Load Testing at WB Games SF

Why Load Test?

- Understand how a system performs with concurrent users.
- Learn where “hot spots” are.
- Know, with Science™, that a system can sustain concurrent users.

Who Are The Stakeholders?

- Executives funding platform efforts.
- Users expecting a decent experience.
- Platform clients, game teams etc.
- Platform engineers.

What Is Load Testing?

Stress Testing

- Run on smallest form factor, i.e. one core Virtual Machine.
- Hit individual APIs repeatedly with N threads.
- Used to determine max load per core.
- Used to determine limiting factor of a service.

What Is Load Testing?

Representative Testing

- Sessions comprising a series of API method calls that accurately represent real user activity.
- Sessions include gaps between calls.
- Used to determine limiting factor of production scale infrastructure.

When to Load Test?

Local Tests

- Each time a new feature with a unique set of performance characteristics is introduced.
- e.g.
  - Reads from cache only.
  - Reads from cache, writes to persistence layer.

When to Load Test?

Representative Tests

- 2-4 weeks before a major launch.
- This gives an appropriate amount of time to address issues found.

Where to Load Test?

Local JMeter

- Quickly allows an engineer to determine if there’s anything glaringly wrong with new feature work.

Where to Load Test?

Remote JMeter Stress Testing

- Run against smallest form factor VMs.
- Separate concerns across VMs.
  - Service Nodes
  - Caching layer
  - Persistence layer
- Used to find hot spots and weaknesses in various APIs and abstractions.

Where to Load Test?

Multi-region JMeter hitting production environment

- Run against full scale environment.
- Instances based on sizing exercise(s)
  - Caching memory requirements
  - Persistence storage requirements
- Used to demonstrate reliability of infrastructure at scale.

How to Load Test?

JMeter

- JMeter is a well documented, java-based GUI on top of XML file.
- See http://goo.gl/cDweu6 for more details.

How to Load Test?

Choose your target metrics

- We chose the following
  - CPU Usage <= 80%
  - Response Times <= 100ms
  - Errors = 0

How to Load Test?

Node --prof mode

- Great during stress testing to determine what you’re calling most.
- Great to see changes working.
- Make sure to run tests for 2-5 minutes to filter out the noise*.
- Don’t sweat the small stuff**.

How to Load Test?

What to watch for

- Low CPU, high response times
  - Indicative of waiting on network data.
  - e.g. reading from or writing to persistence layer.

How to Load Test?

What to watch for

- High CPU, high response times
  - Indicative of disk I/O.
  - Indicative of JSON.parse / JSON.stringify heavy workload.
  - Errors Abound!

How to Load Test?

Networking related errors

- ulimit
  - Set *ulimit -n* to 65336*
  - Set *ulimit -u* to something above 1024, per your use cases (this came up with our Mongo hosts)

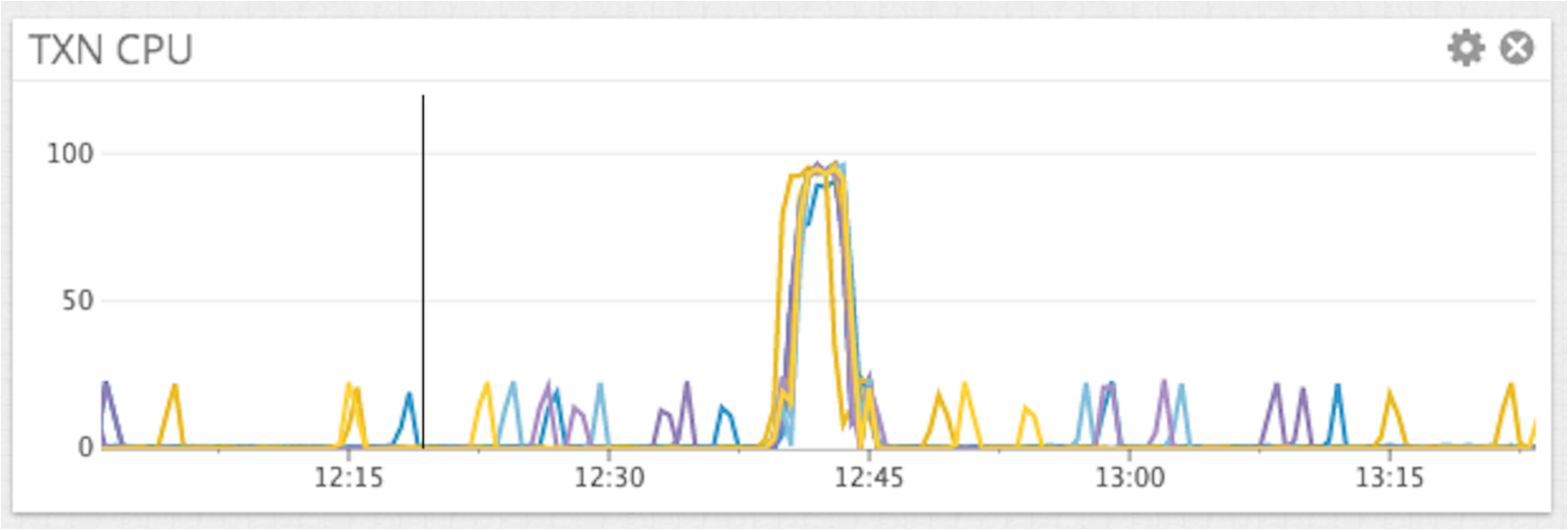
How to Load Test?

Networking related errors

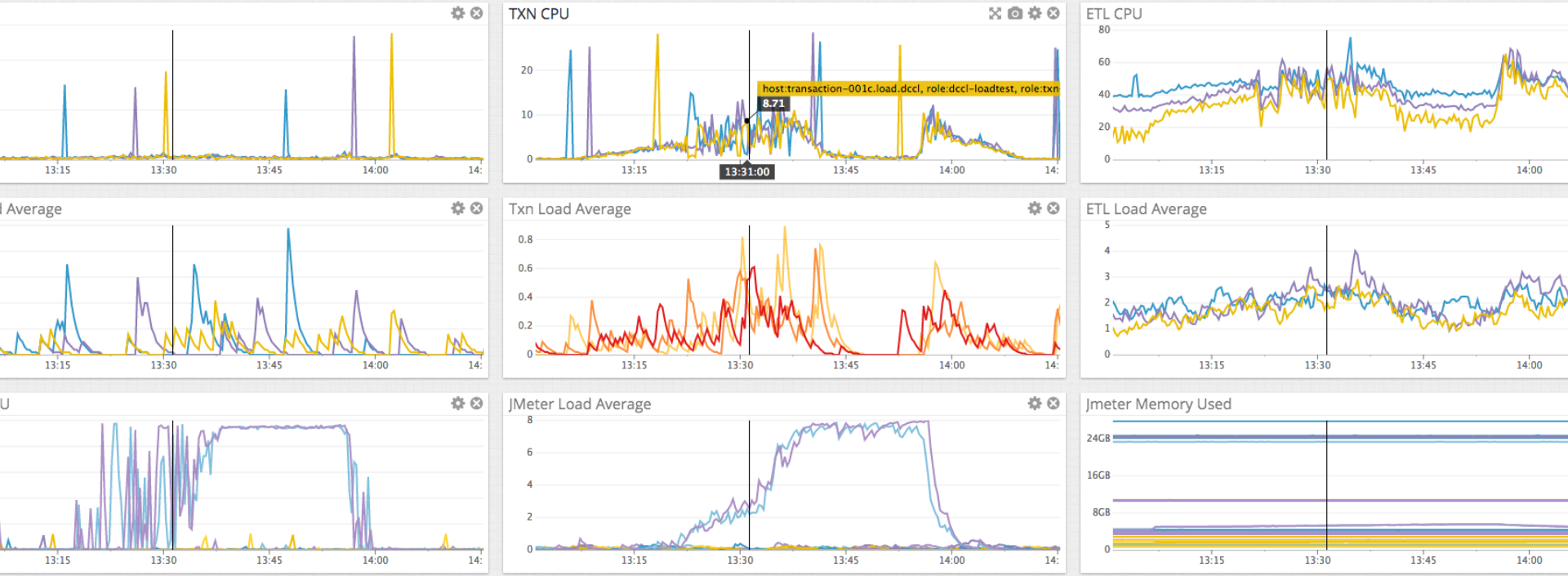
- kernel level TCP settings
  - tcp_tw_reuse
  - tcp_ip_port_range
  - (see http://goo.gl/o2YJqs)

And Now Some Graphs!

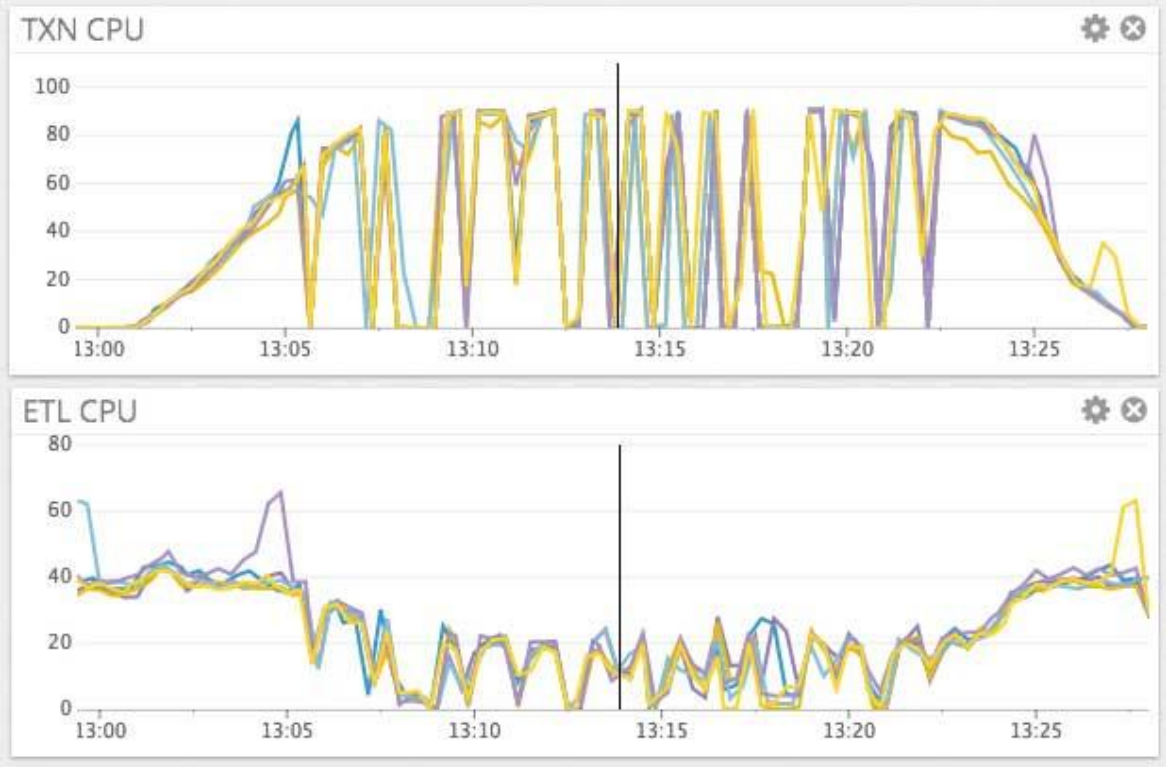
- 100% Errors, 100% of the time!



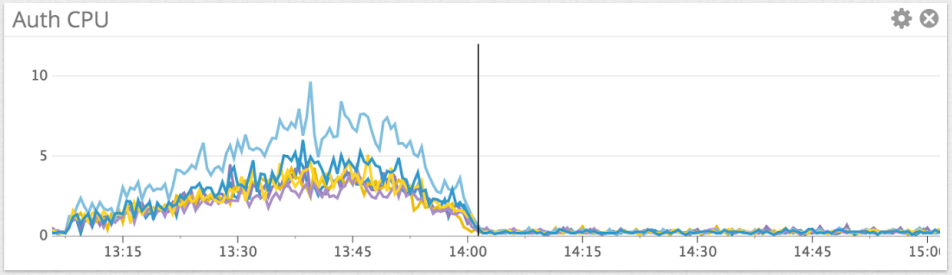
Pushing JMeter to It's Limits

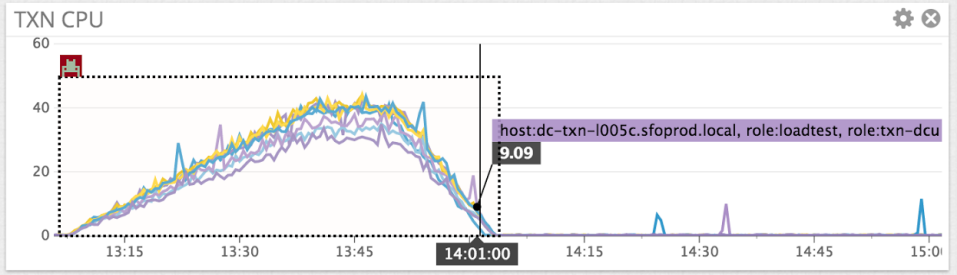


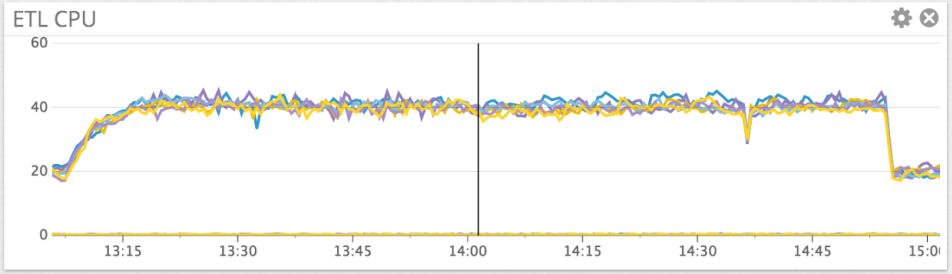
Severe Network Throughput Issues

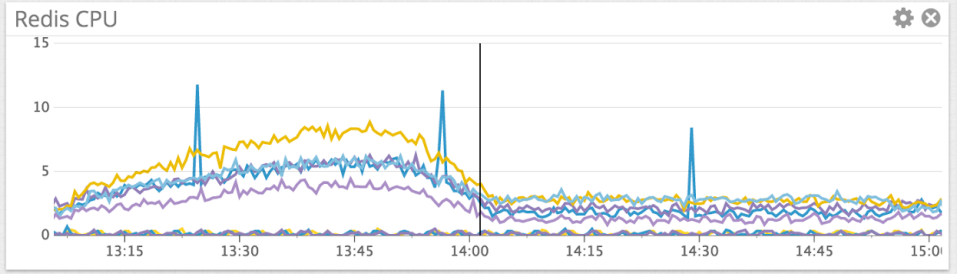


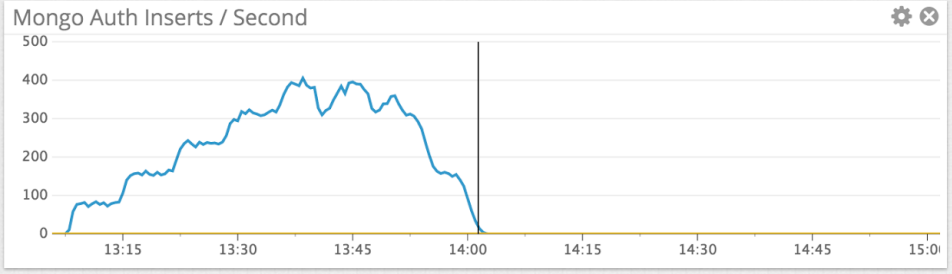
A Healthy Test Run!

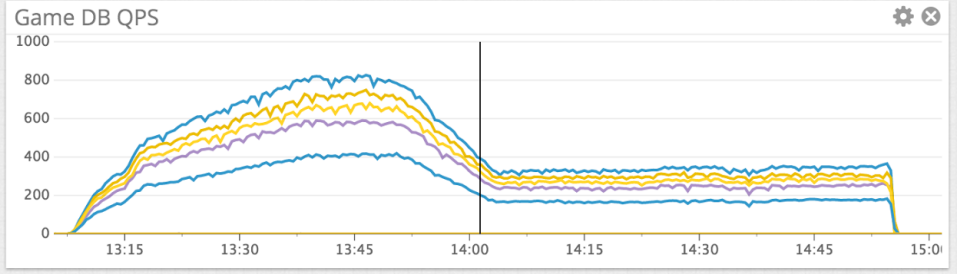












Questions?

Hit Me Up!

@elrasguno

http://attnspan.com