adjust

Wiktor W Brodlo

- sysadmin at adjust
- don't even like databases

<wiktor@adjust.com>

adjust

- the custom datatypes place
- mobile analytics
- lots of datapoints
  - ⇒ 3 PB/month in 2017
- no cloud

Elasticsearch cluster

- ephemeral event log
- 1.3 PB
- 500 billion documents
- 47 machines (7 indexers) on 20 Gbps links
- avg 15 Gbps overall traffic, 80 Gbps peak

and not a single fulfilled query...

Elasticsearch problems?

- too much internal chatter; unscalable
  - ⇒ can't grow cluster
  - ⇒ queries just time out
- JVM GC pauses at 100 GB heap

basically a very expensive /dev/null

2017-08-04

Enter Bagger

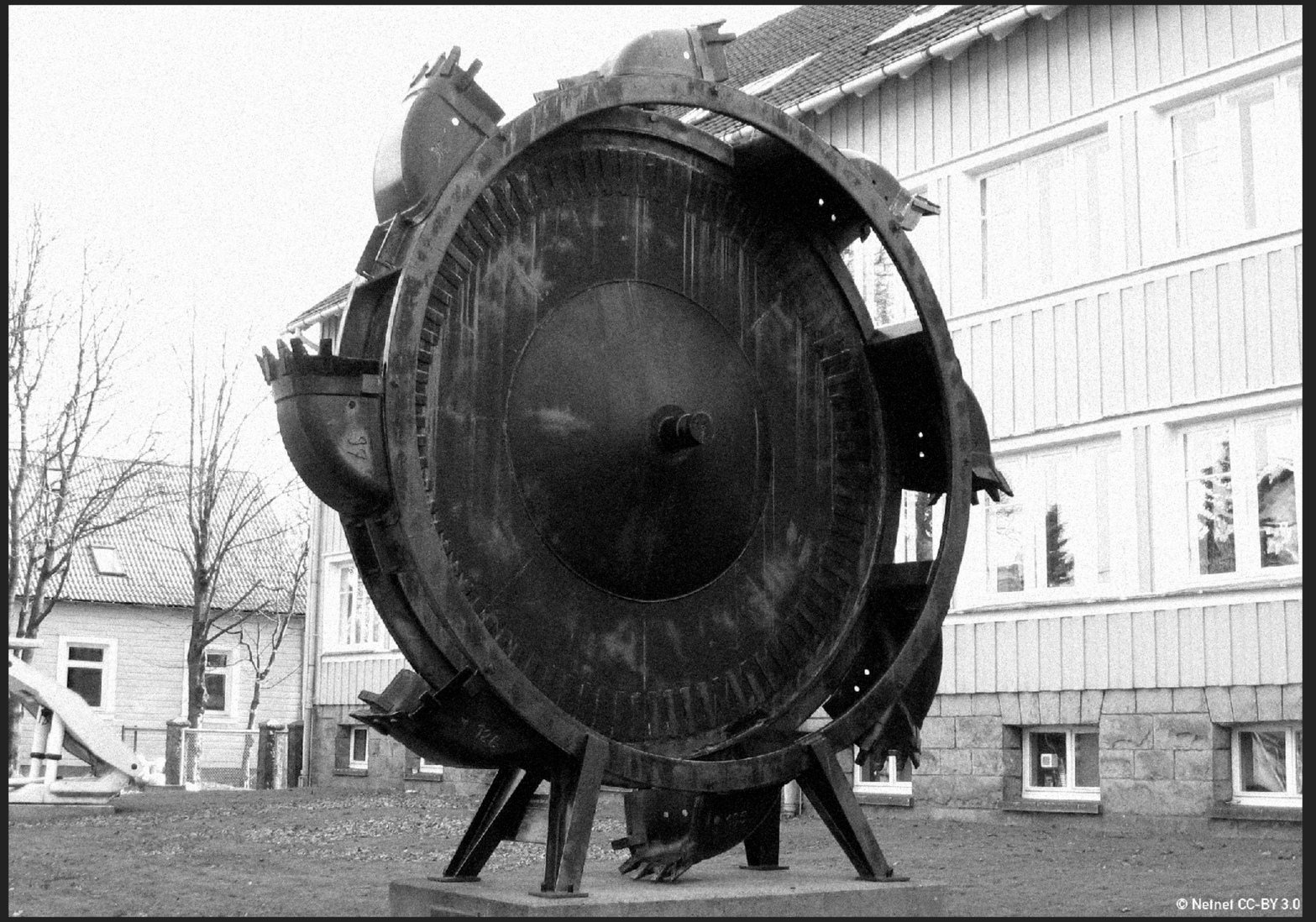
- not an exact replacement

  ⇒ tailored to our use case

Bagger

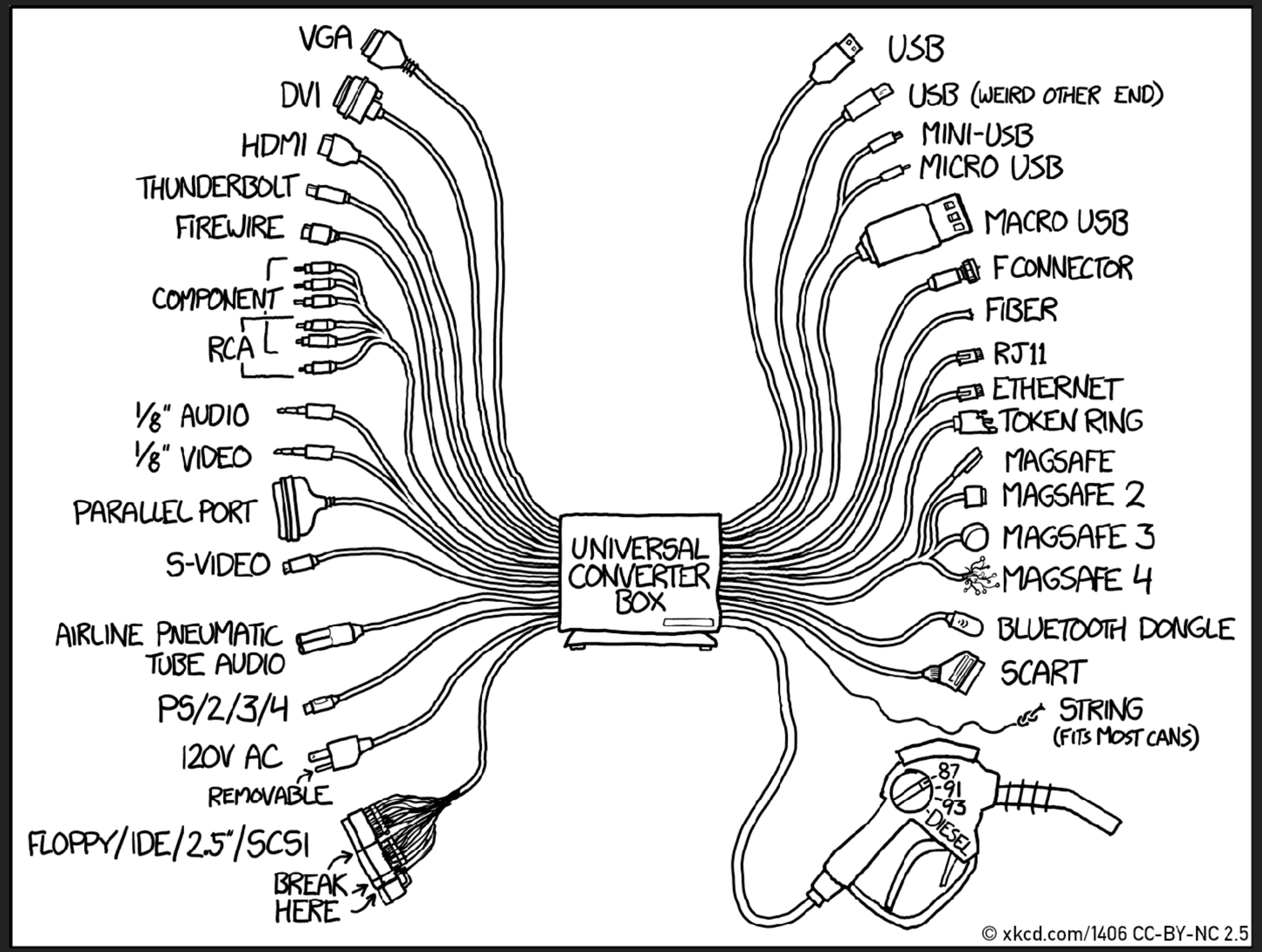
- again, event log
- data points grouped by service, tag, timestamp
- mix of relational and JSON

Redis–Kafka–Postgres


© Netnet CC-BY 3.0

Schaufel

- takes a pile of data and throws it somewhere else
- our (somewhat) universal connector


VGA
DVI
HDMI
THUNDERBOLT
FIREWIRE
COMPONENT
RCA
1/8" AUDIO
1/8" VIDEO
PARALLEL PORT
S-VIDEO
AIRLINE PNEUMATIC TUBE AUDIO
PS/2/3/4
120V AC
REMOVABLE
FLOPPY/IDE/2.5"/SCSI
BREAK HERE
UNIVERSAL CONVERTER BOX
USB
USB (WEIRD OTHER END)
MINI-USB
MICRO USB
MACRO USB
F CONNECTOR
FIBER
RJ11
ETHERNET
TOKEN RING
MAGSAFE
MAGSAFE 2
MAGSAFE 3
MAGSAFE 4
BLUETOOTH DONGLE
SCART
STRING (FITS MOST CANS)
87
91
93
DIESEL
© xkcd.com/1406 CC-BY-NC 2.5

The data

- timestamped events from various services
- grouped by hour
- deleted after some time
- not a big deal if we lose parts of the data set
- latency not a problem either

(Redis)

Kafka

- persistent queue
- just a fancy buffer

Schaufel

- multiple instances
  - ⇒ pairs
- reads from kafka, writes to postgres
- if a schaufel is down, kafka redistributes

Postgres

- each schaufel pair has a postgres pair
  - ⇒ redundancy
- one table per hour

Retention

- no vacuum!
  - ⇒ append-only
- hourly cron job drops expired tables

Bookkeeping

- master db
- generations
- partitions

Recap so far

Querying?

pull.pl – the sysadmin's answer to kibana

- just a perl script
- takes service, time, range, key
- asks master for partitions matching query
- explain
  - ⇒ refuse sequential scans
- prepare + fetch
- like mapreduce

Stats

- 30 machines
- WAL on 4 × 2 TB SSD
- data on 8 × 8 TB HDD (compressed zfs)
- data:index 8:1

Stats

- 2 million partitions
- 240 instances
- million inserts per second

Misc

- great for testing
  - ⇒ send us your test cases <wiktor@adjust.com>

Conclusion

- don’t be afraid to be radical
  - ⇒ don’t be stuck with crappy tech
  - ⇒ we were unhappy so we went and changed it

(worst case quit your job and come work with us)

спасибо за внимание