ACM.org Highlights

For Scientists, Programmers, Designers, and Managers:

- **Learning Center** - [https://learning.acm.org](https://learning.acm.org)
  - View past TechTalks & Podcasts with top inventors, innovators, entrepreneurs, & award winners
  - Access to O’Reilly Learning Platform – technical books, courses, videos, tutorials & case studies
  - Access to Skillsoft Training & ScienceDirect – vendor certification prep, technical books & courses
- **Ethical Responsibility** – [https://ethics.acm.org](https://ethics.acm.org)

**By the Numbers**

- 2,200,000+ content readers
- 1,800,000+ DL research citations
- $1,000,000 Turing Award prize
- 100,000+ global members
- 1160+ Fellows
- 700+ chapters globally
- 170+ yearly conferences globally
- 100+ yearly awards
- 70+ Turing Award Laureates

**Popular Publications & Research Papers**

- Communications of the ACM - [http://cacm.acm.org](http://cacm.acm.org)
- Queue Magazine - [http://queue.acm.org](http://queue.acm.org)
- Digital Library - [http://dl.acm.org](http://dl.acm.org)

**Major Conferences, Events, & Recognition**

- [https://www.acm.org/conferences](https://www.acm.org/conferences)
- [https://www.acm.org/chapters](https://www.acm.org/chapters)
- [https://awards.acm.org](https://awards.acm.org)
OOPS!
Learning from surprise at Netflix

Lorin Hochstein
Sr. Software Engineer, Netflix
Let’s talk about outages!
At Netflix, we call them *incidents*
Incidents are scary!
The system did something we didn’t expect...
...and a bad thing happened!
Uncertainty makes people nervous
We want closure
How can we be confident this won’t happen again?
We do an incident review
Why did this happen?
Do a root cause analysis
Identify action items that will prevent reoccurrence
We can now move past it
Until the next one...
… which is completely different
We can get more out of incidents than preventing the last one
Incidents are unplanned investments; their costs have already been incurred. Your org’s challenge is to get ROI on those events. Right now, in most companies, this ROI is left sitting in the dark because of the “template-driven” approaches and “action item” myopia.
Learning isn’t proportional to impact of an incident
We can learn just as much from “incidents” where there is no business impact!
An operational surprise
OOPSies
OOPS
New Spinnaker resilience feature caused too much pressure on Titus
feat(dynamicBackoff): Dynamically configurable task backoff
#3228

marchello2000 merged 2 commits into spinaker/master from jonsie:dynamic-cloudprovider-retry 20 days ago

@lhochstein

---

jonsie commented 20 days ago - edited -

Task backoff can be configured globally, or drilled down to the cloudProvider and account level. The relevant max backoff time always wins - if you desire no change of behavior simply do not create a dynamic config entry for a back off. Note that the max back off time of 2 minutes can not be overridden.

---

jonsie reviewed 20 days ago

orca-queue/src/test/kotlin/com/netflix/spinaker/orca/q/Handler/RunTaskHandlerTest.kt

```kotlin
102  108    clock,
103  109    listOf(exceptionHandler),
104 - emptyList(),
105 - NoopRegistry()
```

jonsie 20 days ago

I removed the emptyList(), so I could test the max back off set in the taskExecutionInterceptors. This meant I had to update a fair amount of these tests.
Bump Jedis to version that fixes block/wait issue #364

robfletcher commented on Apr 1

No description provided.

robfletcher commented on Apr 1

Dependencies in fiat-web post this change:

redis.clients:jedis:2.9.3
  variant "default-runtime" {  
    org.gradle.status = release (not requested)  
    Requested attributes not found in the selected variant:  
    org.gradle.usage = java-api  
  }  
  Selection reasons:  
  - Excluded  
  - By conflict resolution : between versions 2.9.3 and 2.9.0

redis.clients:jedis:2.9.3
  \--- project :fiat-roles
  \--- compileClasspath

redis.clients:jedis:2.9.0 \--- com.netflix.spinnaker.kork.kork-dynomite:3.11.0
  \--- project :fiat-roles
  \--- compileClasspath

 exchanges:redis: bump Jedis to version that fixes block/wait issue #364

robfletcher requested review from dreynaud and asher on Apr 1

robfletcher committed on Apr 1

dreynaud merged commit 737cc74 into spinmaker/master on Apr 1
Connection leaking in 2.9.1 GenericObjectPool #1945

@jebeaudet commented on Feb 6

Follow up of #1929 since the author closed the issue. I would really appreciate a backport like mentioned in the linked issue. Also, fyi, jedis 2.9.1 is the "official" version provided by Spring Boot 1.5.19.RELEASE as the dependencyManagement so this probably affects a lot of users. The 1.5.x branch is not in active development anymore but they are still supporting it through end of summer 2019 IRC.

Expected behavior
No connection leak

Actual behavior
There's a connection leak with heavy usage

Steps to reproduce:
With a heavy usage of jedis 2.9.1, the connections are slowly getting leaked and the pool starts rejecting connections after a while.

Redis / Jedis Configuration
Nothing special

Jedis version:
2.9.1

Redis version:
N/A

Java version:
Java 8

Thanks!!
Short personal post-mosterm of how a small code change I made in Spinnaker caused Netflix to run with ~10k extra AWS instances overnight.

5:30 PM · Mar 15, 2019 · Twitter Web Client
fix(clouddriver): Revert change to pin source capacity for redblack deploys #2756

ajordens commented on Mar 14

No description provided.

ajordens added 2 commits on Mar 14

- fix(clouddriver): Revert change to pin source capacity for redblack d..

  33a50b97

- Merge branch 'master' into master

  d776dbf

ajordens commented on Mar 14

FYI @dreynaudo

Turns out it's caused a couple problems internally.

cfieber approved these changes on Mar 14

ajordens merged commit 3096eaa into spinaker:master on Mar 14

spinnakerbot added the target-release/1.13 label on Mar 14

@lhochstein
A play in three acts

1. What we hope to learn from OOPSies
2. What to ask when looking into how an OOPS happened
3. How to write up an OOPS
I. What we hope to learn
Fools learn from experience. I prefer to learn from the experience of others.
– Otto von Bismarck (attributed)
Identify gaps
Tooling gaps
example-v405

Sets up autoscaling for this server group.

To disable autoscaling, use the Simple Mode.

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Desired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>12</td>
<td>5000</td>
<td>49</td>
</tr>
<tr>
<td>Resize to</td>
<td>12</td>
<td>5000</td>
<td>49</td>
</tr>
</tbody>
</table>

**Desired** capacity is managed by Autoscaling Policies.
If you need to scale **down** this server group, set **Max** to the new desired size.
If you need to scale **up** this server group, set **Min** to the new desired size.

- Enforce Capacity Constraints

Reason
(Optional) anything that might be helpful to explain the reason for this change; HTML is okay

Changes
No changes

Type the name of the account (**prod**) to continue.
Sets up autoscaling for this server group.

To disable autoscaling, use the Simple Mode.

<table>
<thead>
<tr>
<th></th>
<th>Current</th>
<th></th>
<th>Desired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>12</td>
<td>5000</td>
<td>49</td>
</tr>
<tr>
<td>Max</td>
<td>5000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Desired** capacity is managed by Autoscaling Policies.

If you need to scale **down** this server group, set **Max** to the new desired size.

If you need to scale **up** this server group, set **Min** to the new desired size.

- **Enforce Capacity Constraints**
- **Reason** *(Optional) anything that might be helpful to explain the reason for this change; HTML is okay*
- **Changes** No changes

Type the name of the account *(prod)* to continue
Consider a cluster of servers
The size is configurable
Netflix traffic varies over time
Autoscaling sizes for you

Desired: 128

Min: 20
Max: 1000

Metrics

Server group

EC2

Desired: 128

Autoscaler
One day...
Min: 12
Max: 1000
Desired: 128
One day...

Desired: 256

Autoscaler

Min: 20
Max: 1000

Metrics

Server group

EC2

Desired: 256
1. EC2: Bring up new instances

Desired

256
2. Autoscaler fires: 256 → 128
2. EC2: terminate instances

\[128\] Desired
User sees green → gray

**Autoscaler**

Min: 20
Max: 1000

**Metrics**

**Server group**

**EC2**

Desired: 128
example-v405

Sets up autoscaling for this server group.

To disable autoscaling, use the Simple Mode.

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Desired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>12</td>
<td>5000</td>
<td>49</td>
</tr>
<tr>
<td>Resize to</td>
<td>12</td>
<td>5000</td>
<td>49</td>
</tr>
</tbody>
</table>

**Desired** capacity is managed by Autoscaling Policies.
If you need to scale **down** this server group, set **Max** to the new desired size.
If you need to scale **up** this server group, set **Min** to the new desired size.

- Enforce Capacity Constraints
- Reason: (Optional) anything that might be helpful to explain the reason for this change; HTML is okay
- Changes: No changes

Type the name of the account (prod) to continue
Operational expertise gaps
Resource gaps
Beware the *law of stretched systems*!
Every system is stretched to operate at its capacity.
Beware the law of fluency!
Hard to tell when a skilled engineer starts to become overloaded
Build shared understanding
It came as a surprise that X calls Y’s endpoint
Facilitate skill transfer
Learn by watching experts in action
II. What to ask
Do an investigation afterwards
(but don’t call it that)
“How did we get here?”
How did $X$ seem reasonable in the moment?
What were all of the things that had to be true for the surprise to happen?
Capture perspectives from multiple people
III. How to write it up
Narrative description
Tell a good story
Imagine new team member reading it
Contributing factors
Front50 provides an inconsistent view of application permissions, this triggered endless retries
Similar feature was already in use, so enabling it here seemed low-risk
X was out sick when the feature was deployed
Mitigators
Spinnaker's staging stack was not impacted, which gave us a backdoor way to monitor and make changes
Demand Engineering has tooling & experience in changing size of many server groups automatically, which was sufficient to undo most bad changes
Risks
The regression occurred in an area of Spinnaker that is difficult to test
Misconfigured pools and queues
Difficulties in handling
Observability blind spots: lack of metrics around connection pool or redis command usage made it difficult to determine redis usage change
feat(redis): Instrumented Jedis client & Pool #144

robziernert commented on Apr 10, 2018

Instruments:
- The latency for every command
- The payload size for (most) writes
- The error rate for individual commands
- The combined error rate for the client

robziernert force-pushed the robziernert:instrumented-jedis-client branch 5 times, most recently from 9682e3a to fe7af18 on Apr 10, 2018

asher reviewed on Apr 10, 2018

kork-jedis/src/main/java/com/netflix/spinnaker/kork/jedis/telemetry/InstrumentedJedi
s.java

62 + String command = "set";
66 + registry.distributionSummary(TelemetryHelper.payloadSizeIdIdPrefix
67 + try {
68 + return registry.timer(timerIdIdPrefix, command).record() ->

asher on Apr 10, 2018  Contributor

PercectileTimer (http://netflix.github.io/spectator/en/latest/javadoc/spec	ator-
apicom/netflix/spectator/ap
histogram(PercentileTimer.html) please, it enables things
name,redis.command.set,eq{(5,95,99),percentile in atlas

robziernert on Apr 10, 2018  Author  Member

I don't believe the PercentileTimer is actually exposed through the API? All I see on
our registry is Timer. I was under the impression that timers would be automatically
percentiled
clouddriver was rolled out at 3pm and we were paged at 5:30pm, so not immediately clear that issue had to do with deployment
If you only remember three things...

• Any operational surprise is a potential opportunity for learning
• Ask questions that answer “how did we get here?”
• Tell a good story
I want to learn more about learning more!

• Etsy Debrief Facilitation Guide
• The Field Guide To Understanding ‘Human Error’ by Sidney Dekker
• http://resiliencepapers.club
The Learning Continues...

TechTalk Discourse: https://on.acm.org
TechTalk Inquiries: learning@acm.org
TechTalk Archives: https://learning.acm.org/techtalks
Learning Center: https://learning.acm.org
Professional Ethics: https://ethics.acm.org
Queue Magazine: https://queue.acm.org