Feel the Hum of Your System

Kresten Krab Thorup, Ph.D.
Humio CTO

About Me

Kresten Krab Thorup
@drkrab

NeXT - Trifork - Erlang - Humio


Feel
the
Hum


Feel
the
Hum

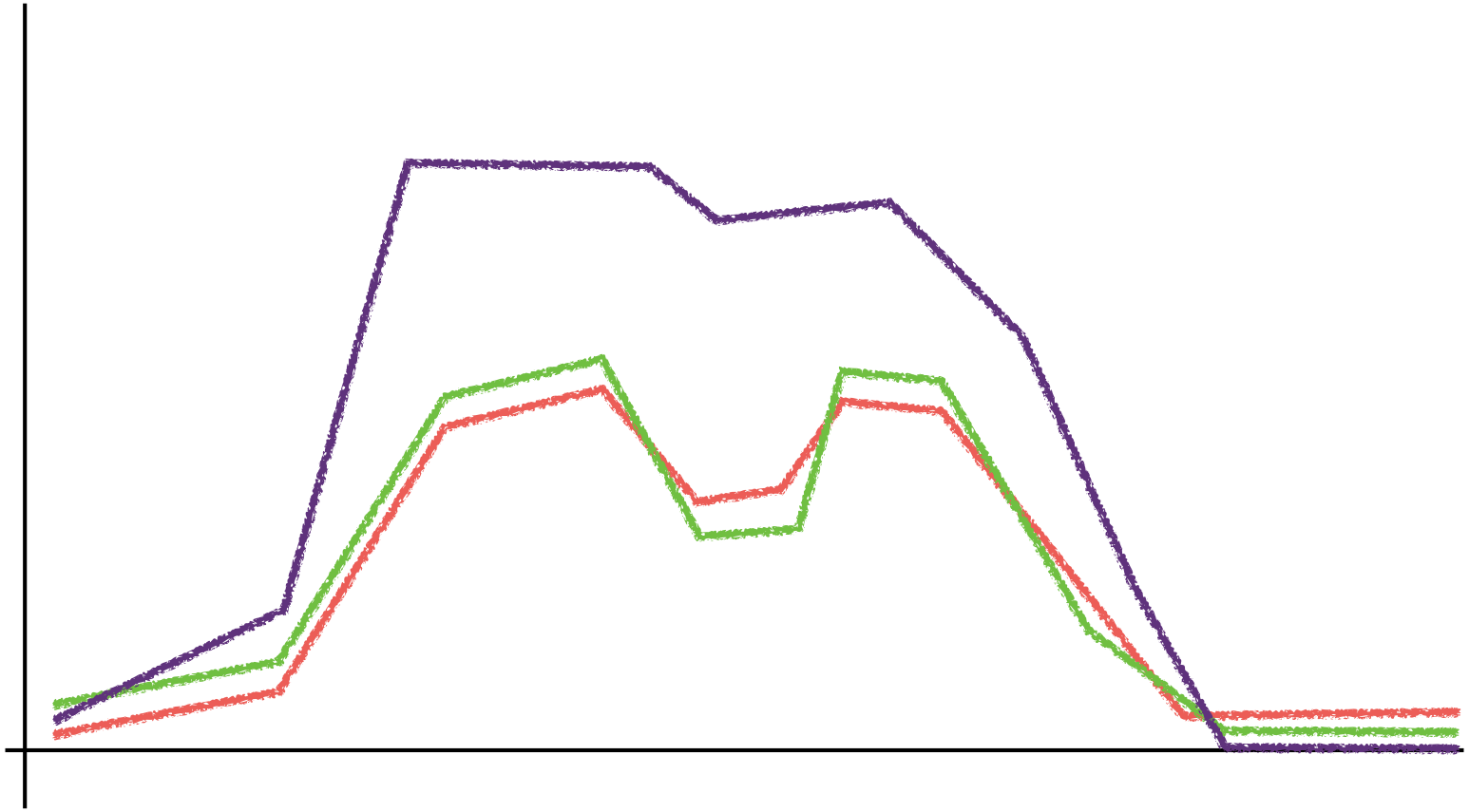

Fri 13 Jan 11:02:46 CET 2017
ID Age Due Description Urg
17 7w -3w Rotate production keys 12.3
19 6d -7d Fest 12
2 tasks
Execution Pipeline
3Ks
0min
0
0
92
3s

Our Background…

- 10 years ago … Trifork building the danish centralised national prescription services.
  - Availability
  - High security requirements
  - 50+ servers, 10+ services
  - 40+ connected systems

HUMIO

Sense of “normal”



HUMIO

Logs as a first class citizen

What you can do with logs

Observe & Monitor

What you can do with logs

Understand and Debug

What you can do with logs

Make projections - trends

What you can do with logs

Archeology / post-mortem

What you can do with logs

Test & Validate

What you can do with logs

Dev | Ops

What you can do with logs

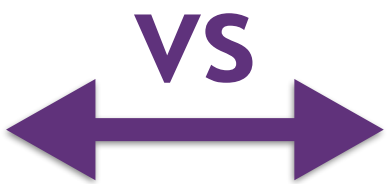
Prototype / Ad-Hoc Intelligence

What you can do with logs

- Observe & Monitor
- Understand & Debug
- Make projections - trends
- Archeology / post-mortem
- Test & Validate
- Dev|Ops
- Prototype / Ad-Hoc Intelligence

Interactive

(iterate! explore!)



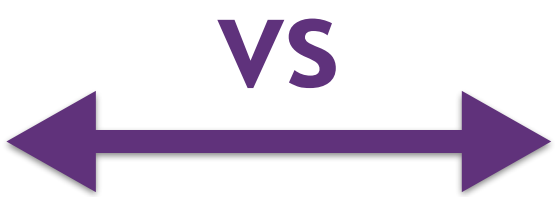
Static

(configure)

HUMIO

Historic

(map/reduce processing)



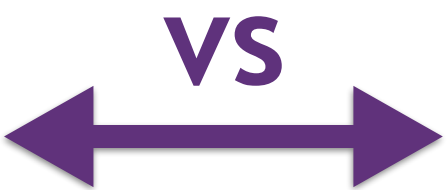
Live

(stream processing)

HUMIO

Schema on Read

(keep the text)



Scheme on Write

(discard the text)

HUMIO

Iterate
Explore
Visualize

Everyone should do this

HUMIO

Our Engine

What we want to achieve

- Fast & Flexible, and Affordable
- Simple to use
- Install on-premise or use as SaaS

Humio is a "Time Series TextDB"

- In-memory Stream Processing
- Support for live queries
- No term indexing
- Compressed storage
- HTTP/JSON API

Data In: Ingest

API's

- HTTP/JSON ingest API
- Elastic Bulk API / (beats, fluentd, …)
- NetFlow (firewall logs)
- Syslog

Ingest Performance

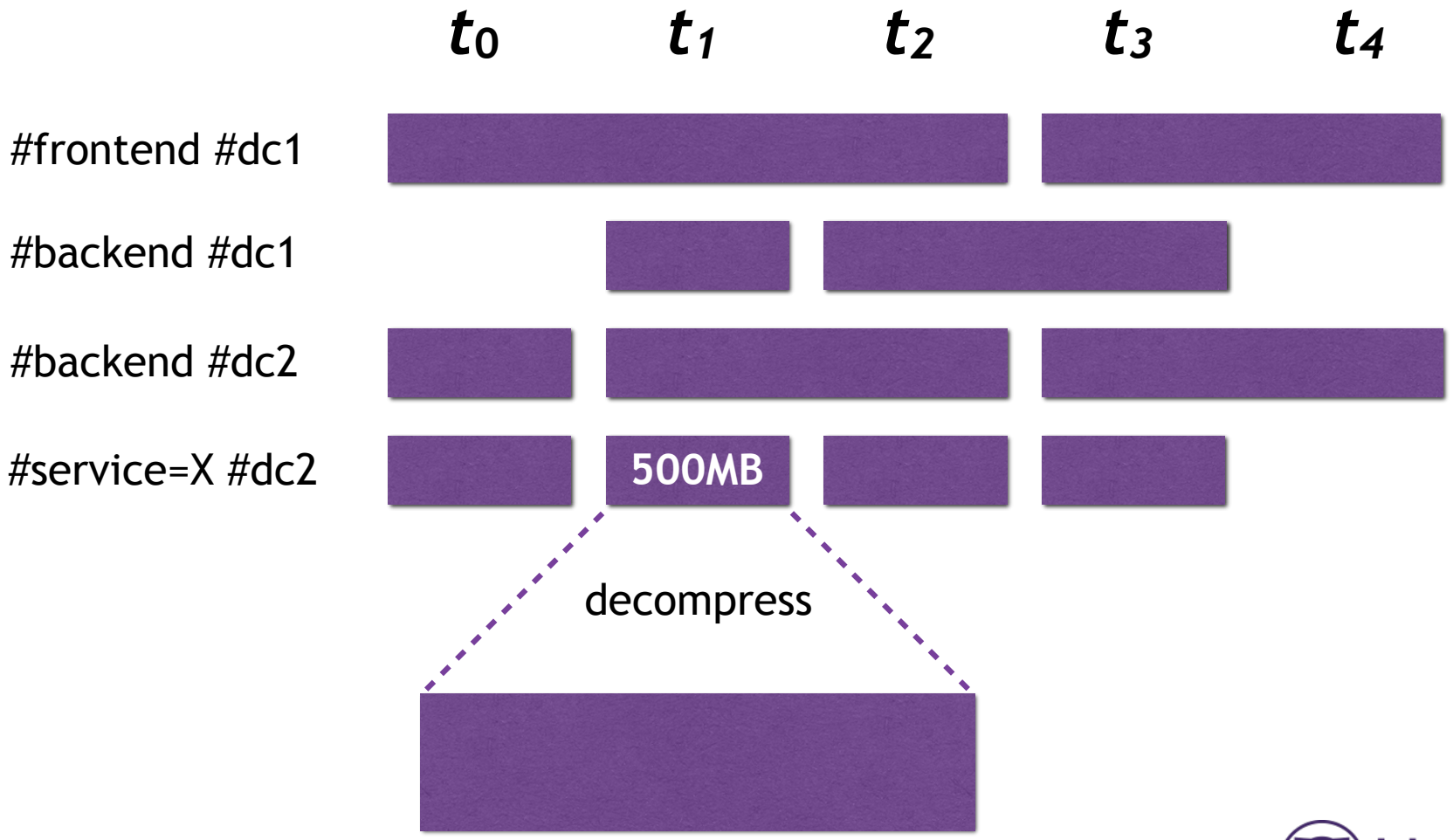
- Arriving data is processed minimally, just compressed and persisted, append-only.
- Reference platform Intel i7, 4core node ~12MB/sec (1TB/day) uses ~50% cpu (leave space for querying).
- Typical compression is 5-10x input data, so 1 day of 1TB takes 100-200GB disk space.
- Ingest scales ~linearly with cluster size.

Data Out: Queries

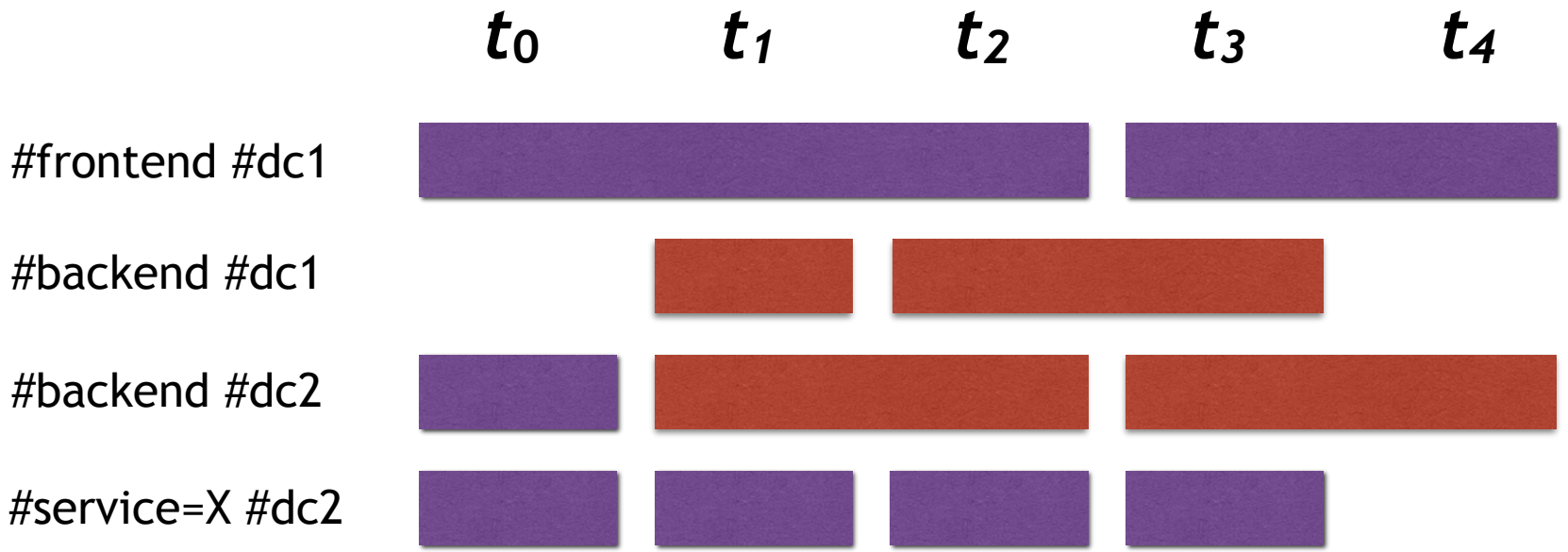
```
#backend #start=1d
  | url=/conference/27/*
  | groupby(method,
              select=avg(response_time))
```

```
SELECT AVG(response_time)
  FROM ‘backend’
  WHERE url LIKE ‘/conference/27/%’
    AND time > ‘2016-10-25T12:30’
  GROUP BY method;
```

Storage Segments

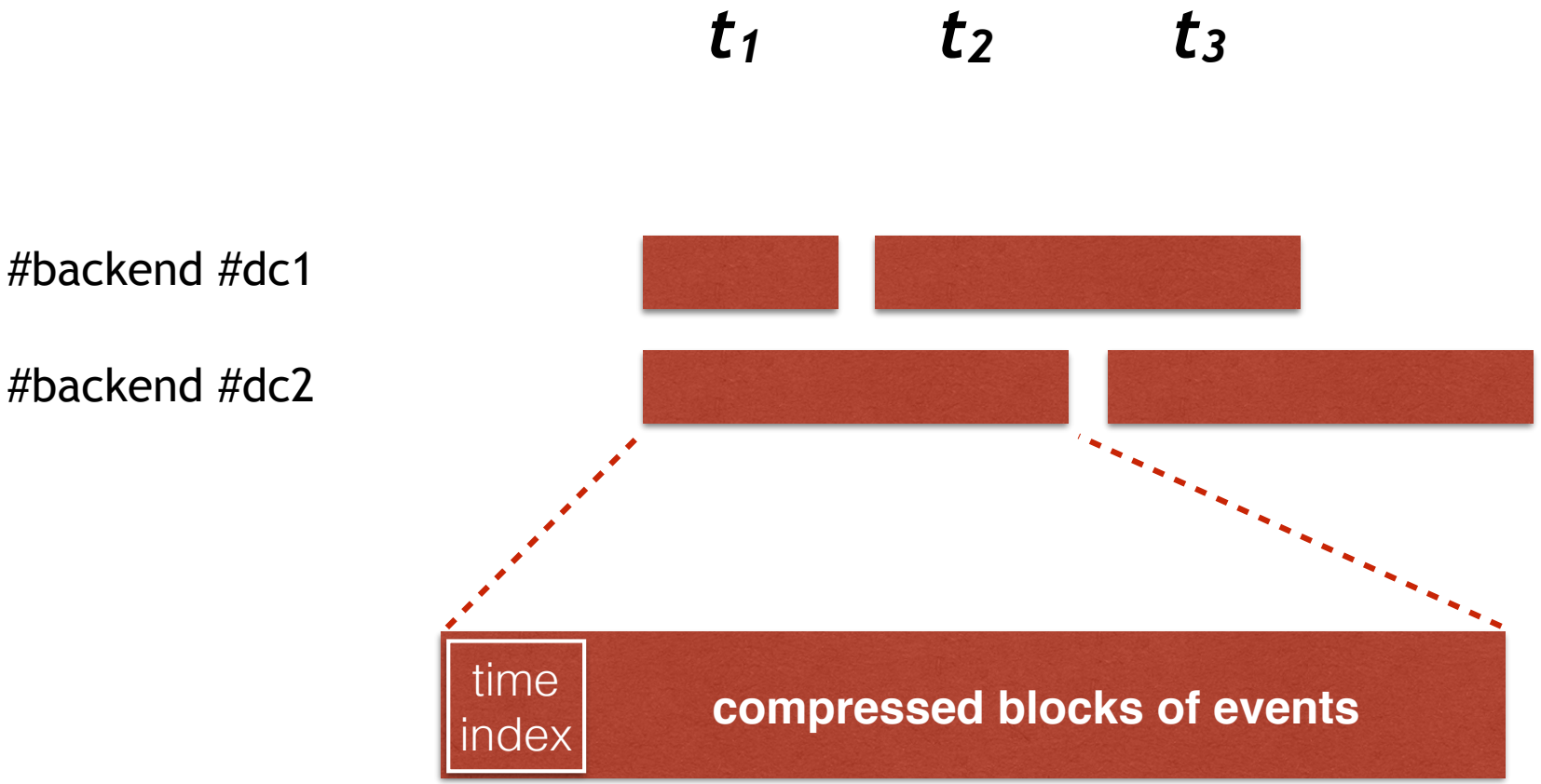


#backend, $t_1..t_3$



HUMIO

#backend, $t_1..t_3$



HUMIO

```
#backend #start=1d
 | regex("/conference/(?<conf_id>[^/]+).*",
         field=url)
 | groupby(conf_id, select=count())
```

Flexibility of free text

HUMIO

Query Speed

- Reference Platform: Intel i7, 4core
- Free text ~4GB/sec/node
- I.e., 7-node cluster, ~28GB/sec
  free text search 1TB in 36 secs
- Live queries respond “immediately”

HUMIO

Humio tries to be...

- Fast & Flexible
- Affordable TCO
- Simple to use
- Tailored for LOGS

Thanks

@MeetHumio