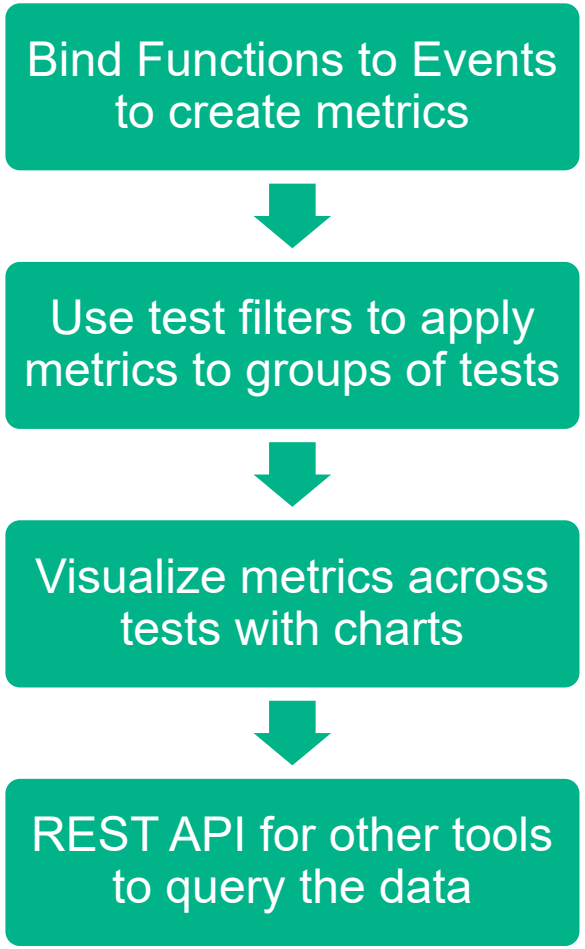
Events Stored

Primitive functions can be applied to stored events to extract metrics

Arithmetic	Logical	Bitwise	Time-based	Series→Scalar
• add, sub • mul, div • abs, neg	• and, or, not • eq, gt, ge, lt	• bitShift • bitAnd, bitOr • extract, popcnt	• length • timeShift • reverse • timeOfFirst	• integrate • min, max • median, mean

Big Coverage Data

Define metrics and analyze the data through a modern web interface



- The user-defined functions that engineers create produce valuable metrics
- Filters create powerful views of the data across multiple tests enabling you to see new interactions
- Waveforms are visualized with line charts
- Scalar values are visualized with bar charts
- Scripts or other processes can ask how a test performed against a specific metric

Where do we go from here?

A Big Data Toolkit

Collect

- Structured DBs
  - Maria
  - Postgres
  - Oracle
- NoSQL DBs
  - Hadoop
  - Vertica*

Analyze

- Direct SQL queries
- Perl or Python ORMs (DBIx / SQL Alchemy)
- Excel* with pivot tables and filtering
- MySQL Workbench

Present

- Web Frameworks
  - Java / Javascript
  - Python / PHP (Django/Yii)
  - Dynamic / Static HTML
- Plotting
  - Excel*
  - GNUPlot / ChartJS
  - Grafana
- Textual
  - Perl / Python / Ruby

* = commercial tool

Future Applications

- What are the possibilities with Big Data?
- There is huge opportunity here for us and EDA vendors!

- Intelligent test grading - More value from our cycles
- Intelligent stimulus generation & Coverage driven stimulus
- Storing more dynamic unstructured test data (as needed, on the fly)

- Machine learning (for stimulus generation)
- Presentation improvements (heatmaps)
- Querying and merging data across multiple data domains

Conclusions

Big Data is no longer big and scary.

It is in reach for every verification engineer.
You just have to get started.

It's not just one type of data.
Collect the data that will give you the information you need.
How you collect it, where you collect it, make a difference.

When you leave today, you should to be asking yourself:

- What questions can you ask to make your verification more intelligent?
- What data can you start collecting today to answer those questions?