Better Living Through Statistics

Monitoring Doesn't (Have To) Suck

Jamie Wilkinson
Site Reliability Engineer, Google
jaq@{spacepants.org,google.com}
@jaqpants

#monitoringsucks

I love monitoring. This hashtag makes me sad.

But what to talk about?

I have no idea what has gone on in the real world since going into the black hole.

These #monitoringsucks people seem to be on the right path... do I have anything new to add?

Validation

http://blog.lusis.org/blog/2012/06/05/monitoring-sucking-just-a-little-bit-less/

"Instead of alerting on data and then storing it as an afterthought (perfdata anyone?) let's start collecting the data, storing it and then alerting based on it."

Monitoring systems

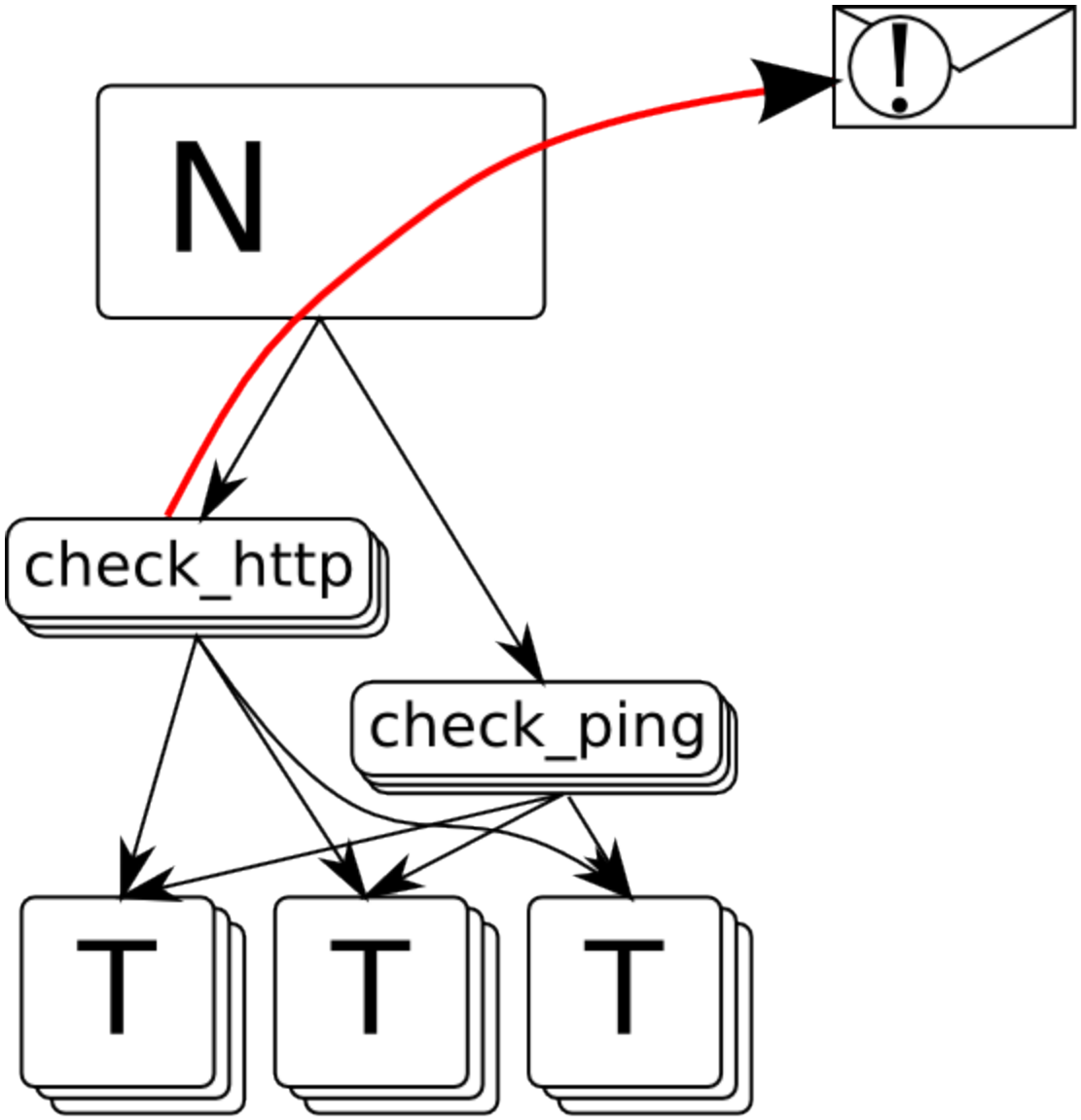
automate boring parts of experimental method

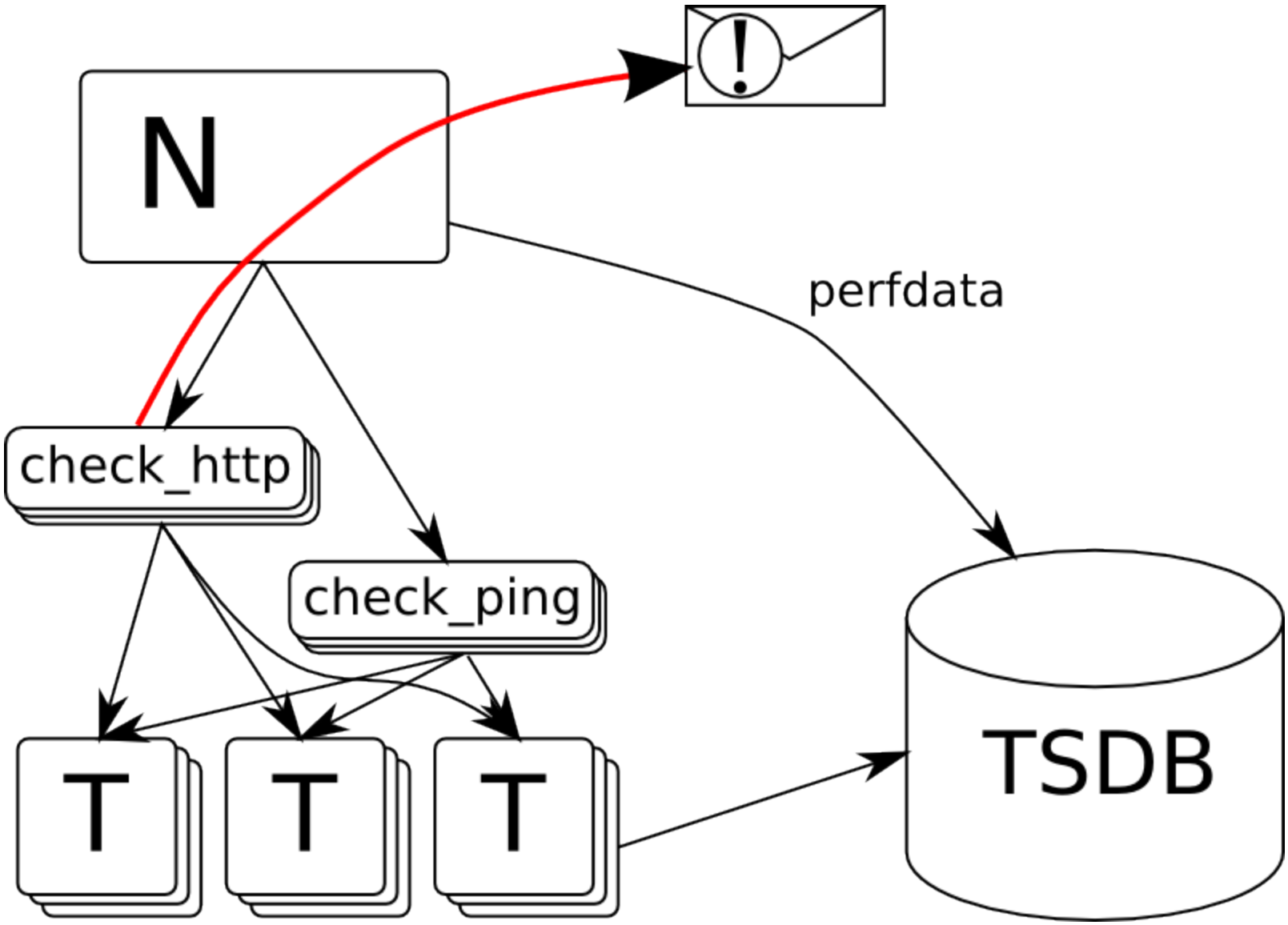
- measuring,
- recording,
- alerting,
- visualisation

so you have more time to do fun things... and debug the occasional emergency.

The current state of monitoring

1. blackbox monitoring resulting in alerts
2. whitebox monitoring resulting in charts


N
!
check_http
check_ping
T
T
T


N
!
perfdata
check_http
check_ping
T
T
T
TSDB

Blackbox vs Whitebox

Blackbox: treat the system as opaque: you can only use it as a user would.

a.k.a. "**probing**"

c.f. cucumber-nagios

fuel indicator light

Blackbox vs Whitebox

Whitebox: expose the internal state of the system for inspection

a.k.a. **instrumentation**, **telemetry,**
... ROCKET SCIENCE

c.f. ... graphite? new relic? metrics?
tacho, fuel gauge, water meter

What's wrong with blackbox?

Only boolean: no visibility into why

- Why is the site slow?
- Why has image serving stopped working?

No predictive capability

- How long until we need more disks? cpus? datacenters?

Problems with "check+alert" model

- Thresholds vary among instances, tuning difficult.
- Adding new targets, new checks is lots of effort.
- Check script logic performs the measurement and the "judgement" all in one.
- Alerts for things you can't act on.
- Duplicates
- Physical resource limits.

So, Why *does* #monitoringsuck?

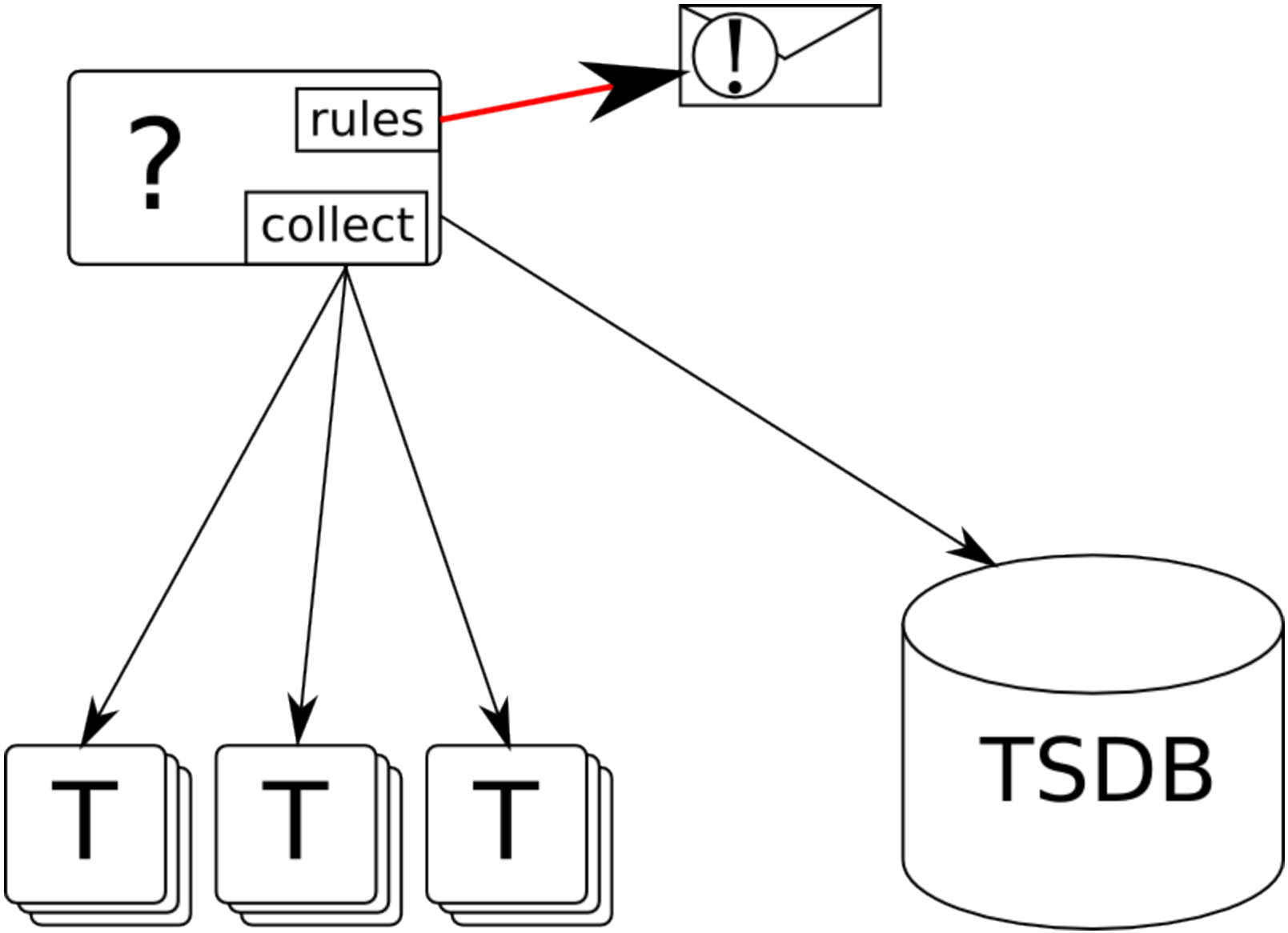
TL;DR:

when the cost of maintenance is too high to improve the quality of alerts

An idea...

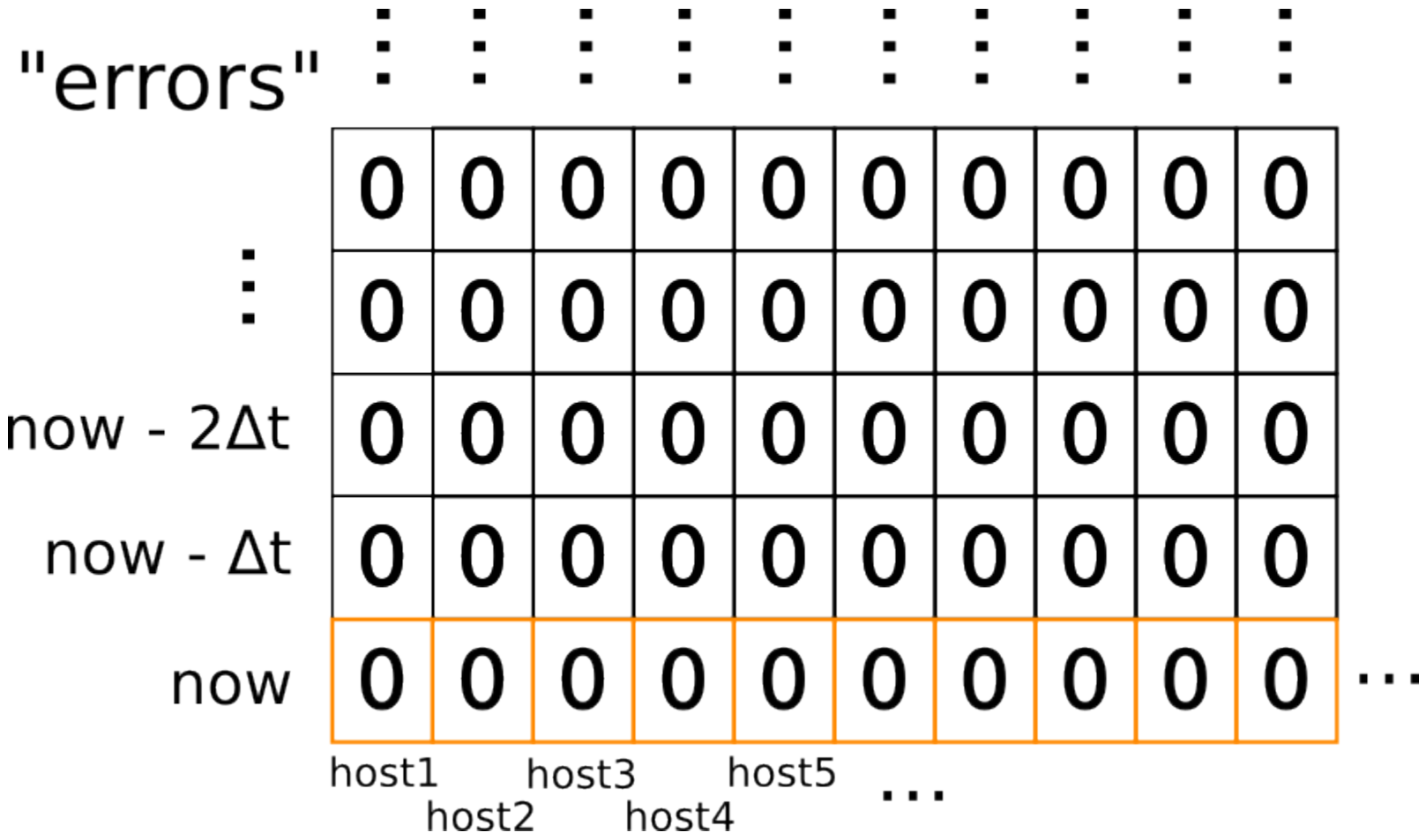
http://blog.lusis.org/blog/2012/06/05/monitoring-sucking-just-a-little-bit-less/

"Instead of alerting on data and then storing it as an afterthought (perfdata anyone?) let's start collecting the data, storing it and then **alerting based on it**."


?
rules
collect
!
T
T
T
TSDB


WAT

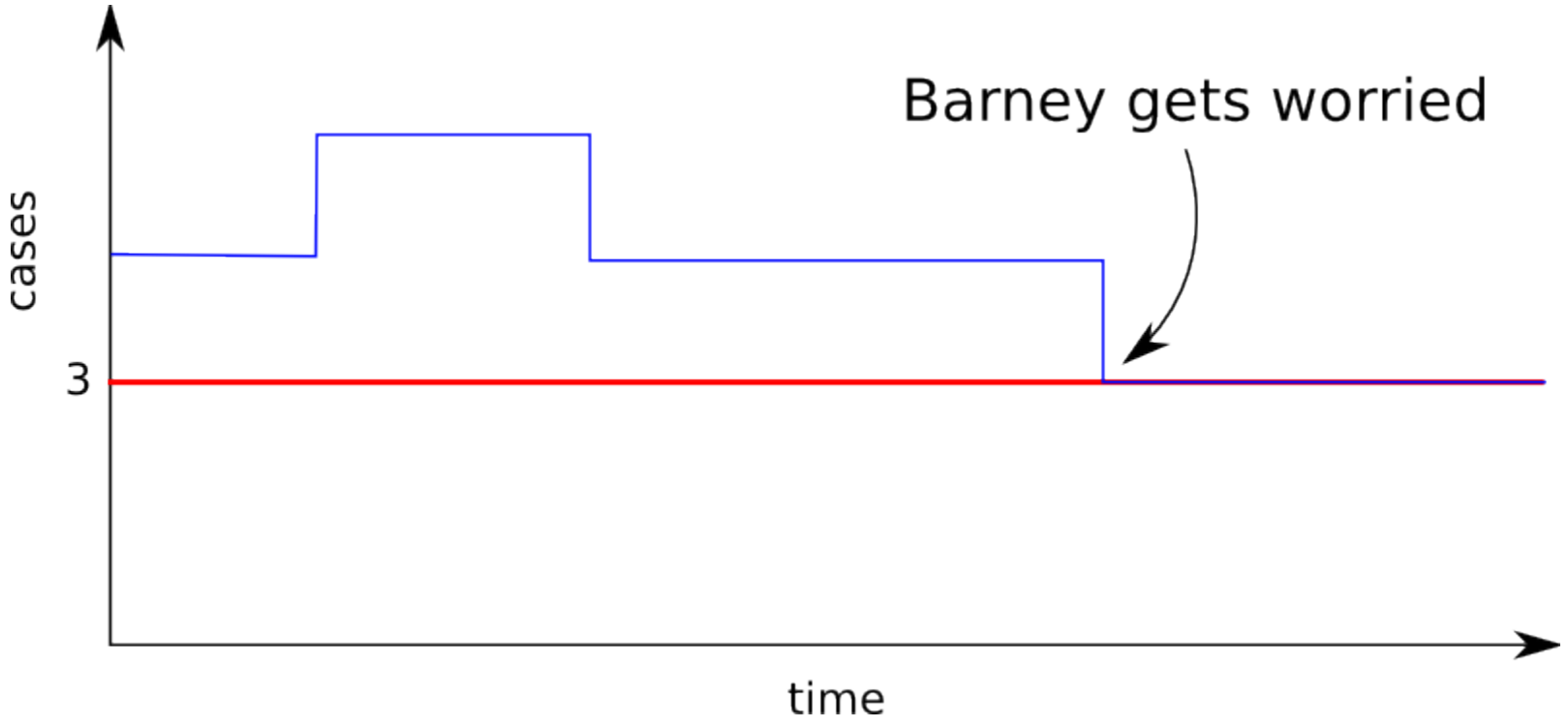
Structure of timeseries



Alerting on thresholds

Alert when beer supply low

```
if cases - 1 - 1 <= 1:
  alert BarneyWorriedAboutBeerSupply
```



Disk full alert

Alert when 90% full

Different filesystems have different sizes

10% of 2TB is 200GB

False positive!

Alert on absolute space, < 500MB

Arbitrary number

Different workloads with different needs: 500MB might not be enough warning

Disk full alert

More general alert:

How long before the disk is full?

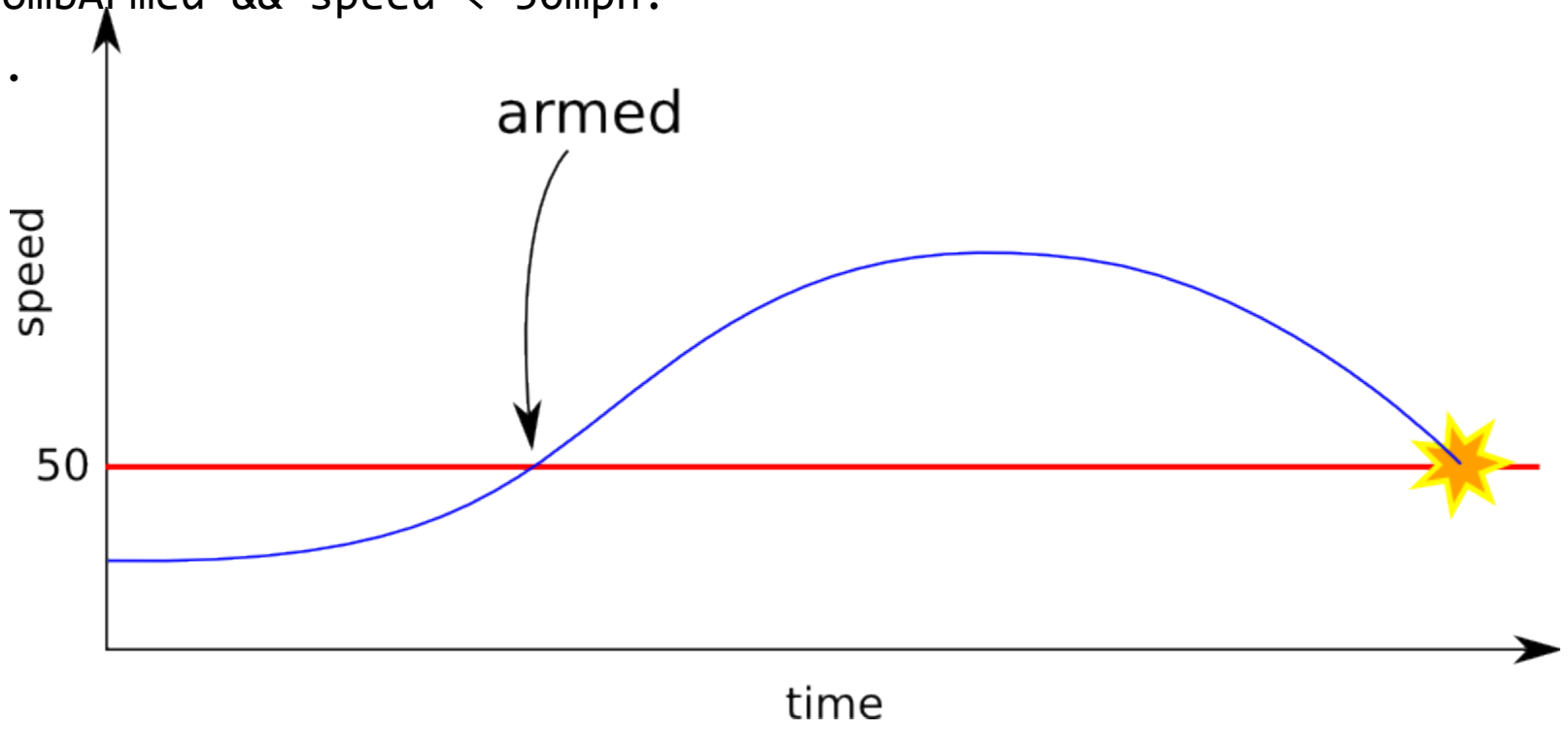
and

How long will it take for a human to fix an (almost) full disk?

Alerting on rates of change

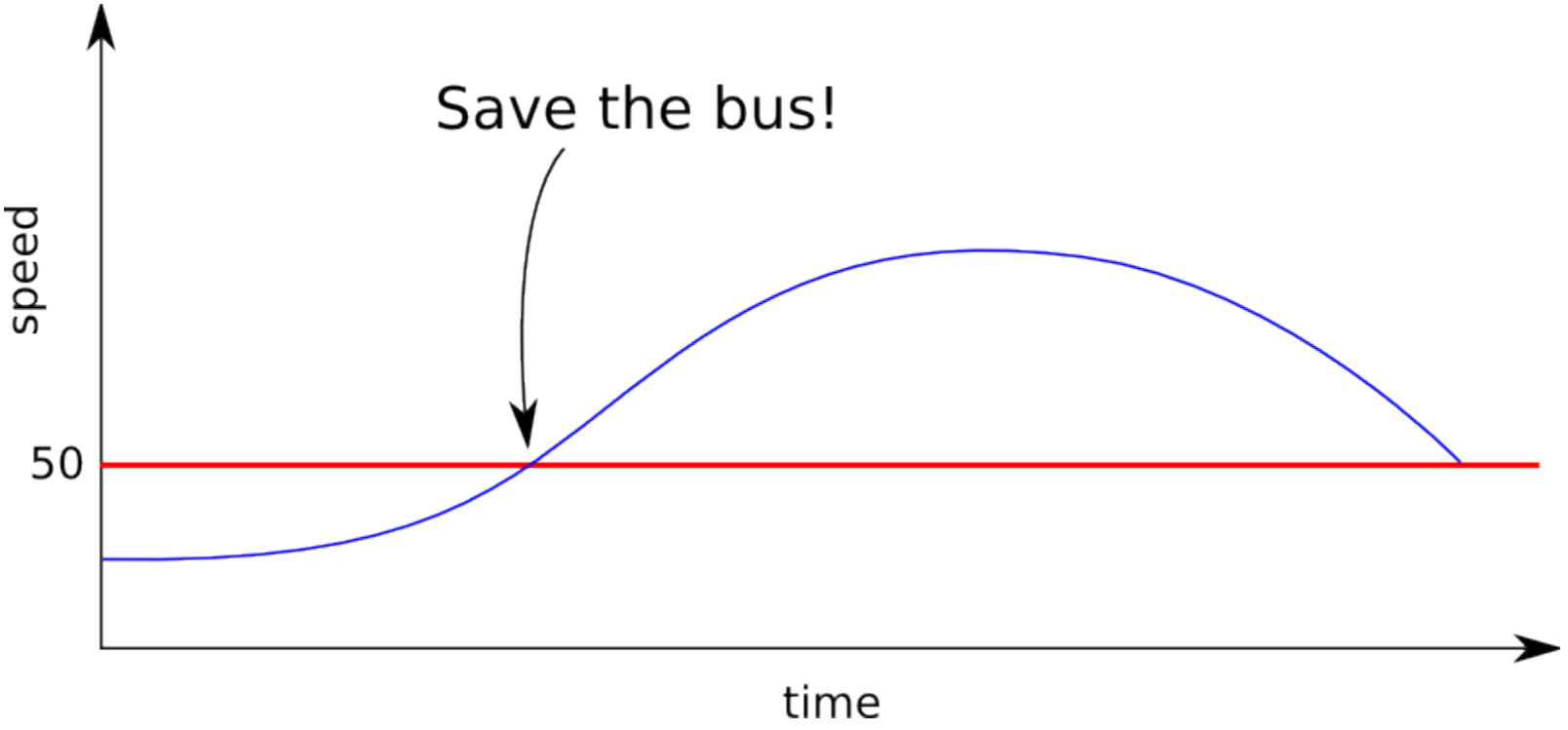
Dennis Hopper's Alert

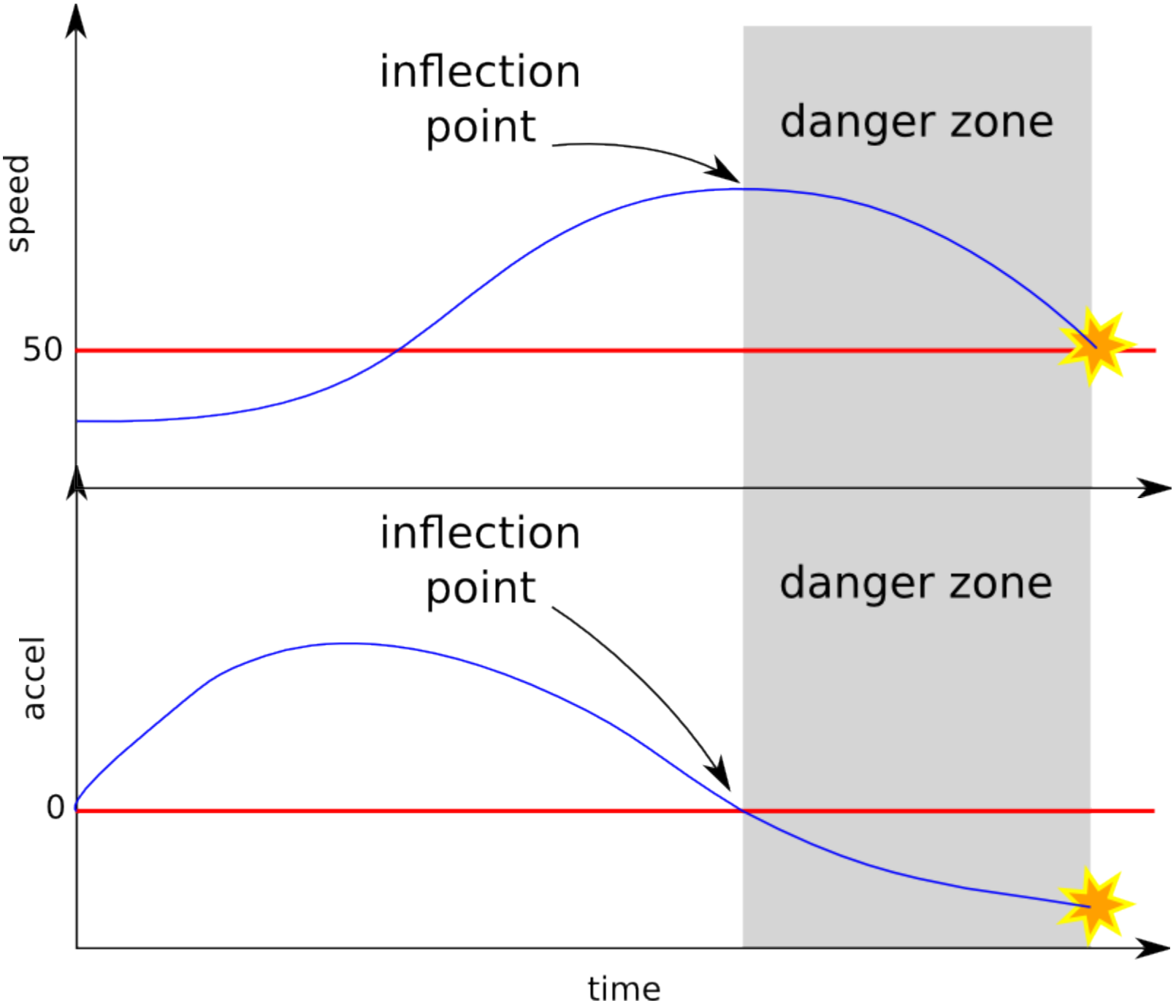
```
if speed >= 50mph:
  alert BombArmed
if BombArmed && speed < 50mph:
  ...
```



Keanu's Alert

```
if speed >= 50mph:
 alert SaveTheBus!
```




inflection
point
danger zone
speed
50
inflection
point
danger zone
accel
0
time

Keanu's alert

$$v - a * t = 50$$

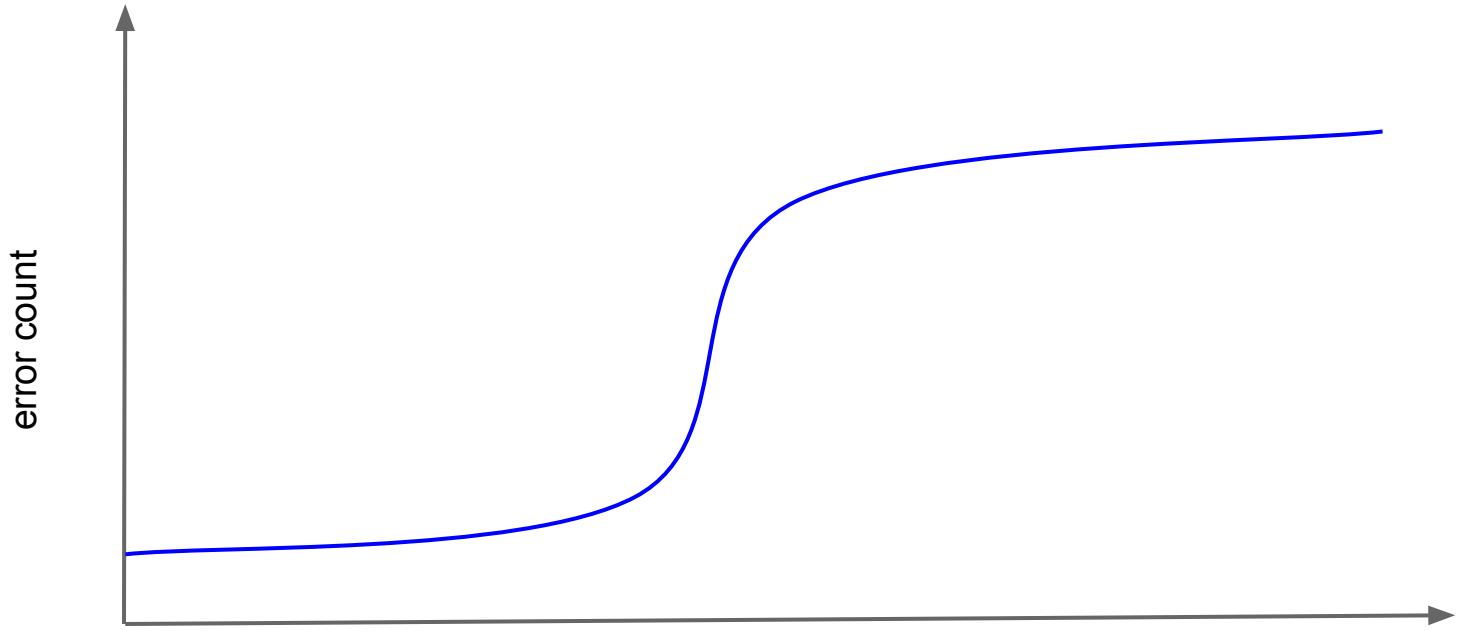
$$50 - v = - a * t$$

$$(v - 50)/a = t$$

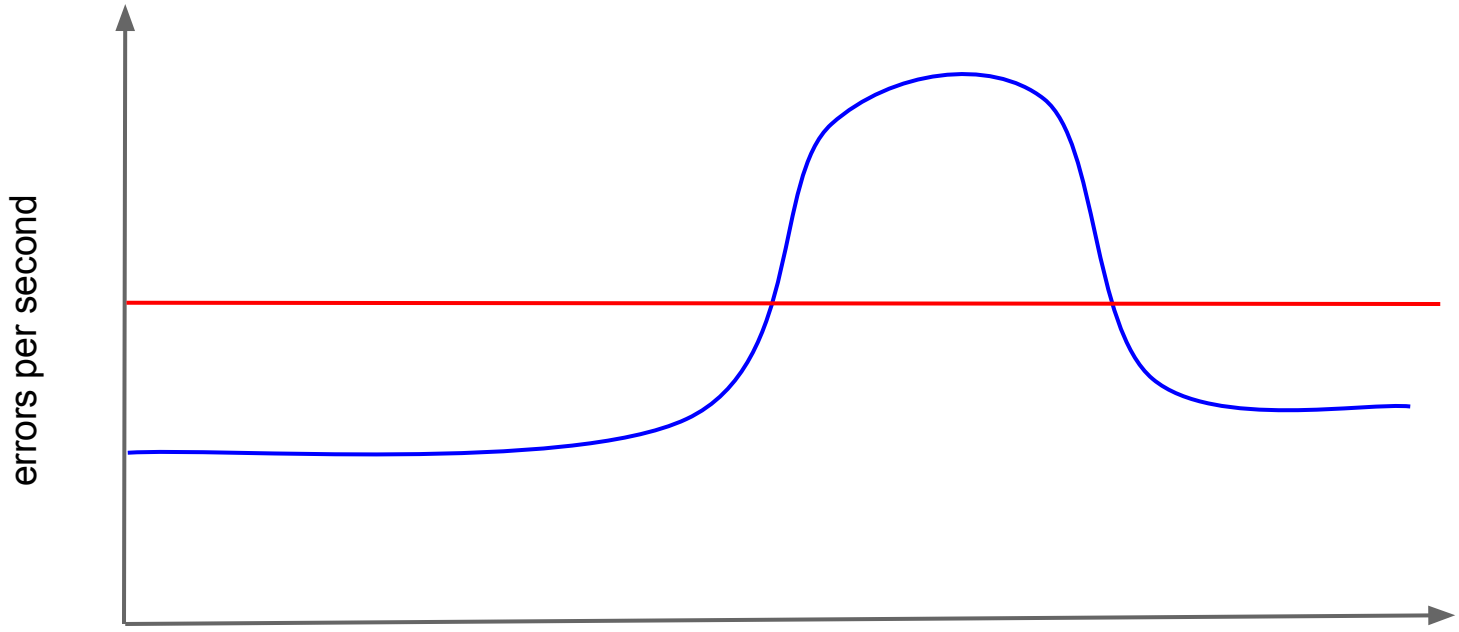
```
if (v - 50)/a <= time to save bus:
   alert StartSavingTheBus!
```

#1 calculate rates of change of timeseries

Example: Error spike



Rate of errors vs nominal rate



rate of change increases greater than expected

Rates and Derivatives

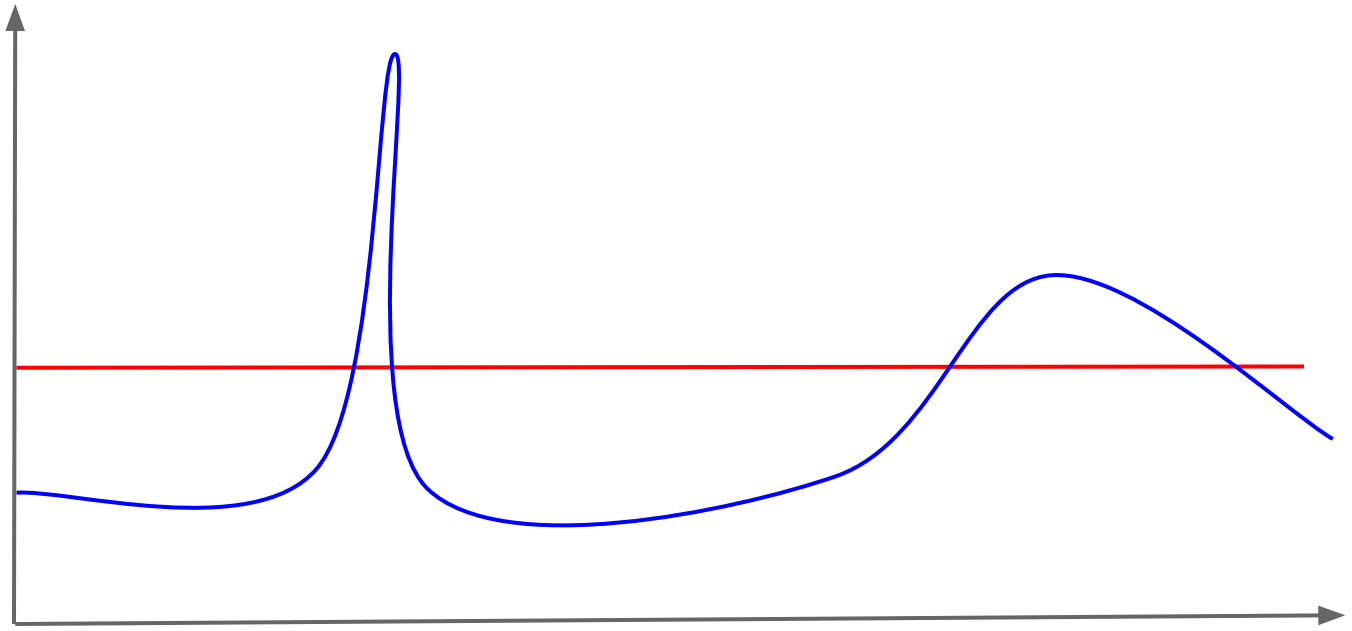
In a discreet timeseries:

$$\delta x/\delta t = (x_t - x_{t-1})/\Delta t$$

(assuming no missing samples)

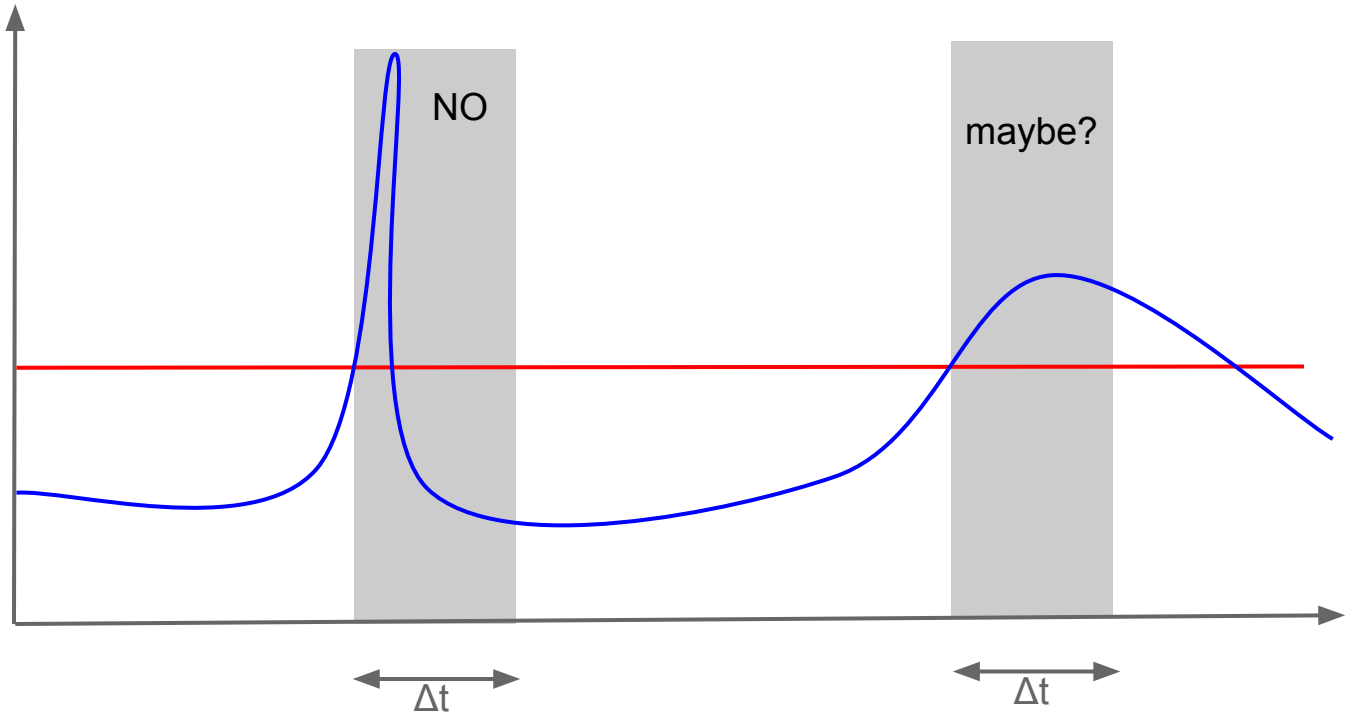
#2 observe timeseries history to gain context

Example: Traffic spike



worth getting out of bed for?

Duration of traffic spike



worth getting out of bed for?

Duration of abnormality

Not just looking at the latest data point, or the derivative at the latest point. (Otherwise we have just reinvented check scripts.)

Look back 5 minutes, 1 hour, 7 days, back to the dawn of time.

Heuristic: 2.5x the sampling interval to work around missing values

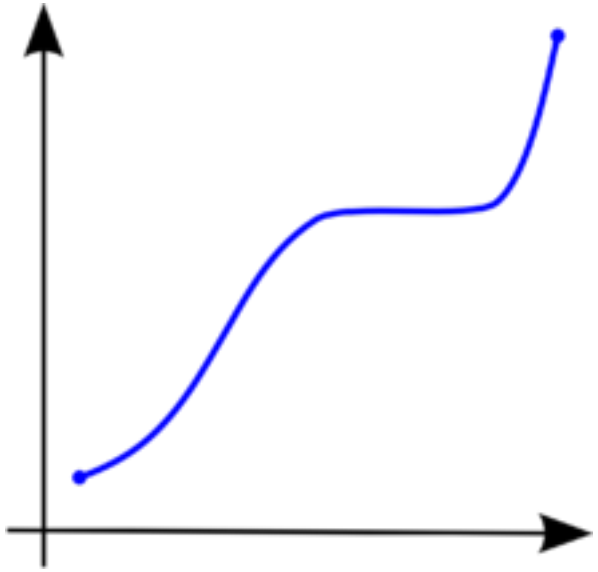
#3 prefer counters over gauges

Timeseries Have Types

Counter: monotonically nondecreasing

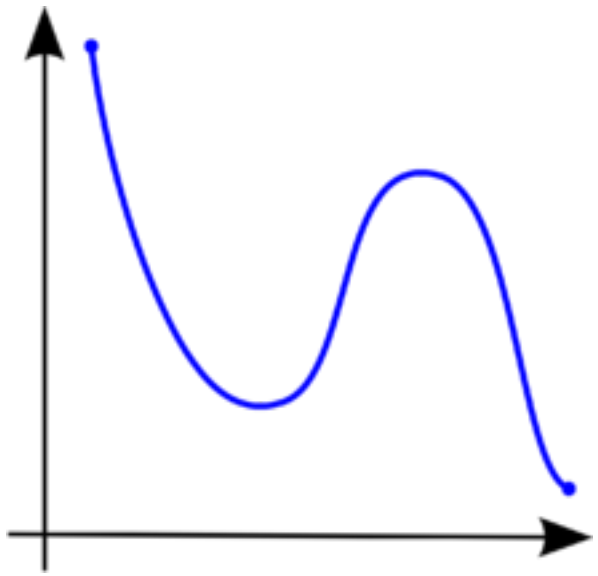
"preserves the order" i.e. UP

"nondecreasing" can be flat

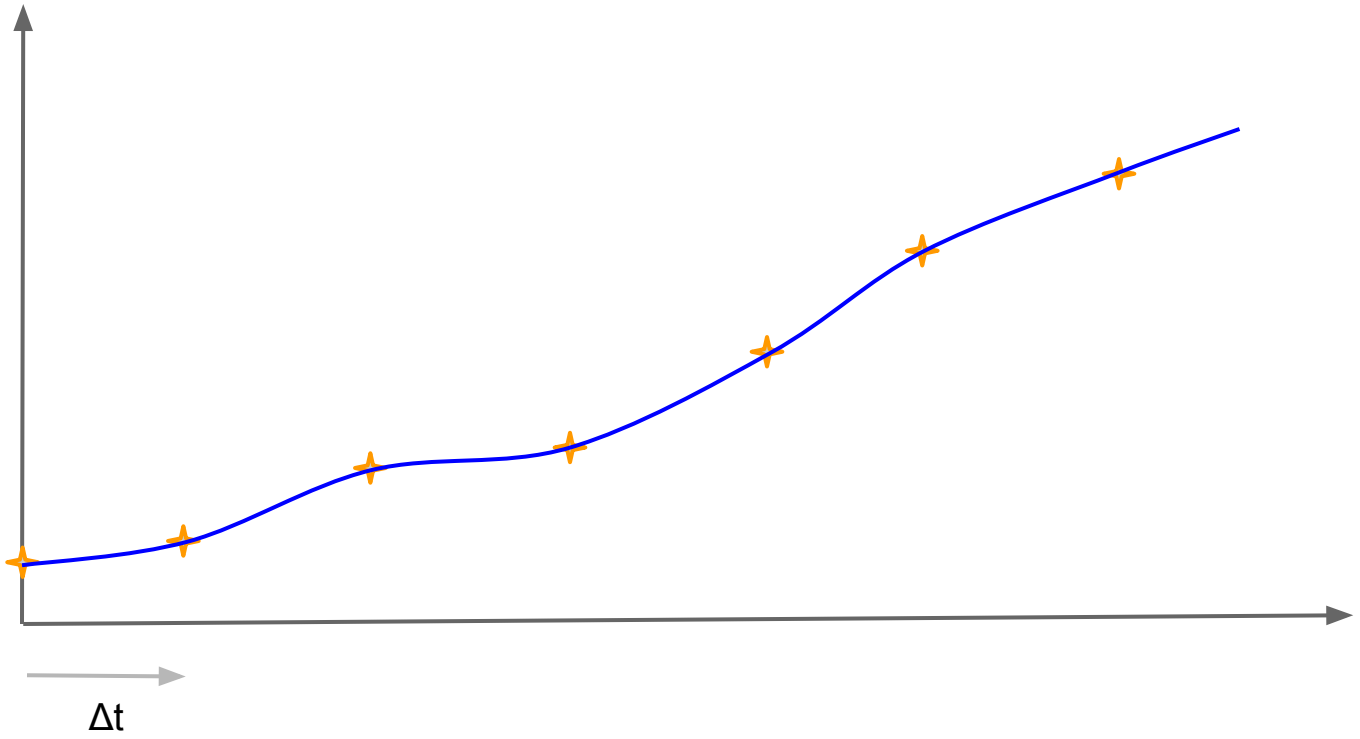


Timeseries Have Types

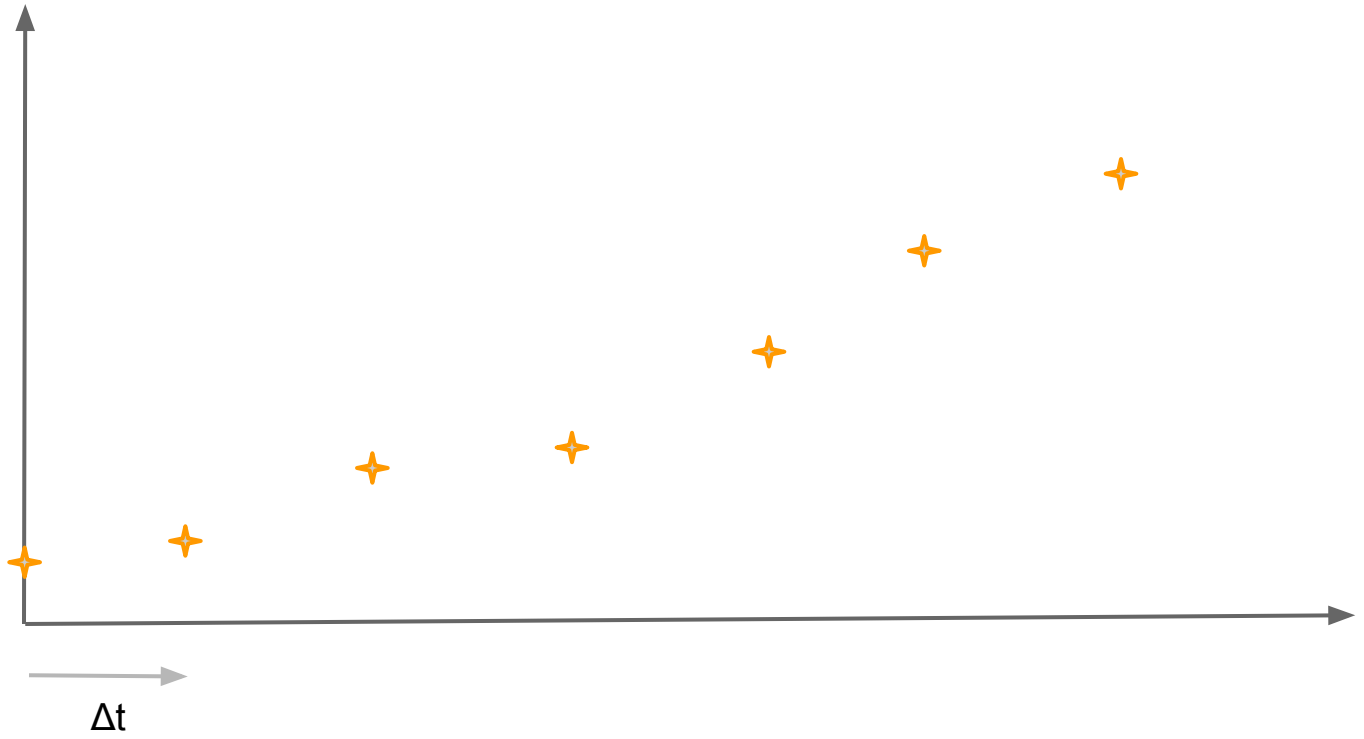
Gauge: everything else... not monotonic



Counters FTW

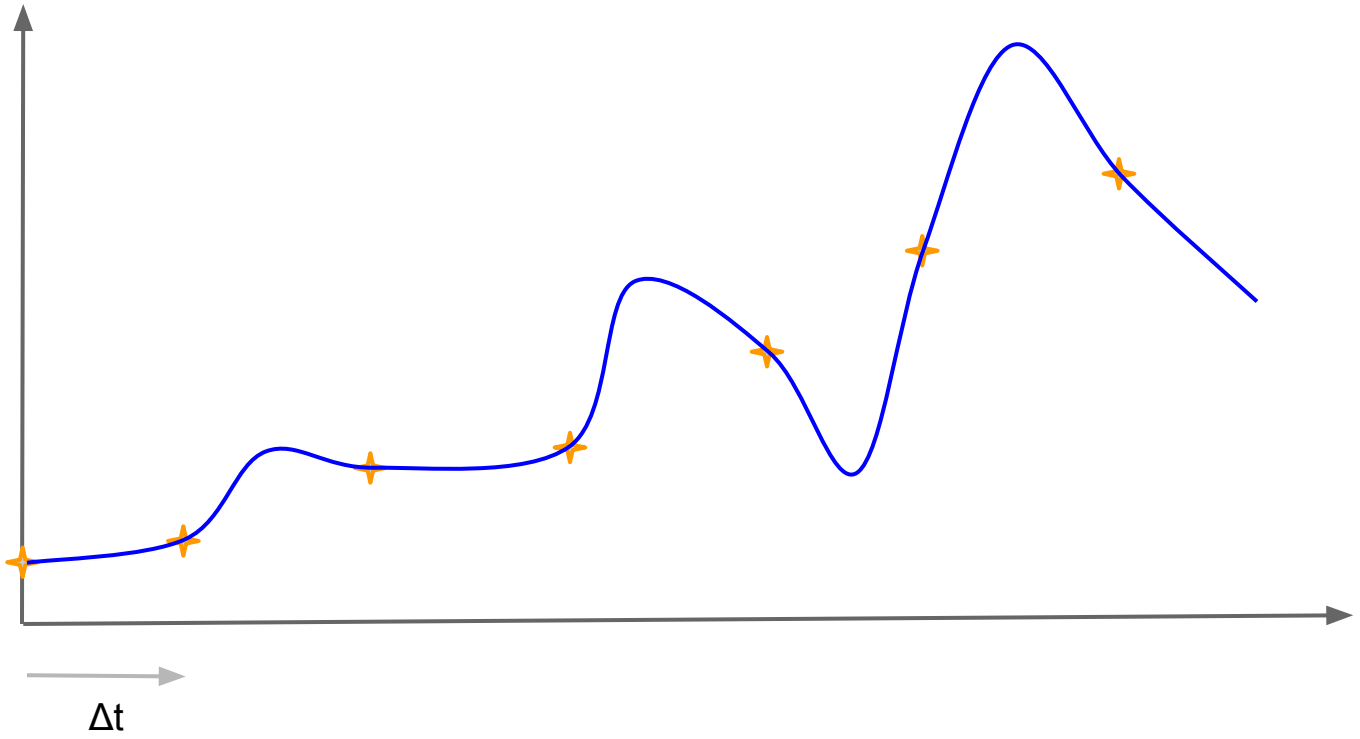


Counters FTW

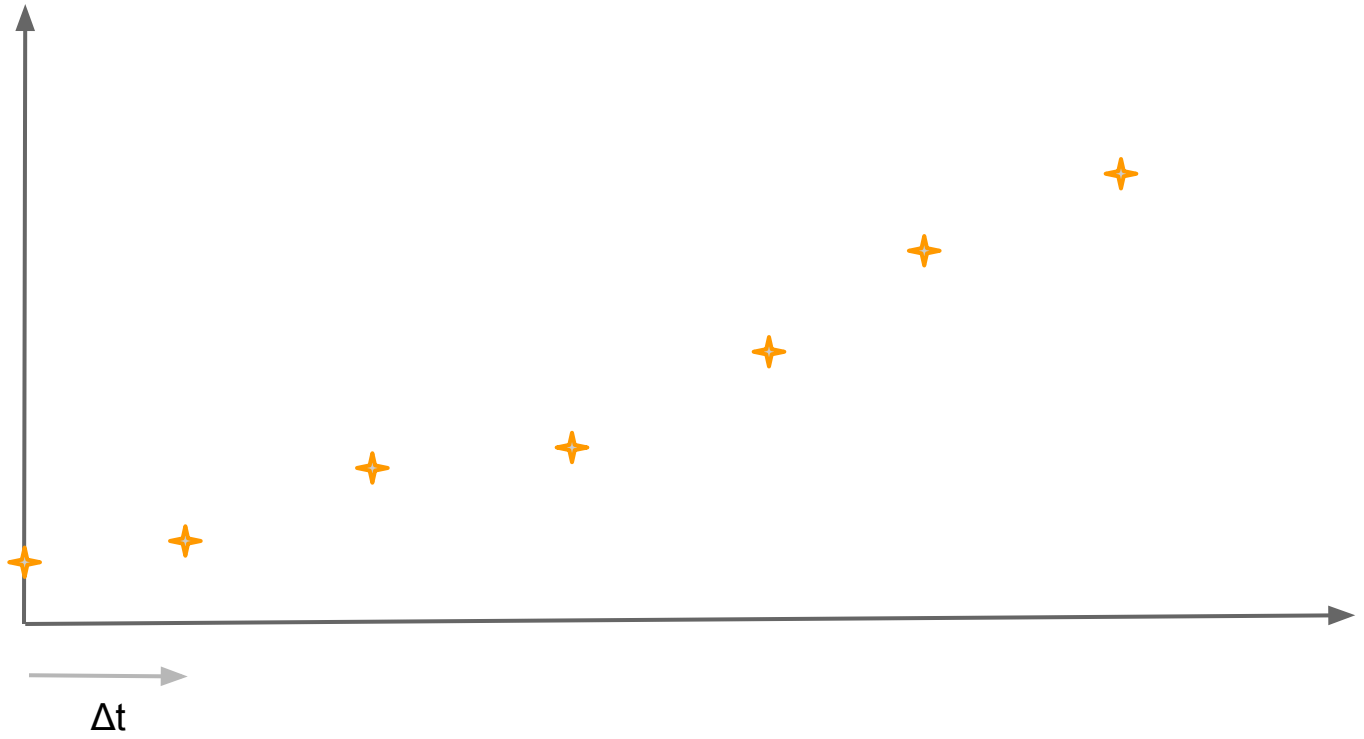


no loss of meaning after sampling

Gauges FTL



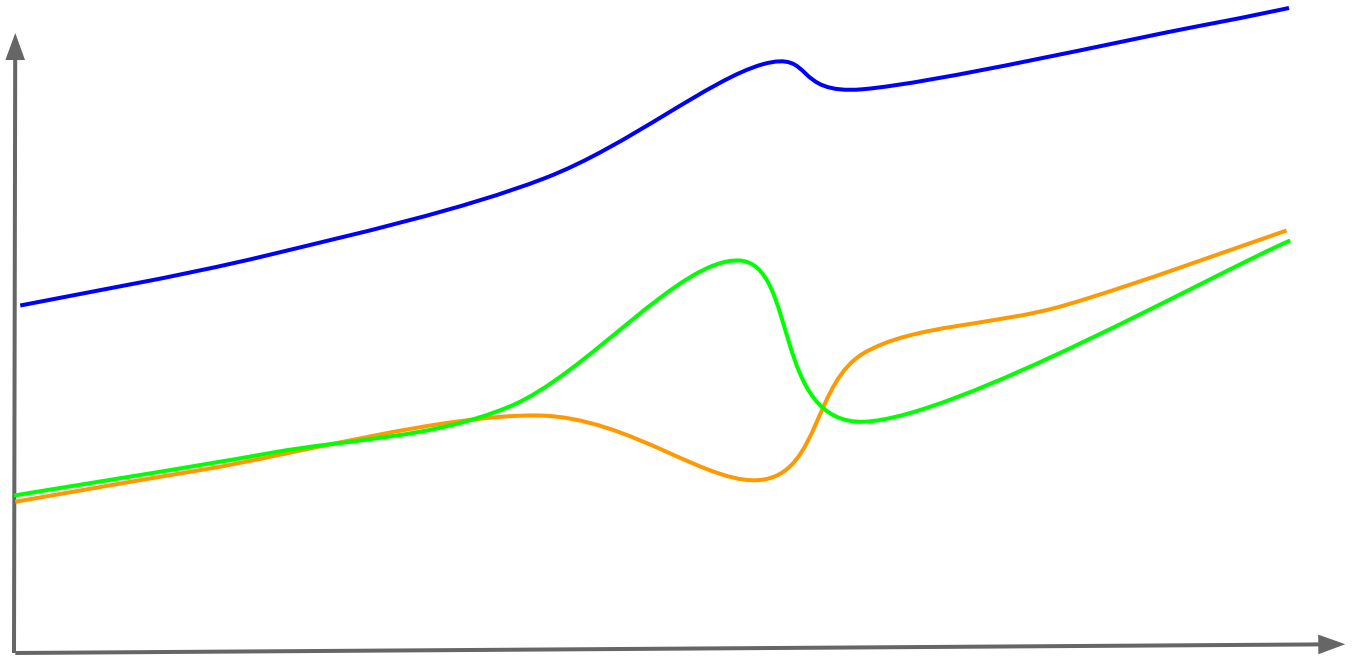
Gauges FTL



lose spike events shorter than sampling interval

#4 aggregate to each grouping in the system

Example: Aggregation



$$cluster\ rate_t = rate(instance\ 1_t + instance\ 2_t)$$

Aggregation of ensembles

Instances in a cluster don't work alone.

How many queries per second is your cluster receiving?

Take the rates of each counter, and then sum the rates.

#5 ratios of rates

Timeseries Operations: Ratios

counter / counter = counter: instant means

gauge / gauge = gauge: rate comparisons

e.g. New deployment

δ*(errors) /* δ*(queries) > threshold*?

What to alert on?

- Rate of change of QPS outside normal cycles
- Ratio of errors to queries
- Latency (median, 95th percentile) too high
- Rate of change of bandwidth

Whatever is important to the business!

When to alert?

Perhaps the topic of a whole other talk.

Alerts are not logs.

Make sure it's **ACTIONABLE...**
then **DOCUMENT IT**

3-month-in-the-future-you will thank you.

Blackbox testing still necessary

Blackbox tests are end-to-end tests.

End-to-end testing by definition covers everything you have missed.

You still have charts in the timeseries to inspect, right?

TL;DL

- **Do maths on your timeseries**
- Keep counters instead of gauges, derive rates
- Compare them to one another
- Do historical analysis (compare values over time)
- Alert only when action can be taken

HOW?

Attach a statistical package to your timeseries database, and experiment

- R
- numpy
- your favourite here

Make smarter alerts!
(AWKWARD DEMO TIME)

Questions?

Demo code:

http://github.com/jaqx0r/blts