Fast Cars, Big Data

How Streaming Can Help Formula 1

Contact Information

Ted Dunning

Chief Applications Architect at MapR Technologies

Committer & PMC for Apache's Drill, Zookeeper & others

VP of Incubator at Apache Foundation

Email tdunning@apache.org tdunning@maprtech.com

Twitter @ted_dunning

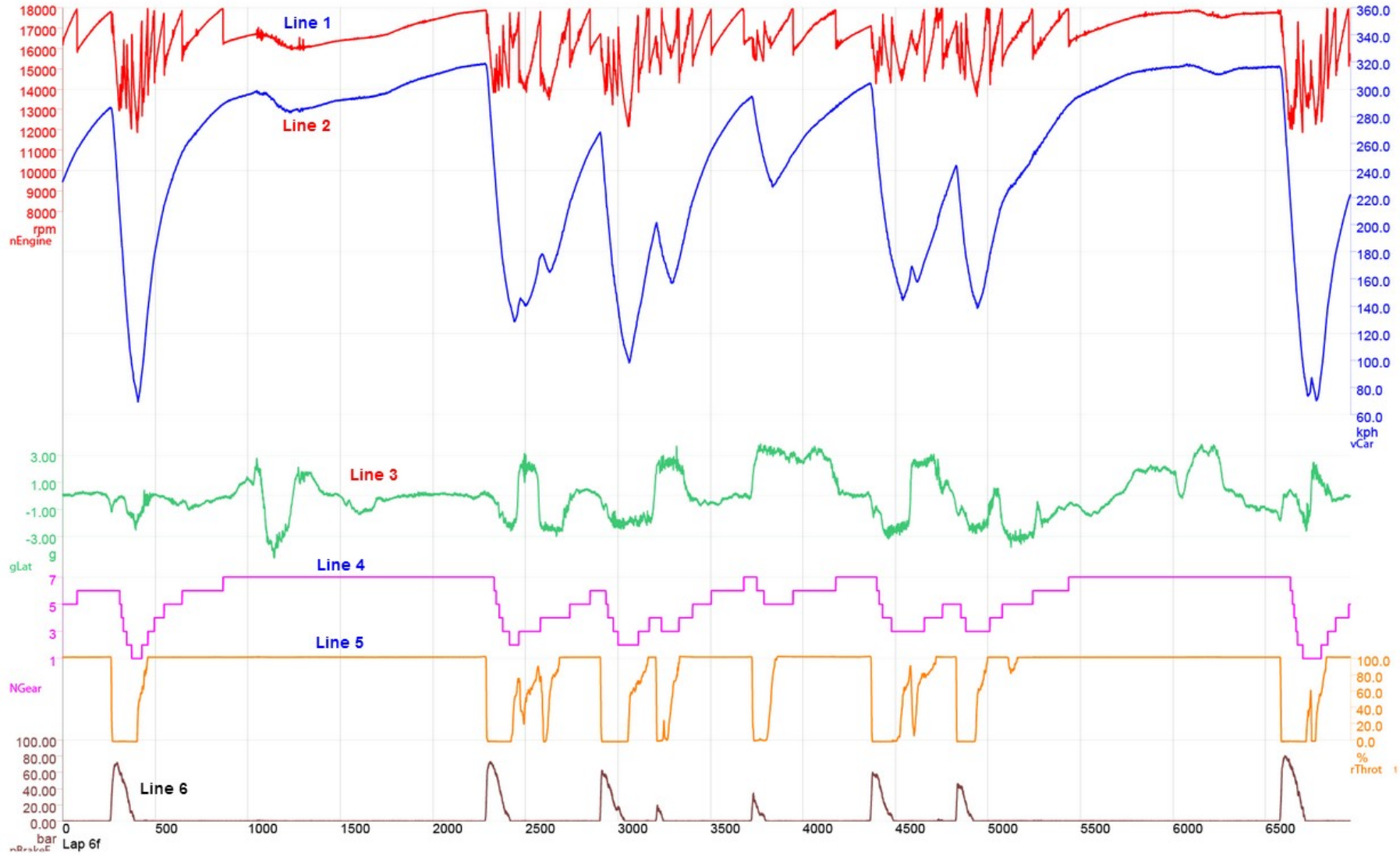
Hashtags today: #bbuzz @mapr

Agenda

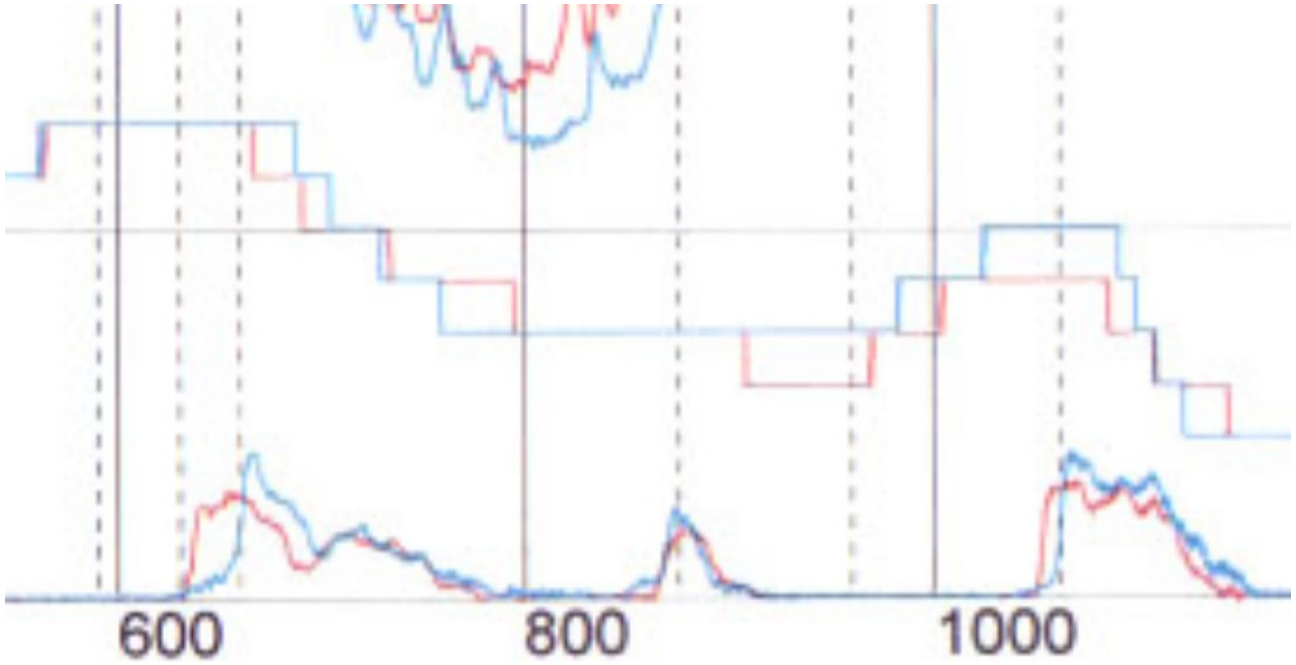
- What's the point of data in motorsports?
- How can we play too?
- KPI preserving generation
- How the sim works
- What works?
- Live demo

How data plays in F1 motorsports

Data in Motorsports



https://scarbsf1.wordpress.com/2011/08/18/telemetry-and-data-analysis-introduction/



Difference is due to later and sharper braking

Real Analytics as Well as Visualization

- Inputs

  Predictive analysis of consumables and tires

  Physical models of car + driver performance

  – Tire wear slows lap times, lower fuel weight speeds lap times

  Game theoretic analysis of competitors' options

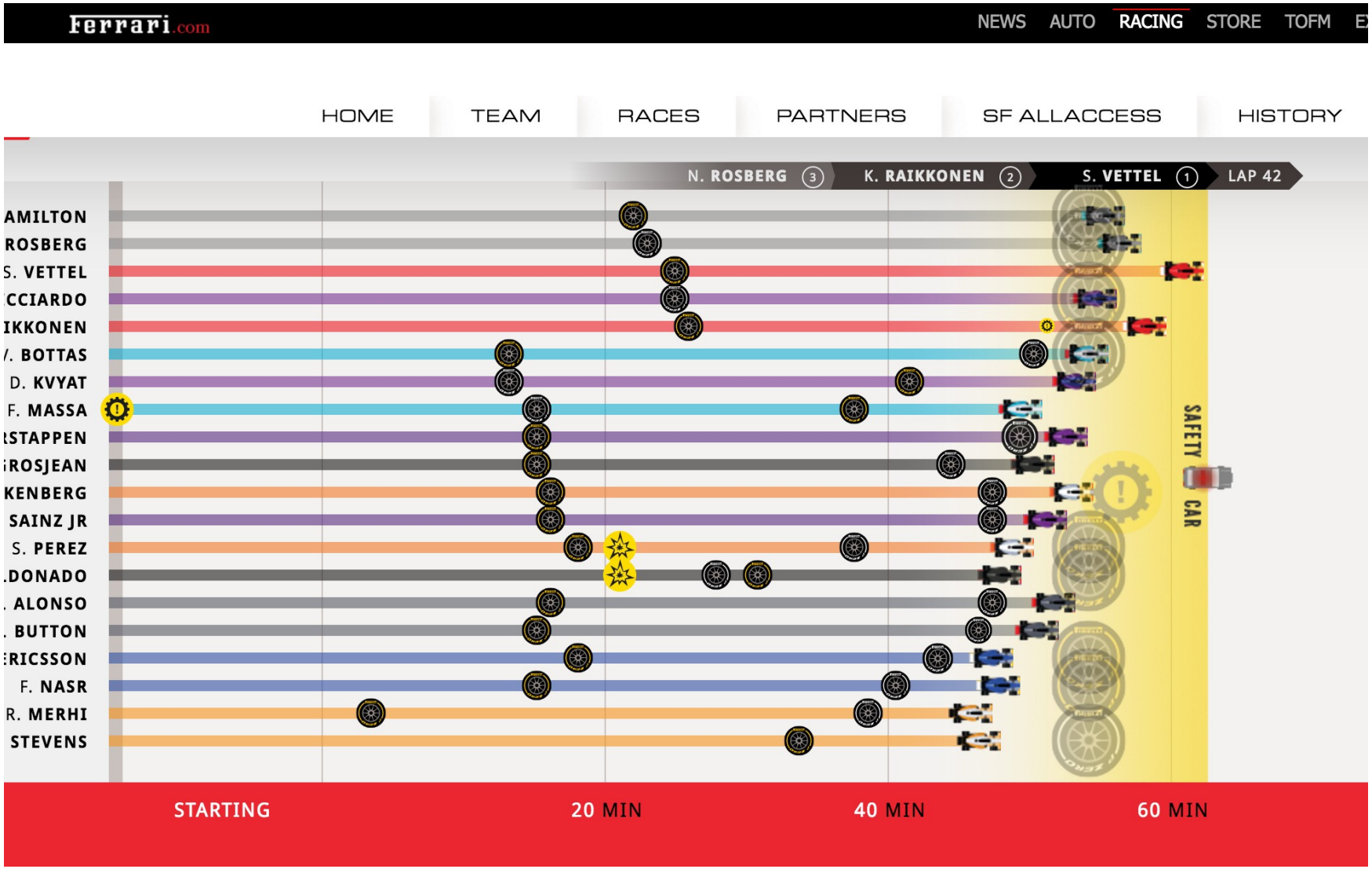
  Monte Carlo analysis of likely weather conditions

  Current GP points status

- Outputs

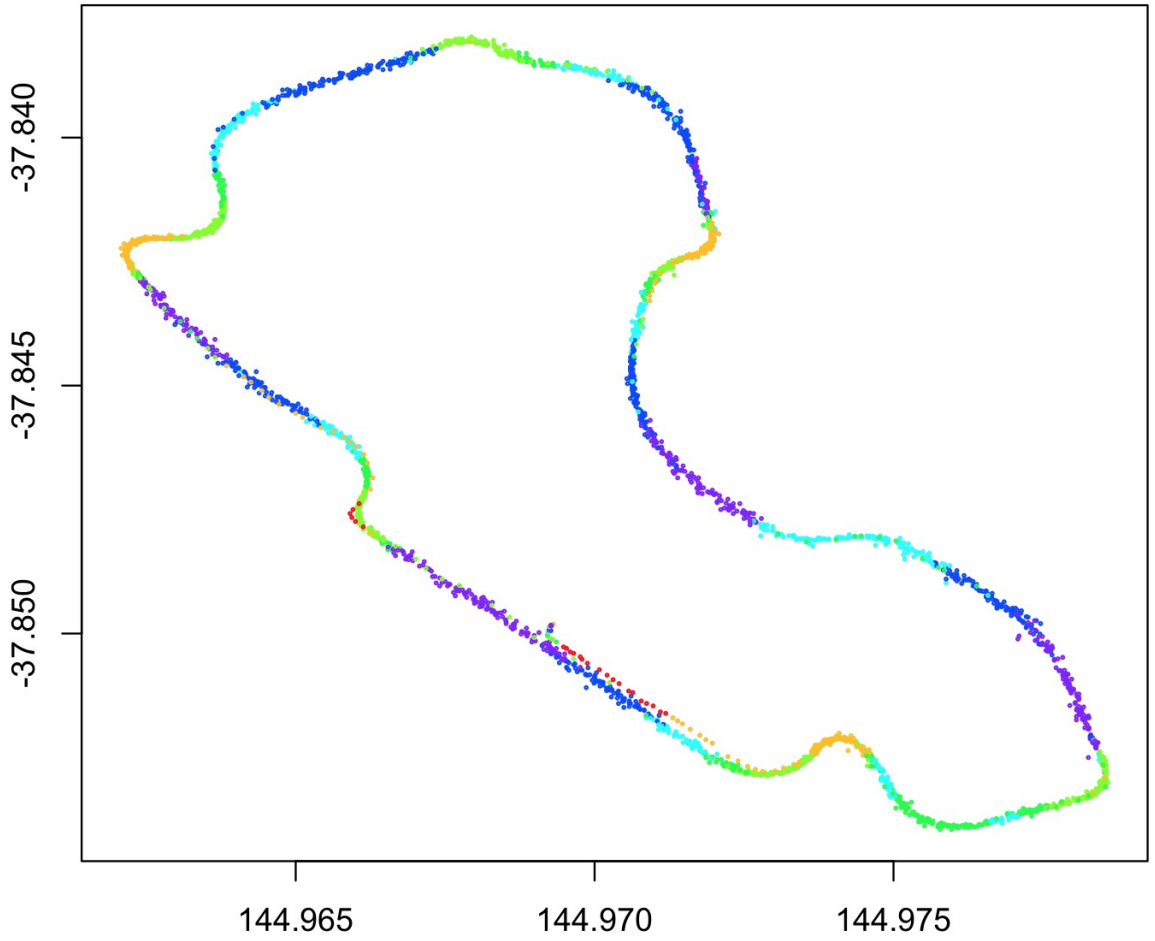
  Tactical options, outcome distributions

Data for Marketing as well



http://formula1.ferrari.com/en/inforacing-hungarian-gp-2015/

More Data = More Gearhead Engagement



Occasionally, somebody makes a tiny bit of data available like this data dump of Button's ride in the Australian Gran Prix

Fans go nuts with this

http://bit.ly/f1-data-plotting-with-r

http://bit.ly/f1-data-dump

But it still doesn't work ...

ENODATA !

- Available data is definitely not good enough for more than an occasional blog post
- Available data is intermittent, out of date and inconsistent
- Except in special cases we can’t really build publicly available systems from scrounged data

The Unrealistic Nature of Real Data

- Real data has several defects
  - We can’t share it
  - We can’t get it
  - We can’t break it
  - We can’t understand it

We have guarantees about practical fidelity

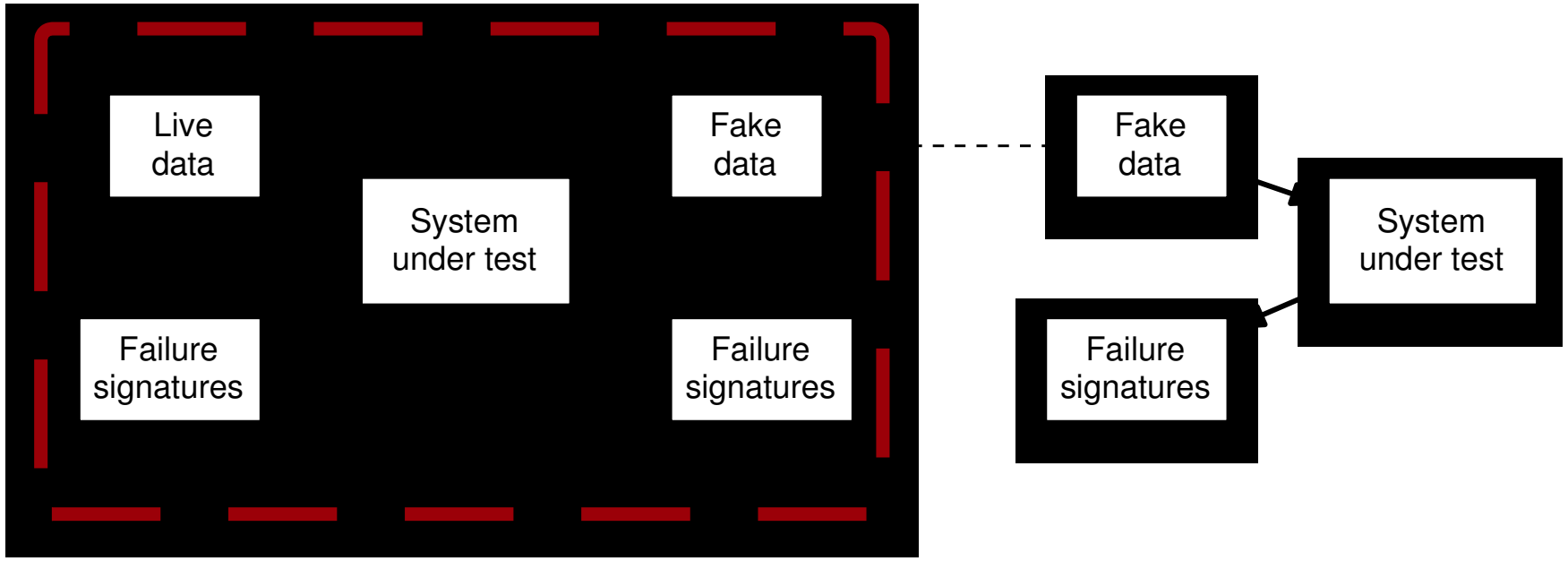
The Unrealistic Nature of Real Data

- Real data has several defects
  - We can't share it (***due to confidentiality***)
  - We can't get it (***too big, wrong scale, out of date***)
  - We can't break it (***injecting major real-world faults frowned upon***)
  - We can't understand it (***we don't know what really happened***)

We have guarantees about practical fidelity

KPI Matching Simulation

Parametric matching of key performance and failure signatures allows emulation of complex data properties



Matching on KPI's and failure modes guarantees ***practical*** fidelity

If it breaks the same,
it's as good as the same

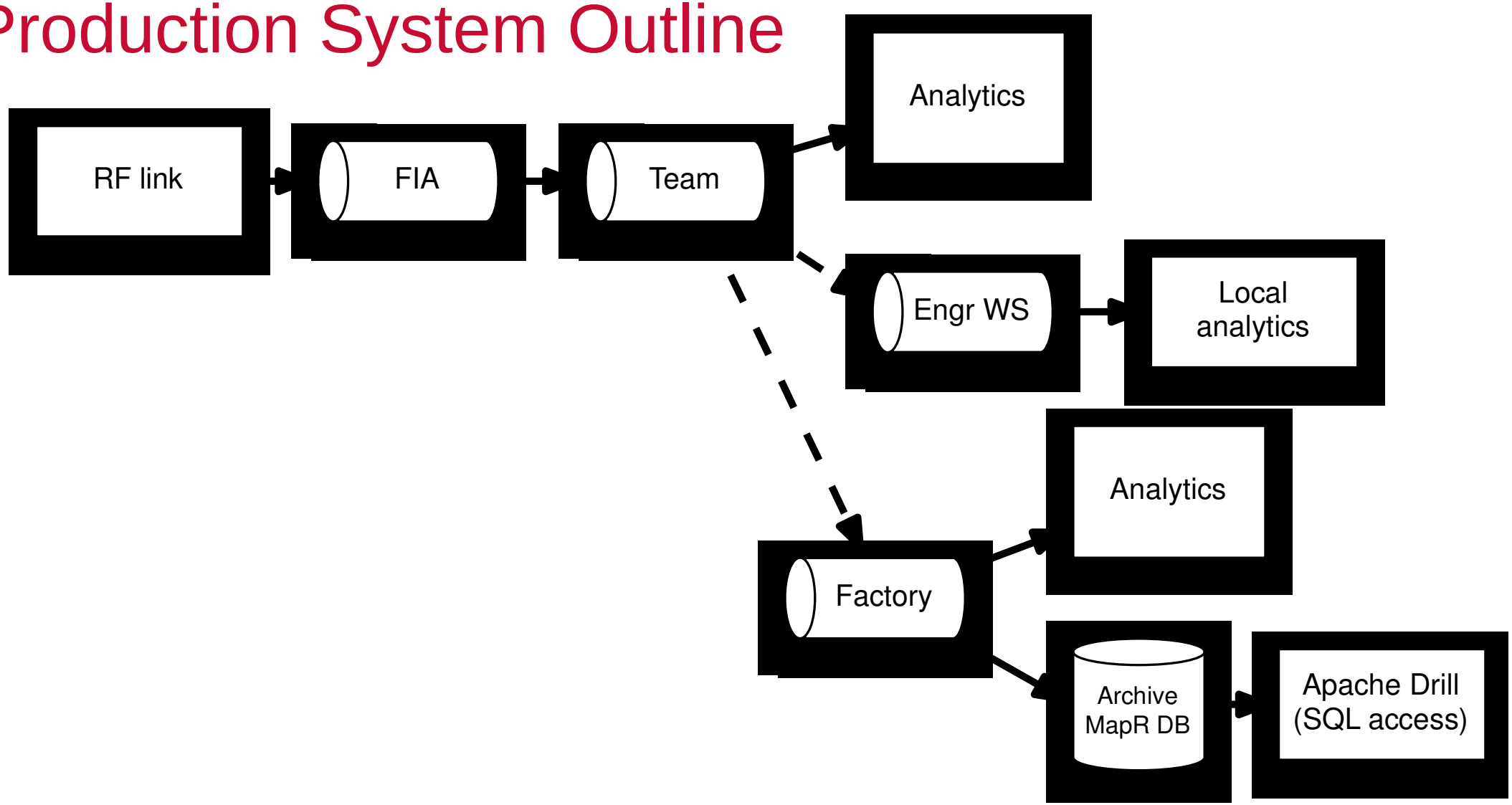
The Method

- Pick realistic and important KPI’s and failure measures
  - Sample rates, data volumes
  - Plausible physics
  - Plausible data semantics
  - Your mileage may vary
- Build emulation roughly based on real system
- Tune data spec to match KPI’s using real models
- Export data spec to alternative models
- Re-tune data spec to match on alternative models

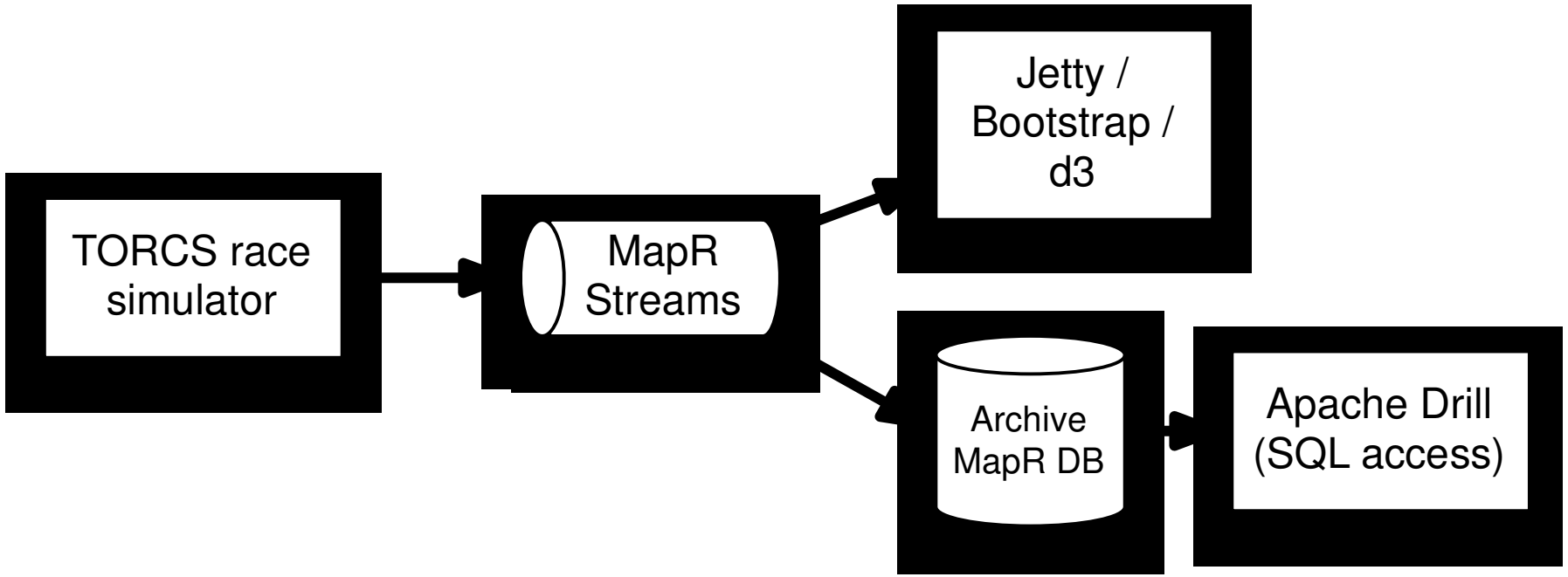
So how does that work?

So how does that work?
Especially for real-time data?

Production System Outline



Simplified Demo System Outline



TORCS for Cars, Physics and Drivers

TORCS is a pseudo-physics based racing simulator with full graphics output and pluggable control modules.

TORCS is commonly used for AI research, but the control model can just as well collect data

What is the Point?

- We would like to
  - Prove out software architectures
  - Test data pipelines and visualization systems
  - Tune UI’s

What is the Point?

- We would like to
  - Prove out software architectures
  - Test data pipelines and visualization systems
  - Tune UI's

  - Play video games?

What is the Point?

- We would like to
  - Prove out software architectures
  - Test data pipelines and visualization systems
  - Tune UI's

- We also need to
  - Simulate system failure scenarios
  - Push limits for future usage

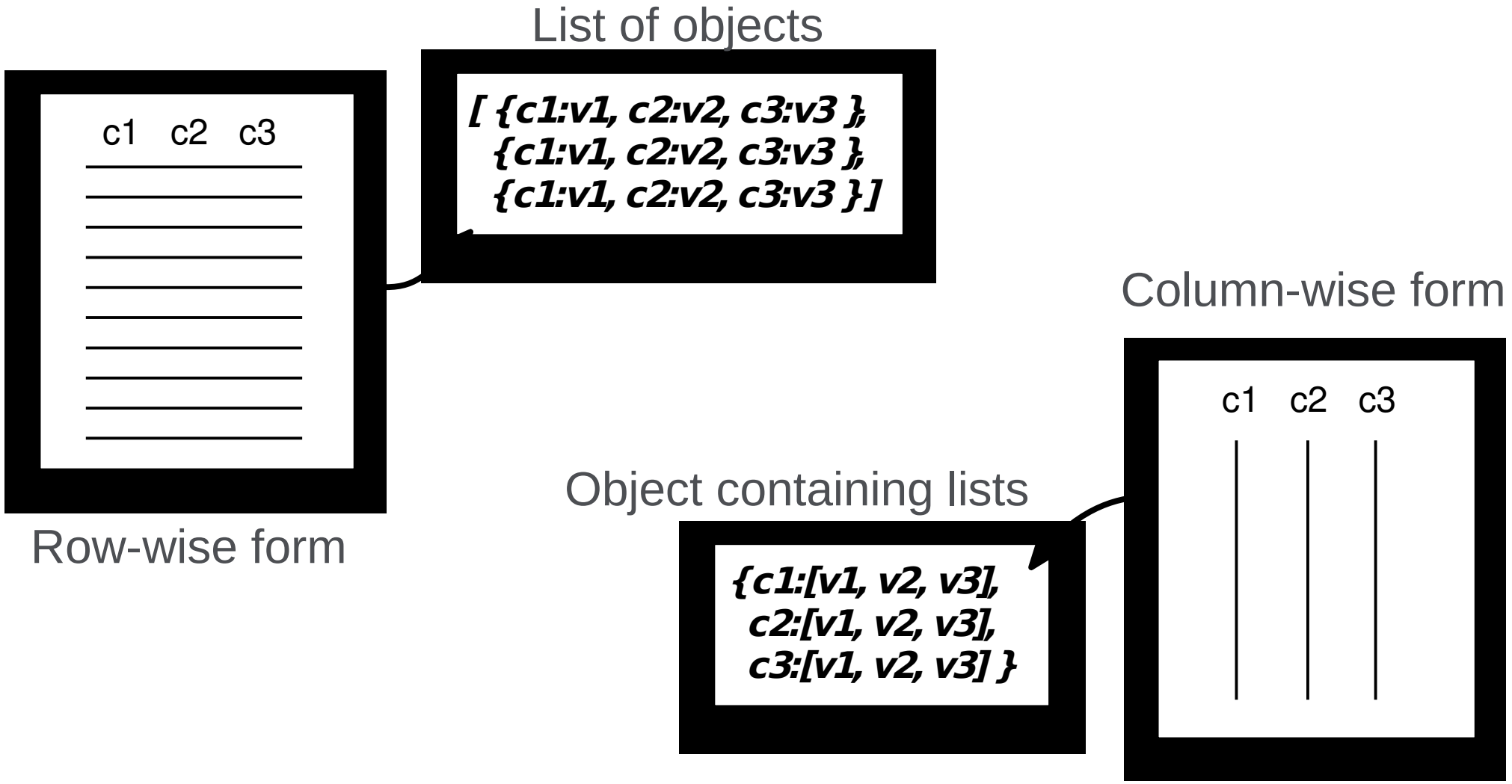
Current Status

- It works, is available on github, ASL 2

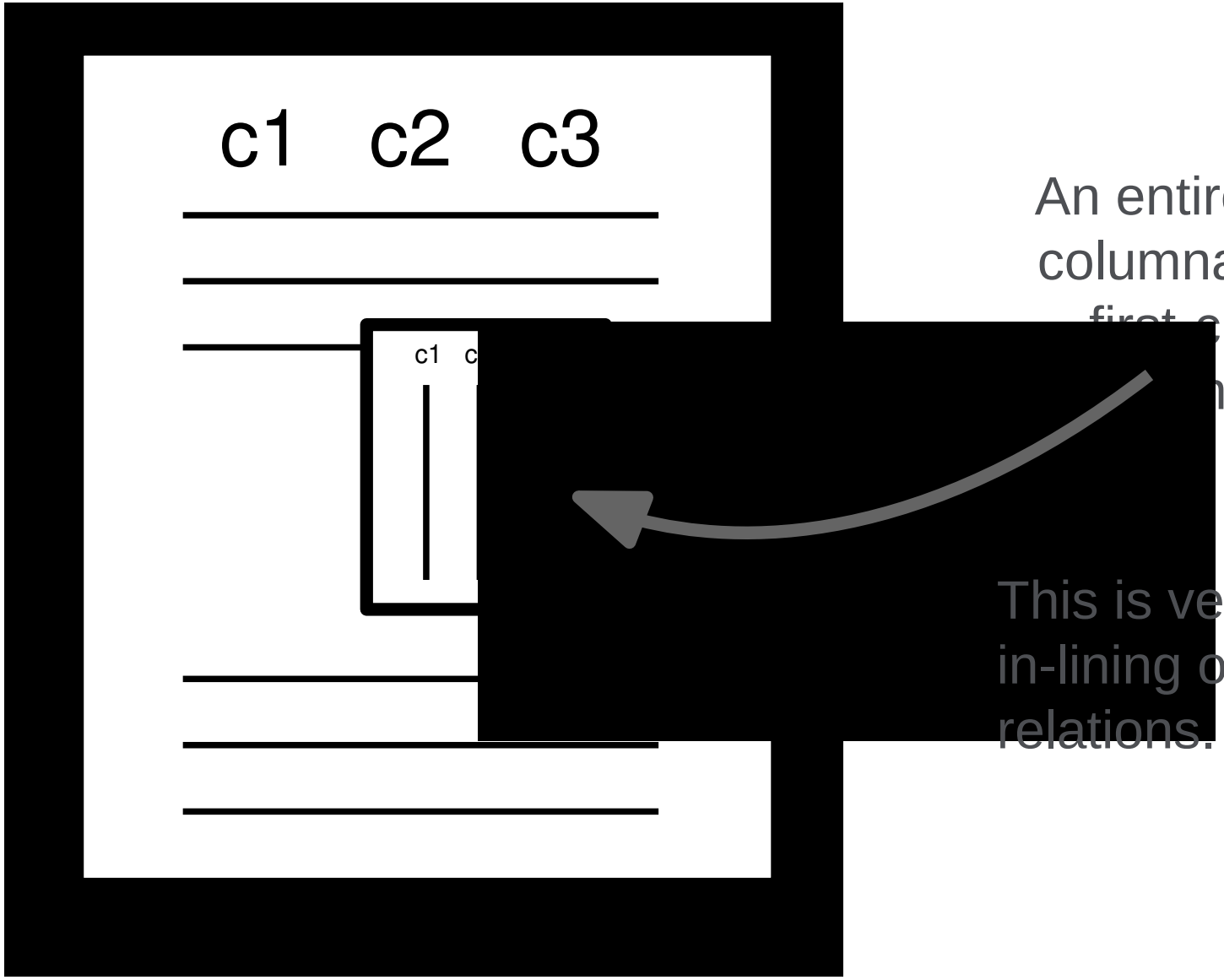
- Data collected is unrealistically limited, lacks
  - Tire pressure, temperature x 4
  - Brake usage, temperature x 8
  - Engine monitoring is primitive (RPMs only, no KERS)
  - Data rate is fixed, real data comes in at highly variable rates
  - Real data has variable delays due to RF dropout + buffering

- Data collected is in pure JSON
  - Real data is columnar compressed blobs

Tables as Objects, Objects as Tables



Micro Columnar Formats



An entire table stored in columnar form can be a first-class value using these techniques

This is very powerful for in-lining one-to-many relations.

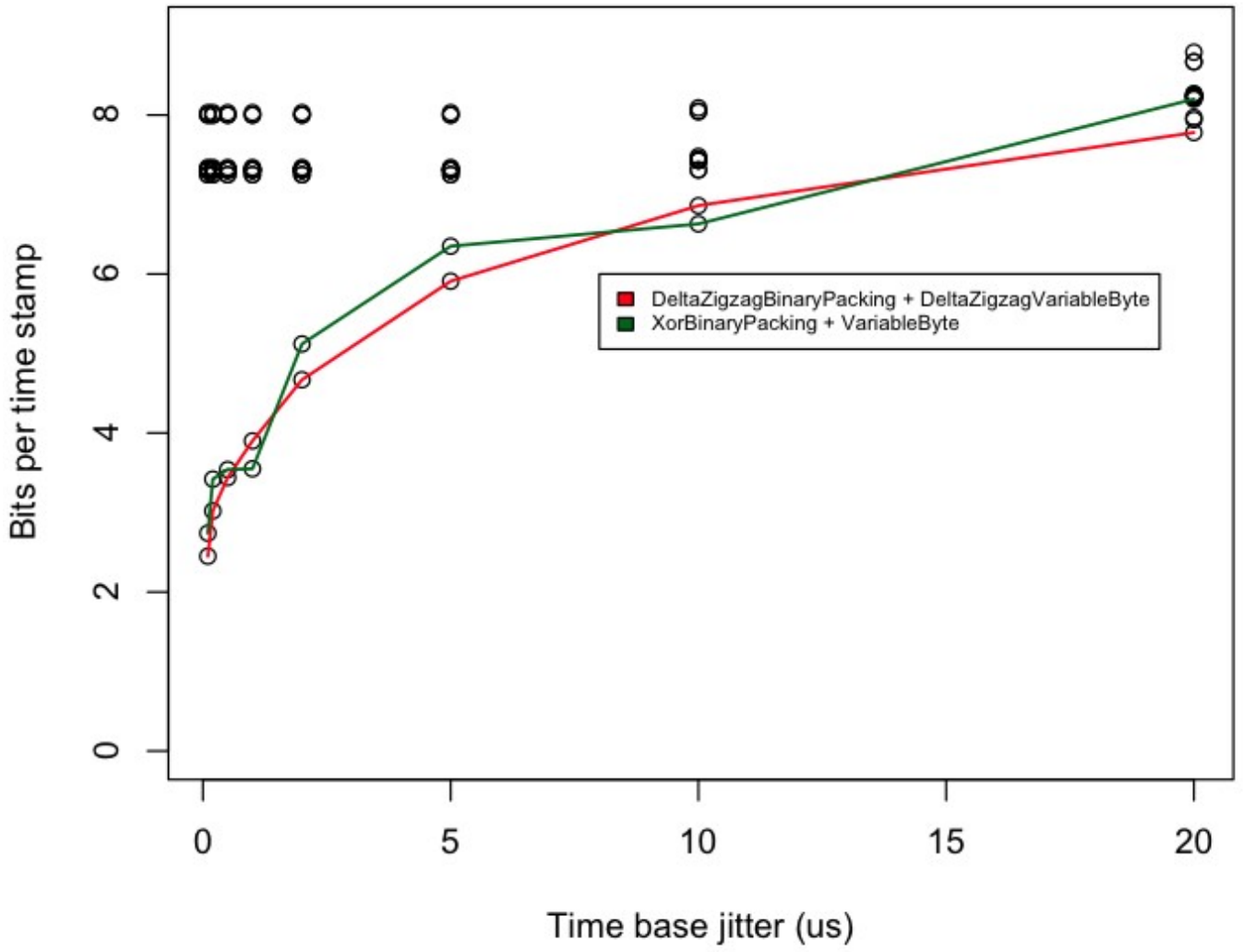
Compression Results

Samples are
    64b time, 16 bit sample

Sample time at 10kHz

Sample time jitter makes it important to keep original time-stamp

How much overhead to retain time-stamp?



Sensor Data V1

- 3 main data points:
  - Speed (m/s)
  - RPM
  - Distance (m)
- Buffered

```
{   "_id":"1.458141858E9/0.324",
    "car" = "car1",
    "timestamp":1458141858,
    "racetime”:0.324,
    "records":
        [
            {
                "sensors":{
                    "Speed":3.588583,
                    "Distance":2003.023071,
                    "RPM":1896.575806
                },
                "racetime":0.324,
                "timestamp":1458141858
            },
            {
                "sensors":{
                    "Speed":6.755624,
                    "Distance":2004.084717,
                    "RPM":1673.264526
                },
                "racetime":0.556,
                "timestamp":1458141858
            },
```

Sensor Data V2

- 3 main data points:
  - Speed (m/s)
  - RPM
  - Distance (m)
  - **Throttle**
  - **Gear**
  - **...**
- Buffered

```
{   "_id":"1.458141858E9/0.324",
    "car" = "car1",
    "timestamp":1458141858,
    "racetime”:0.324,
    "records":
       [
          {
             "sensors":{
                "Speed":3.588583,
                "Distance":2003.023071,
                "RPM":1896.575806,
                "gear" : 2
             },
             "racetime":0.324,
             "timestamp":1458141858
          },
          {
             "sensors":{
                "Speed":6.755624,
                "Distance":2004.084717,
                “RPM":1673.264526,
                "gear" : 2
             },
             "racetime":0.556,
             "timestamp":1458141858
          },
```

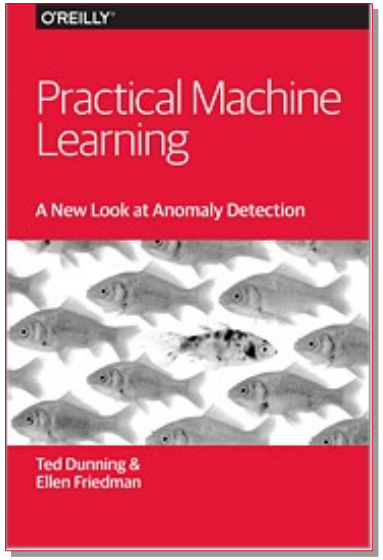
Let's see it work!
(Murphy be praised)

Thank you for coming today!

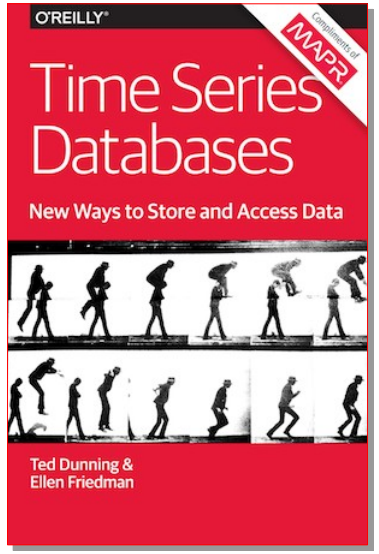
Short Books by Ted Dunning & Ellen Friedman

- Published by O'Reilly in 2014 and 2015
- For sale from Amazon or O'Reilly
- Free e-books currently available courtesy of MapR

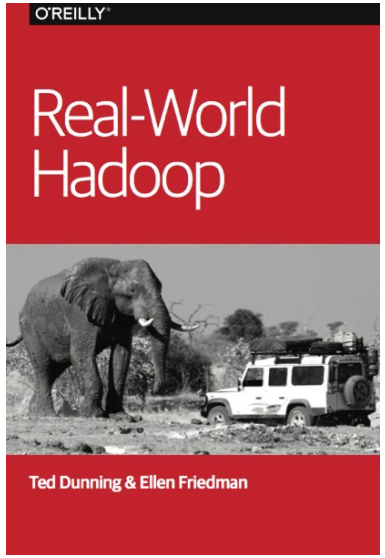
http://bit.ly/recommendation-ebook



http://bit.ly/ebook-anomaly



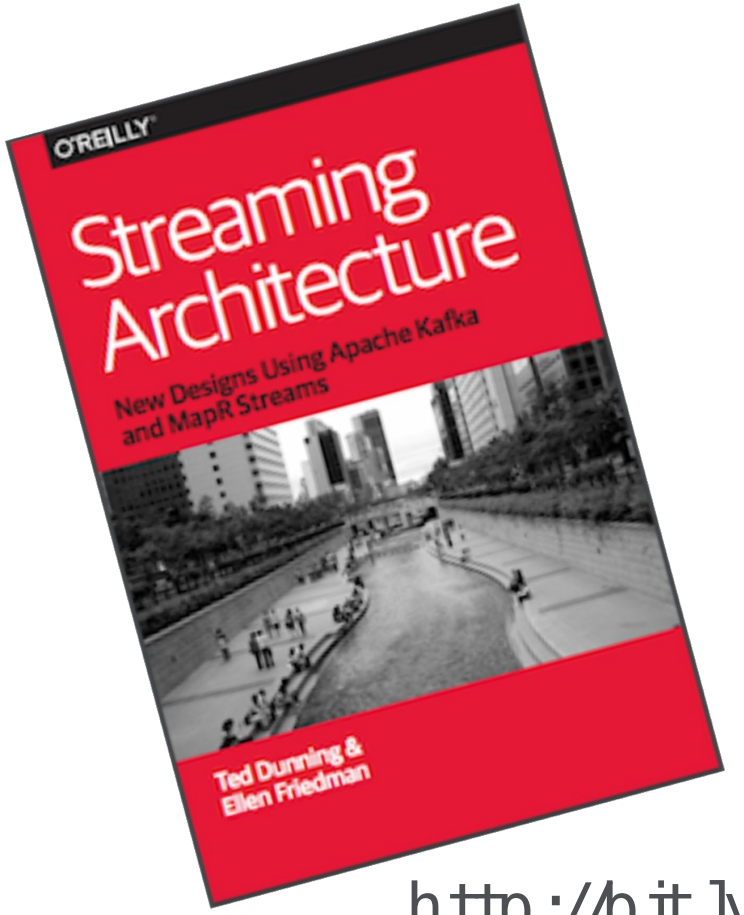
http://bit.ly/mapr-tsdb-ebook



http://bit.ly/ebook-real-world-hadoop

Streaming Architecture

by Ted Dunning and Ellen Friedman © 2016 (published by O'Reilly)



Free copies at book signing tomorrow morning before Tug's talk

http://bit.ly/mapr-ebook-streams

Thank You!

Q&A

Engage with us!

@mapr   maprtech

mapr-technologies   MapR

tdunning@maprtech.com   maprtech