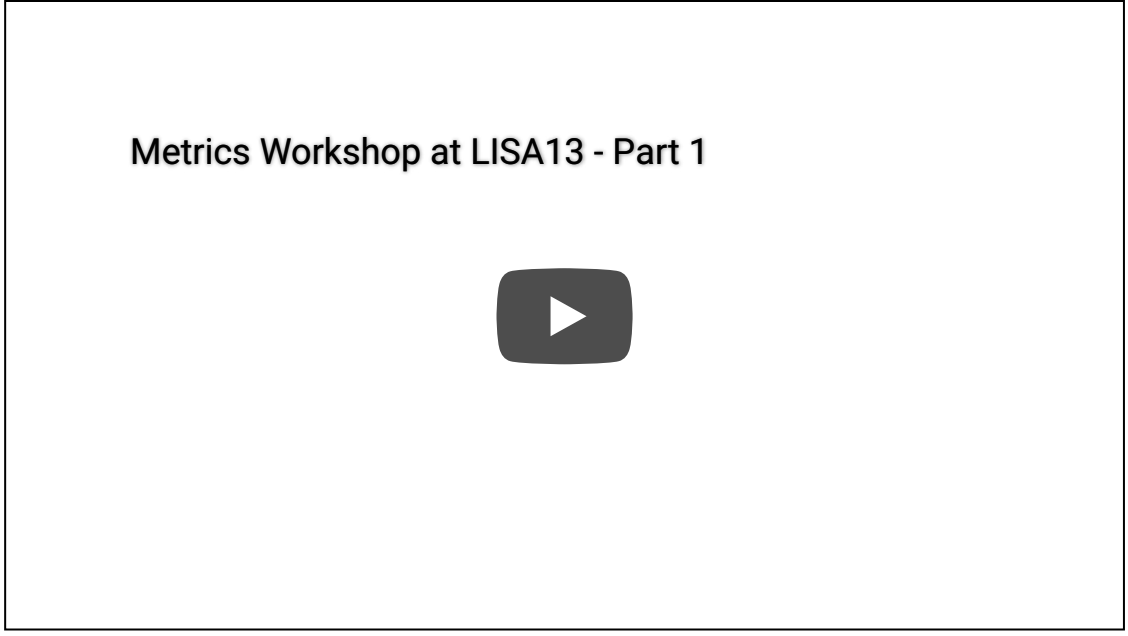
USENIX/LISA 2013 Metrics Workshop

16 May 2014

At USENIX LISA'13 I helped run a Metrics Workshop, along with Narayan Desai (Argonne National Laboratory), Kent Skaar (VMware, Inc.), Theo Schlossnagle (OmniTI), and Caskey Dickson (Google). This was an opportunity for many industry professionals to discuss problems with performance metrics and monitoring, and to propose and discuss solutions. It was a lot of fun, and was very useful to hear the different opinions and perspectives from those who attended.

We provided guidance for choosing more effective performance metrics, which involved helping people think more freely and creatively, instead of being bounded by the metrics that are currently or typically offered. I also covered key methodologies, including the USE Method, which provide a checklist of concise metrics that are designed to solve issues. I ended the day with five minute lightning talks on statistics and visualizations.

There were about 30 participants, and Deirdré Straughan videoed the entire day long event, which includes the talks by the other moderators. The videos are on youtube:



As an exercise, we identified several targets of performance monitoring, formed groups to propose ideal metrics, then presented and discussed these metrics. I've listed a summary of the metrics below, and also submitted them to the monitoringsucks project on github.

Network Infrastructure

- Physical Infrastructure
    - bandwidth, utilization of individual links
    - CoS/QoS rate/drops
    - L2/L2 protocol health
    - churn
    - reachabality
- Per port:
    - packets/sec
    - packet size
    - buffer utilization
    - perf flow into:
    - app injection BW
    - app injectiov rate
    - app consumption rate
    - app consumption BW
- Component:
    - links
    - errors
    - latency
    - utilization
- Topology:
    - app to app latency
    - app to app low
    - symmetry

Configuration

- Apps should export flags, to check for consistency
    - a metadata to show the target configuration
- Versioning:
    - ldd, libraries linked against
    - time a config was applied

- Platform Type:
    - server H/W
- Cost of Configuration
    - cost of configuration upload/download
    - time to deployment: security changes (high priority), vs others
    - CPU and RAM usage during configuration
- People
    - deployment report
- Hardware
    - current hardware
    - max expected performance
- Process
    - compliance measurement of configuration: percent of systems
- Failure
    - failure of configuration deployment
    - rollbacks, rollforward: config metric didn't apply
- OS flags

Distributed system

- Perceived latency: service time and queueing
- Request rate
- Error rate
- Traffic origins
- Histogram of latencies for each server, for comparisons
- Visualizations:
    - heatmaps
    - for service
    - per server
    - per backend
    - system 'flame graph'
    - visualize traffic as graph, queue time, request flow

Message Queueing

- Distribution of message latency (ns)
- Throughput
- Total number of ns
- Errors, drop, retransmits, discards
- Message fanout distribution (gain: ratio of input to put)
- For distribution message queues: see distributied systems
- Queue lengths
- Saturation: run out of space
- Resource constraints on queueing systems
- Last time of access

Web servers

- Requests: referrer, origin, UA, resp code, count
    - origin
    - response code
- Req size: distribution
- Response Size: resp code, distribution
- Responce Count: resp code, counter

- Time To First Bite: resp code, distribution
- Time To Last Bite: resp code, distribution
- Active Workers: guage
- Worker Age: guage
- Connections: counter
- Process Metrics from host

Application servers

- Total requests served, rate
- Latency:
    - time to serve a client
    - complete a client transaction
    - request queue time
- App error rate
- Error counts on backend H/W
- Bandwidth usage front and backend
- System load on primary application server: CPU, memory, disk, swapping
- Usage patterns:
    - which user, client time, session time, active vs idle time

Databases

- Queries/sec
- # of connections
- connections/sec
- avg time per query
- cache hit rate
- avg io latency
- aggregate io
- % of query time in io
- # of locks
- # of versions (for read consistency)
- terminated connects
- SQL statements
- cache evictions
- query errors by type
- saturation: plan to execute
    - queueing on pool
- change in number of executed plans
- latency of last checkpoint, and on-disk representation of wall log
    - (how much of DB to reply)
- checkpoint times

Resources/Devices

- Utilization
    - per-device: eg, as a heat map for distribution over time
- Saturation
    - average queue length, or time waiting on queue
- Errors

Thanks to all those who attended and helped out!

You can comment here, but I can't guarantee your comment will remain here forever: I might switch comment systems at some point (eg, if disqus add advertisements).

About this blog