Why Does (My) Monitoring Suck?

Todd Palino
Senior Staff Engineer, Site Reliability
LinkedIn
This Is The Only Slide You May Need a Picture Of

https://slideshare.net/ToddPalino
**What’s On Our List Today?**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alerting Anti-Patterns</td>
<td></td>
</tr>
<tr>
<td>Setting Goals</td>
<td></td>
</tr>
<tr>
<td>What Is Monitoring?</td>
<td></td>
</tr>
<tr>
<td>Designing For Success</td>
<td></td>
</tr>
<tr>
<td>Wrapping Up</td>
<td></td>
</tr>
</tbody>
</table>
Alerting Anti-Patterns
Network Operations Center

• Central monitoring and alerting
• Gatekeeping monitored alerts with no deep knowledge
• Information overload for a moderate sized system
• Glorified telephone operators
Kafka Under-Replicated Partitions

- Unclear meaning
- Sometimes it’s not a problem at all
- Does the customer care as long as requests are getting served?
- Frequently gets ignored in the middle of the night
CPU Load

- Relative measure of how busy the processors are
- Who cares? Processors are supposed to be busy
- What’s causing it?
- Might be capacity. Maybe
Setting Goals
Service Level
Whatever

- SLI: Indicator
- SLO: Objective
- SLT: Target
- SLA: Agreement
Let’s Be Smart About This

- Specific
- Measurable
- Agreed
- Realistic
- Time-limited, Testable
Common SLOs

**Availability**
Is the service able to handle requests?

**Latency**
Are requests being handled promptly?

**Correctness**
Are the responses being returned correct?
What Is Monitoring?
Observe and check the progress or quality of (something) over a period of time; keep under systematic review.
So WTF is Observability?

- Comes from control theory
- A measure of how well internal states of a system can be inferred from knowledge of its external outputs
- It’s a noun – you have this (to some extent). You can’t “do” it.
What Are We Looking For?

Rumsfeld Quadrant

<table>
<thead>
<tr>
<th>Detection</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>Known</td>
</tr>
<tr>
<td>Known</td>
<td>Unknown</td>
</tr>
<tr>
<td>Unknown</td>
<td>Known</td>
</tr>
<tr>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>
What Can We Work With?

**Metrics**
- Single numbers
  - Counters
  - Gauges
  - Histograms (and Summaries)

**Events**
- Structured data
  - Log messages
  - Tracing (collection of events)
Where Can We Get It?

Subjective

• Rich data on internal state
• Necessary for high observability
• Tons of data possible, but the utility is often questionable
• Beware! Here be dragons!

Objective

• Customer view of your system
• Think of “Down For Everyone Or Just Me?”
• Critical for SLO monitoring
• More difficult to do, but it’s the authority on whether or not something is broken
Build For Failure

**Intelligence**
Rich instrumentation on every aspect

**Availability**
Tolerate single component failures (not just N+1)

**Capacity**
Limit resource creation and utilization
Using the SLO

It’s the only thing that matters

- Always measure the SLIs
- Objective monitoring is best
- Don’t beat the SLO
- Only alert on the SLO
ONLY???

- SLO alerts find unknown-unknowns
- Known-unknowns and unknown-knowns must only exist transiently
- A known-known should not require a human. Automate responses to known issues
- For all else, if you have a 100% signal it can be an alert. But if it doesn’t impact the SLO, does it need to wake you up?
What About Capacity?

Use Quotas
Assure no single user can quickly overrun capacity

Report & Review
Frequently enough to respond to trend changes

Act Promptly
Never ignore or put off expansion work
Wrapping Up
What Should I Do Next?

Define Your SLOs
- Talk to your customers and agree on what they can expect
- Add objective monitoring for these expectations

Clean Up Alerts
- Inventory alerts and eliminate any that are not a clear signal
- Add alerts for the SLOs that you have agreed on
- Implement quotas, if needed, to assure capacity isn’t suddenly overrun

Add Instrumentation
- Switch to structured logging
- For distributed systems, consider adding request tracing
- But make sure you don’t hold this extra information for longer than it’s needed for debugging
More Resources

- **Code Yellow** – How we help overburdened teams
  - devops.com/code-yellow-when-operations-isnt-perfect
  - Usenix LISA 2018 “Code Yellow: Helping Top-Heavy Teams the Smart Way”
- **SRE** – What does the culture look like at LinkedIn
  - “Building SRE” usenix.org/conference/srecon18asia/presentation/palino
  - Every Day is Monday in Operations - everydayismondayinoperations.com
- **Kafka** – Deep dive on monitoring for Apache Kafka
  - confluent.io/kafka-summit-london18/urp-excuse-you-the-three-metrics-you-have-to-know

- **Finding Me**
  - linkedin.com/in/toddpalino
  - slideshare.net/ToddPalino
  - @bonkoif
Questions?