Appendix D - Example Postmortem

Example Postmortem

Shakespeare Sonnet++ Postmortem (incident #465)

Date: 2015-10-21

Authors: jennifer, martym, agoogler

Status: Complete, action items in progress

Summary: Shakespeare Search down for 66 minutes during period of very high interest in Shakespeare due to discovery of a new sonnet.

Impact:[163] Estimated 1.21B queries lost, no revenue impact.

Root Causes:[164] Cascading failure due to combination of exceptionally high load and a resource leak when searches failed due to terms not being in the Shakespeare corpus. The newly discovered sonnet used a word that had never before appeared in one of Shakespeare's works, which happened to be the term users searched for. Under normal circumstances, the rate of task failures due to resource leaks is low enough to be unnoticed.

Trigger: Latent bug triggered by sudden increase in traffic.

Resolution: Directed traffic to sacrificial cluster and added 10x capacity to mitigate cascading failure. Updated index deployed, resolving interaction with latent bug. Maintaining extra capacity until surge in public interest in new sonnet passes. Resource leak identified and fix deployed.

Detection: Borgmon detected high level of HTTP 500s and paged on-call.

Action Items:[165]

Action Item	Type	Owner	Bug
Update playbook with instructions for responding to cascading failure	mitigate	jennifer	n/a **DONE**
Use flux capacitor to balance load between clusters	prevent	martym	Bug 5554823 **TODO**
Schedule cascading failure test during next DiRT	process	docbrown	n/a **TODO**
Investigate running index MR/fusion continuously	prevent	jennifer	Bug 5554824 **TODO**
Plug file descriptor leak in search ranking subsystem	prevent	agoogler	Bug 5554825 **DONE**

Action Item	Type	Owner	Bug
Add load shedding capabilities to Shakespeare search	prevent	agoogler	Bug 5554826 **TODO**
Build regression tests to ensure servers respond sanely to queries of death	prevent	clarac	Bug 5554827 **TODO**
Deploy updated search ranking subsystem to prod	prevent	jennifer	n/a **DONE**
Freeze production until 2015-11-20 due to error budget exhaustion, or seek exception due to grotesque, unbelievable, bizarre, and unprecedented circumstances	other	docbrown	n/a **TODO**

Lessons Learned

What went well

- Monitoring quickly alerted us to high rate (reaching ~100%) of HTTP 500s
- Rapidly distributed updated Shakespeare corpus to all clusters

What went wrong

- We're out of practice in responding to cascading failure
- We exceeded our availability error budget (by several orders of magnitude) due to the exceptional surge of traffic that essentially all resulted in failures

Where we got lucky[166]

- Mailing list of Shakespeare aficionados had a copy of new sonnet available
- Server logs had stack traces pointing to file descriptor exhaustion as cause for crash
- Query-of-death was resolved by pushing new index containing popular search term

Timeline[167]

2015-10-21 *(all times UTC)*

- 14:51 News reports that a new Shakespearean sonnet has been discovered in a Delorean's glove compartment
- 14:53 Traffic to Shakespeare search increases by 88x after post to */r/shakespeare* points to Shakespeare search engine as place to find new sonnet (except we don't have the sonnet yet)
- 14:54 **OUTAGE BEGINS** — Search backends start melting down under load
- 14:55 docbrown receives pager storm, `ManyHttp500s` from all clusters
- 14:57 All traffic to Shakespeare search is failing: see *http://monitor*
- 14:58 docbrown starts investigating, finds backend crash rate very high
- 15:01 **INCIDENT BEGINS** docbrown declares incident #465 due to cascading failure,

coordination on `#shakespeare`, names jennifer incident commander
- 15:02 someone coincidentally sends email to *shakespeare-discuss@* re sonnet discovery, which happens to be at top of martym's inbox
- 15:03 jennifer notifies *shakespeare-incidents@* list of the incident
- 15:04 martym tracks down text of new sonnet and looks for documentation on corpus update
- 15:06 docbrown finds that crash symptoms identical across all tasks in all clusters, investigating cause based on application logs
- 15:07 martym finds documentation, starts prep work for corpus update
- 15:10 martym adds sonnet to Shakespeare's known works, starts indexing job
- 15:12 docbrown contacts clarac & agoogler (from Shakespeare dev team) to help with examining codebase for possible causes
- 15:18 clarac finds smoking gun in logs pointing to file descriptor exhaustion, confirms against code that leak exists if term not in corpus is searched for
- 15:20 martym's index MapReduce job completes
- 15:21 jennifer and docbrown decide to increase instance count enough to drop load on instances that they're able to do appreciable work before dying and being restarted
- 15:23 docbrown load balances all traffic to USA-2 cluster, permitting instance count increase in other clusters without servers failing immediately
- 15:25 martym starts replicating new index to all clusters
- 15:28 docbrown starts 2x instance count increase
- 15:32 jennifer changes load balancing to increase traffic to nonsacrificial clusters
- 15:33 tasks in nonsacrificial clusters start failing, same symptoms as before
- 15:34 found order-of-magnitude error in whiteboard calculations for instance count increase
- 15:36 jennifer reverts load balancing to resacrifice USA-2 cluster in preparation for additional global 5x instance count increase (to a total of 10x initial capacity)
- 15:36 **OUTAGE MITIGATED**, updated index replicated to all clusters
- 15:39 docbrown starts second wave of instance count increase to 10x initial capacity
- 15:41 jennifer reinstates load balancing across all clusters for 1% of traffic
- 15:43 nonsacrificial clusters' HTTP 500 rates at nominal rates, task failures intermittent at low levels
- 15:45 jennifer balances 10% of traffic across nonsacrificial clusters
- 15:47 nonsacrificial clusters' HTTP 500 rates remain within SLO, no task failures observed
- 15:50 30% of traffic balanced across nonsacrificial clusters
- 15:55 50% of traffic balanced across nonsacrificial clusters
- 16:00 **OUTAGE ENDS**, all traffic balanced across all clusters
- 16:30 **INCIDENT ENDS**, reached exit criterion of 30 minutes' nominal performance

Supporting information:[168]

- Monitoring dashboard, *http://monitor/shakespeare?end_time=20151021T160000&duration=7200*

[163]Impact is the effect on users, revenue, etc.

[164]An explanation of the circumstances in which this incident happened. It's often helpful to use a technique such as the 5 Whys [Ohn88] to understand the contributing factors.

[165]"Knee-jerk" AIs often turn out to be too extreme or costly to implement, and judgment may be needed to re-scope them in a larger context. There's a risk of over-optimizing for a particular issue, such as adding specific monitoring/alerting when reliable mechanisms like unit tests can catch problems much earlier in the development process.

[166]This section is really for near misses, e.g., "The goat teleporter was available for emergency use with

other animals despite lack of certification.”

[167]A “screenplay” of the incident; use the incident timeline from the Incident Management document to start filling in the postmortem’s timeline, then supplement with other relevant entries.

[168]Useful information, links, logs, screenshots, graphs, IRC logs, IM logs, etc.

previous

Appendix C - Example Incident State Document

next

Appendix E - Launch Coordination Checklist