Availability Objectives of SoundCloud’s Microservices

Bora Tunca
SoundCloud is the largest online audio distribution platform
Bora Tunca
@grandbora

- Backend developer
- @SoundCloud ~3 years
- Core engineering
Microservices
~140 services
~140 engineers
How do we improve availability in a microservices architecture?
@ShorneMarkley @SoundCloud Seriously WTF. Broke 700+ descriptions...

why the fuck is soundcloud broken

Week 3: Either my music really sucks ass or soundcloud’s stats are still broken as fuck.

Kanye broke Soundcloud wtf
Post Mortem Meetings

help us spot “risky” services
Service Level Objectives

deﬁne an availability objective for each microservice
If the availability of a service drops below its objective, we raise the red flag.
Service Level Objectives

- SLO infrastructure
- Usage analyses
SLO Infrastructure
Availability

- Error Rate Threshold
- Latency Threshold

99th percentile
SLO Parameters

SLO_ERROR_RATE_THRESHOLD=0.0001
SLO_LATENCY_P99_THRESHOLD SECONDS=0.500
SLO_AVAILABILITY_TARGET=0.99999
SLO Metrics

- TotalTransactions
- FailedTransactions
- SlowTransactions
Availability

\[
\frac{\text{FailedTransactions}}{\text{TotalTransactions}} < \frac{\text{error rate threshold}}{0.01} \quad \& \&
\]

\[
\frac{\text{SlowTransactions}}{\text{TotalTransactions}} < 0.01
\]
Prometheus

Services Directory

Metrics

Availability

SLO Wrapper

Req

Res
# SoundCloud - Services

Last updated at 2016-6-27 21:25.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Owner</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>widget</td>
<td></td>
<td>web</td>
<td>99.937%</td>
</tr>
<tr>
<td>developers</td>
<td></td>
<td>web</td>
<td>100%</td>
</tr>
<tr>
<td>sc-rollout</td>
<td></td>
<td>core-services</td>
<td>100%</td>
</tr>
<tr>
<td>v2</td>
<td></td>
<td>web</td>
<td>99.920%</td>
</tr>
<tr>
<td>notifications</td>
<td></td>
<td>user-retention</td>
<td>87.913%</td>
</tr>
<tr>
<td>search-suggest</td>
<td></td>
<td>search</td>
<td>99.152%</td>
</tr>
<tr>
<td>payments-reports</td>
<td></td>
<td>payments</td>
<td>99.208%</td>
</tr>
<tr>
<td>discovery-ranking</td>
<td></td>
<td>discovery</td>
<td>99.940%</td>
</tr>
</tbody>
</table>
Usage Analyses
Graceful Degradation

API

Tracks

Stats

99.995%

95%

luara
die antwoord „Ugly Boy“

Gnarls Barkley
Crazy

Coolio
Gangsta's Paradise

❤ 292 likes

View all
Learnings
● Without a formal process identifying risky services is hard
● Don’t over-engineer your services
● Let the data guide your engineering decisions
THANK YOU

bora@soundcloud.com