Apache Samza:
Taking stream processing to the next level

Martin Kleppmann — @martinkl

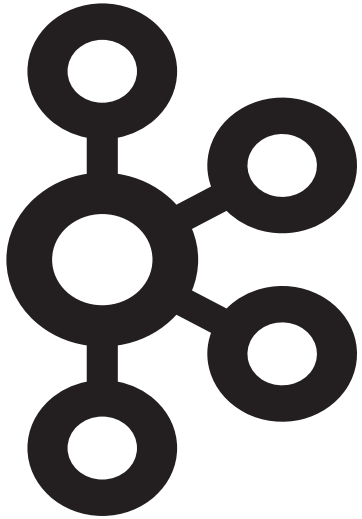
LinkedIn

Martin Kleppmann

Hacker, designer, inventor, entrepreneur

- Co-founded two startups, Rapportive ⇒ LinkedIn
- Committer on Avro & Samza ⇒ Apache
- Writing book on data-intensive apps ⇒ O'Reilly
- martinkl.com | @martinkl

Apache **Kafka**



Apache **Samza**

samza

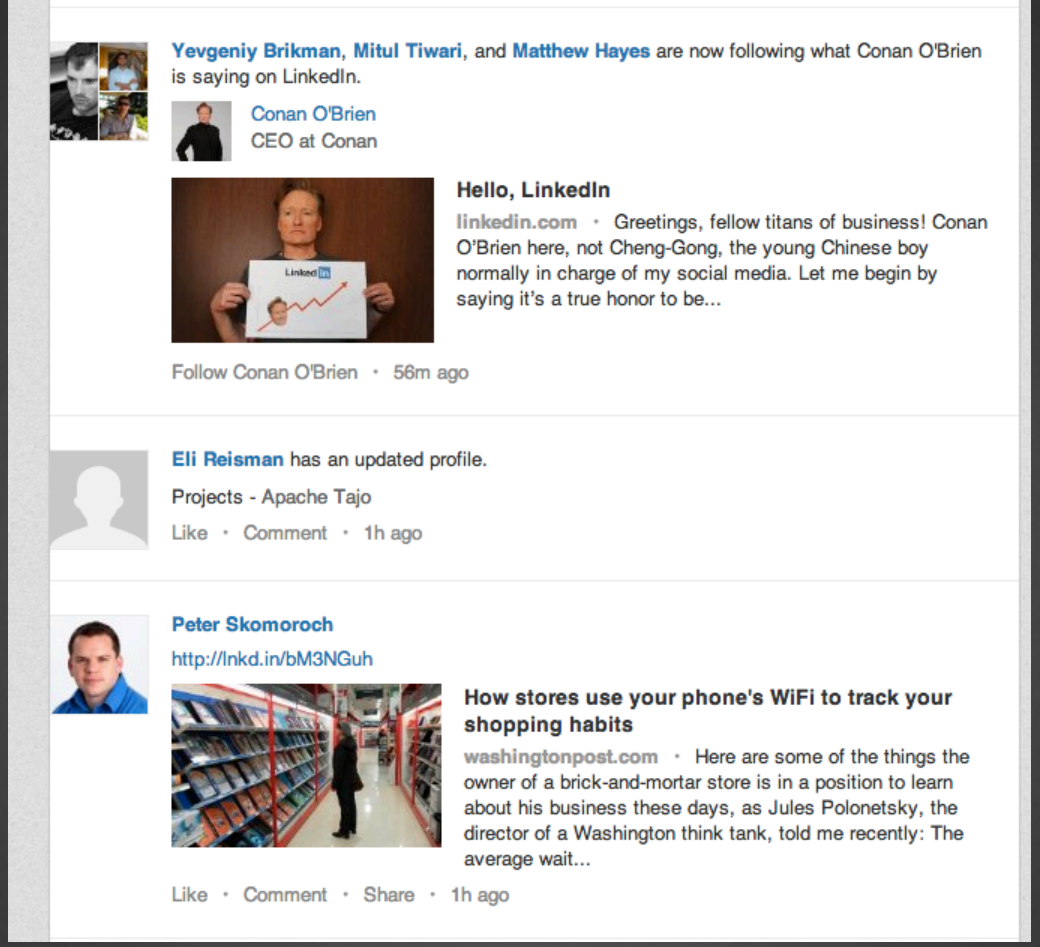
Apache **Kafka**

Credit: Jason Walsh on Flickr
https://www.flickr.com/photos/22882695@N00/2477241427/

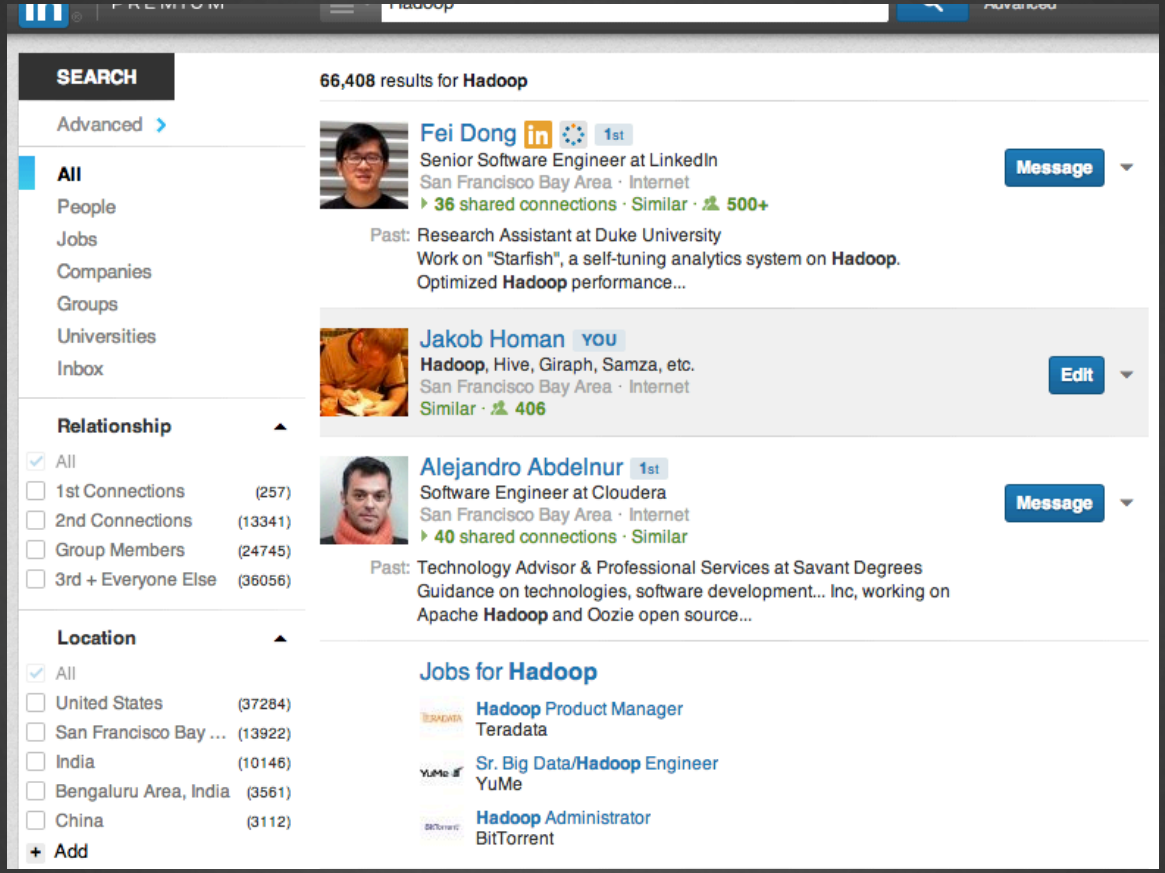
Apache **Samza**

Credit: Lucas Richarz on Flickr
https://www.flickr.com/photos/22057420@N00/5690285549/

Things we would like to do (better)



Provide timely, relevant updates to your newsfeed

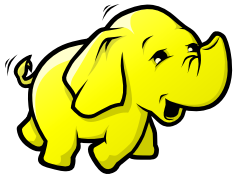


Update search results with new information as it appears

“Real-time” analysis of logs and metrics

Tools?

REST Kafka & Samza

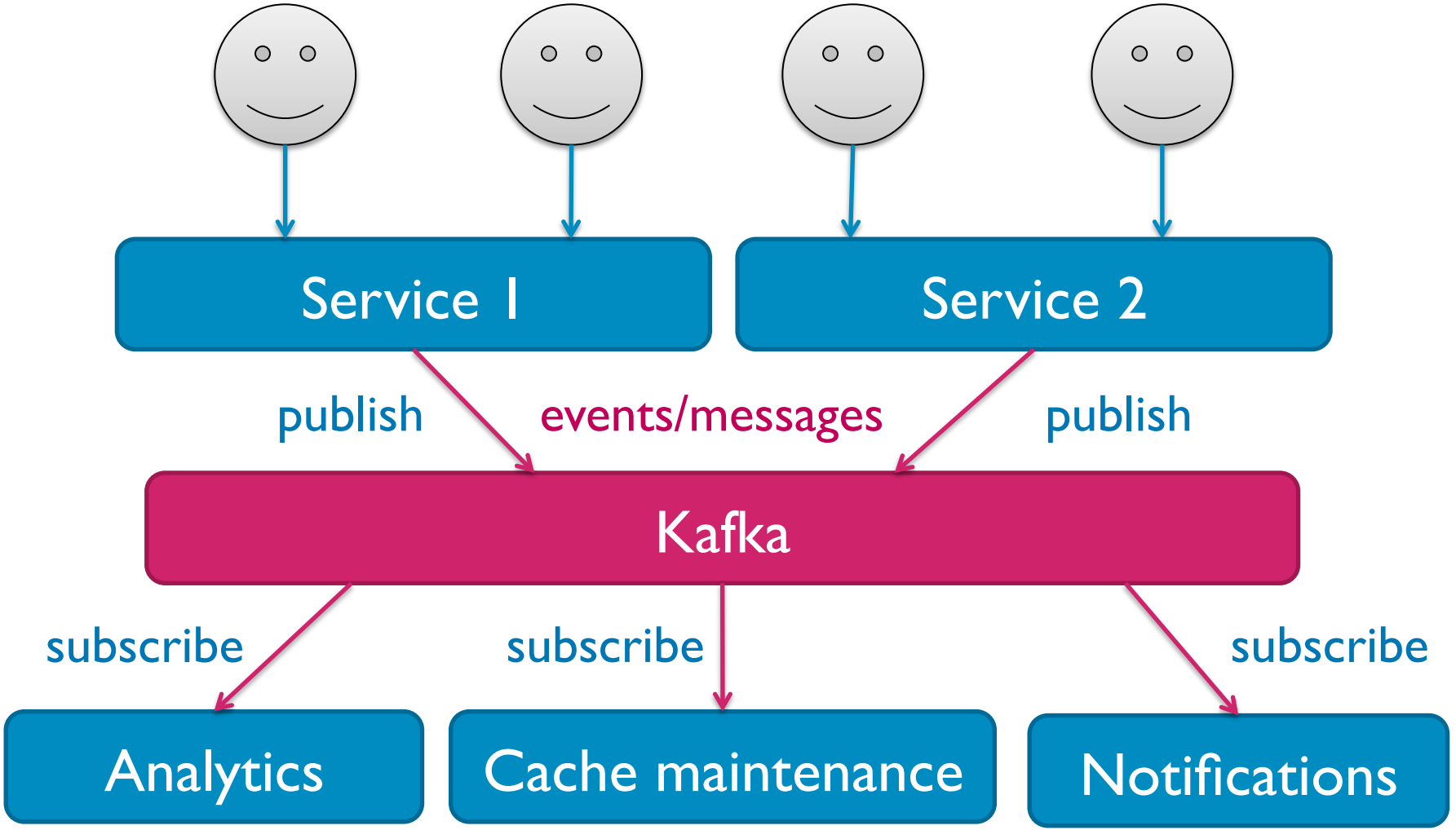


Response latency

Milliseconds to minutes
Loosely coupled

Synchronous
Closely coupled

Hours to days
Loosely coupled


Service 1
Service 2
publish
events/messages
publish
Kafka
subscribe
subscribe
subscribe
Analytics
Cache maintenance
Notifications

Publish / subscribe

- Event / message = “something happened”
  - Tracking: User **x** clicked **y** at time **z**
  - Data change: Key **x**, old value **y**, set to new value **z**
  - Logging: Service **x** threw exception **y** in request **z**
  - Metrics: Machine **x** had free memory **y** at time **z**

- Many independent consumers
- High throughput (millions msgs/sec)
- Fairly low latency (a few ms)

Kafka at LinkedIn

- 350+ Kafka brokers
- 8,000+ topics
- 140,000+ Partitions

- 278 Billion messages/day
- 49 TB/day in
- 176 TB/day out

- Peak Load
  - 4.4 Million messages per second
  - 6 Gigabits/sec Inbound
  - 21 Gigabits/sec Outbound

Samza API: processing messages

```
public interface StreamTask {
    void process(IncomingMessageEnvelope envelope,
                 MessageCollector collector,
                 TaskCoordinator coordinator);
}
```

getKey(), getMsg()

sendMsg(topic, key, value)

commit(), shutdown()

Familiar ideas from MR/Pig/Cascading/…

- **Filter** records matching condition
- **Map** record ⇒ func(record)
- **Join** two/more datasets by key
- **Group** records with the same value in field
- **Aggregate** records within the same group
- **Pipe** job 1's output ⇒ job 2's input

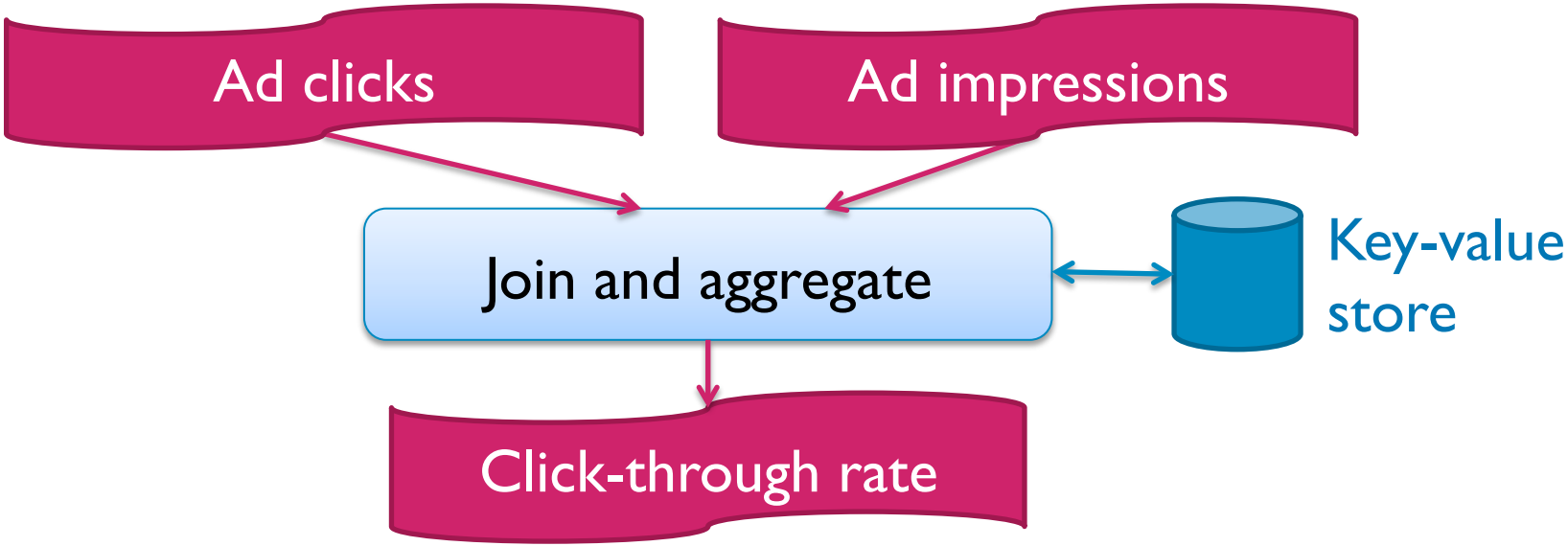
- MapReduce assumes fixed dataset.
  Can we adapt this to unbounded streams?

Operations on streams

- **Filter** records matching condition ✔ **easy**
- **Map** record ⇒ func(record) ✔ **easy**
- **Join** two/more datasets by key
  …within time window, need buffer
- **Group** records with the same value in field
  …within time window, need buffer
- **Aggregate** records within the same group
  ✔ **ok… when do you emit result?**
- **Pipe** job 1's output ⇒ job 2's input
  ✔ **ok… but what about faults?**

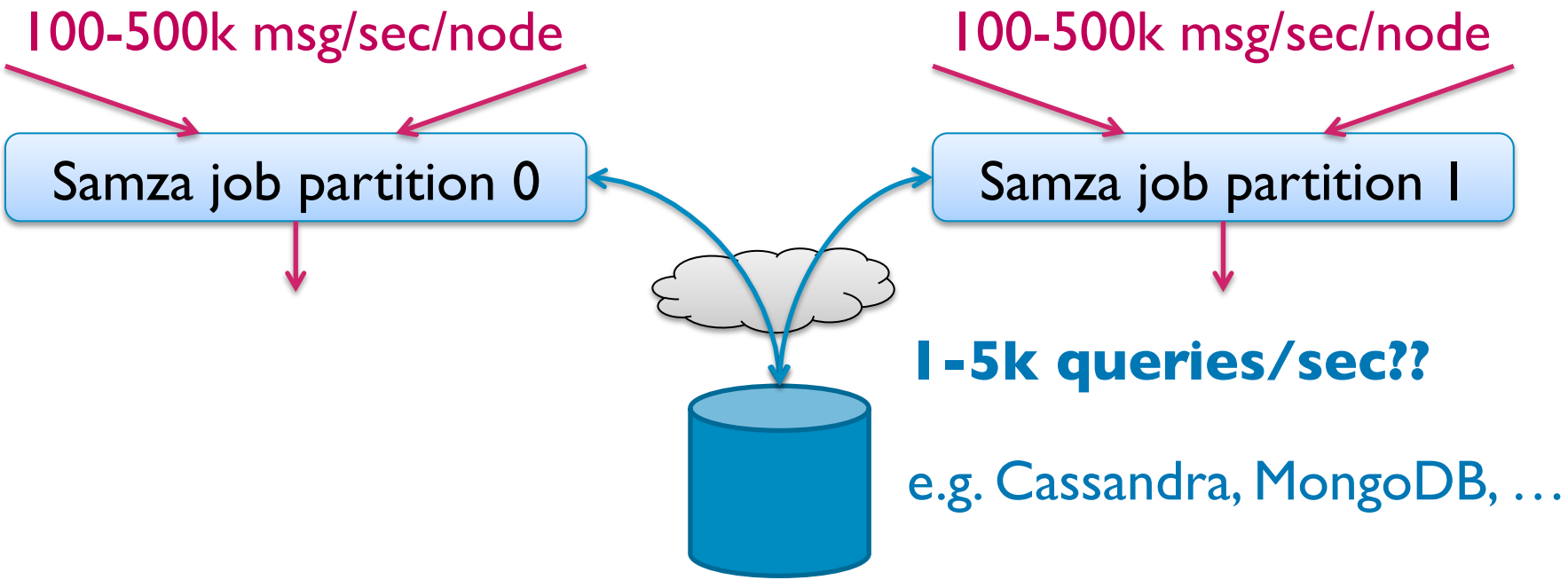
Stateful stream processing
(join, group, aggregate)

Joining streams requires state

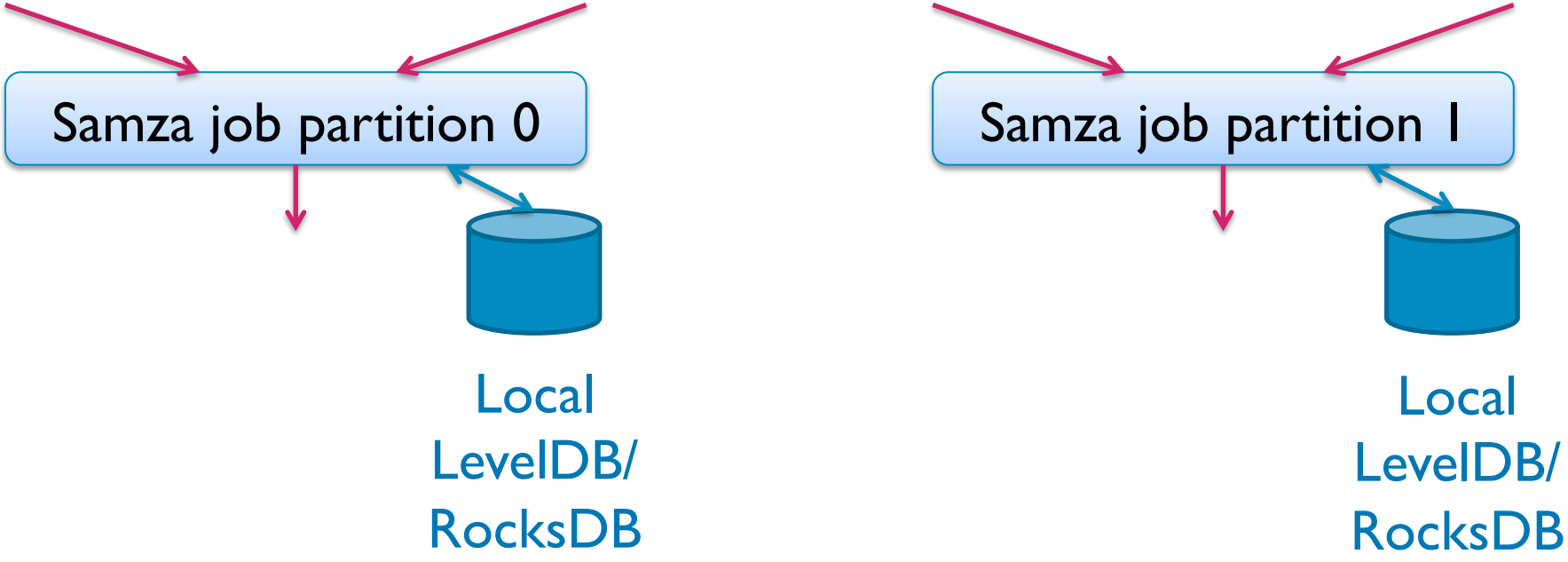


- User goes to lunch ⇒ click long after impression
- Queue backlog ⇒ click before impression
- "Window join"

Remote state or local state?



Remote state or local state?

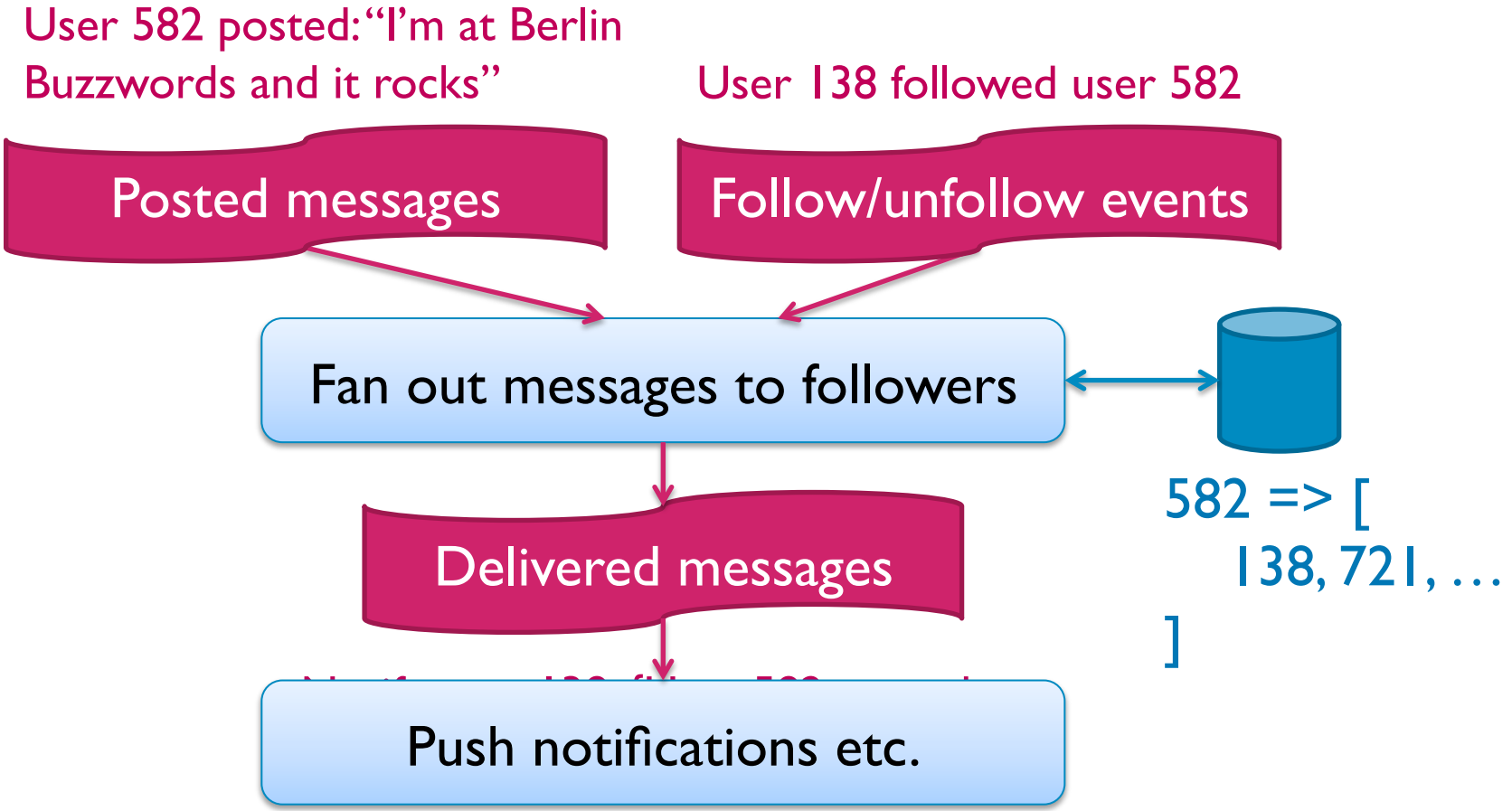


Another example: Newsfeed & following

- User 138 followed user 582
- User 463 followed user 536
- User 582 posted: "I'm at Berlin Buzzwords and it rocks"
- User 507 unfollowed user 115
- User 536 posted: "Nice weather today, going for a walk"
- User 981 followed user 575

- Expected output: "inbox" (newsfeed) for each user

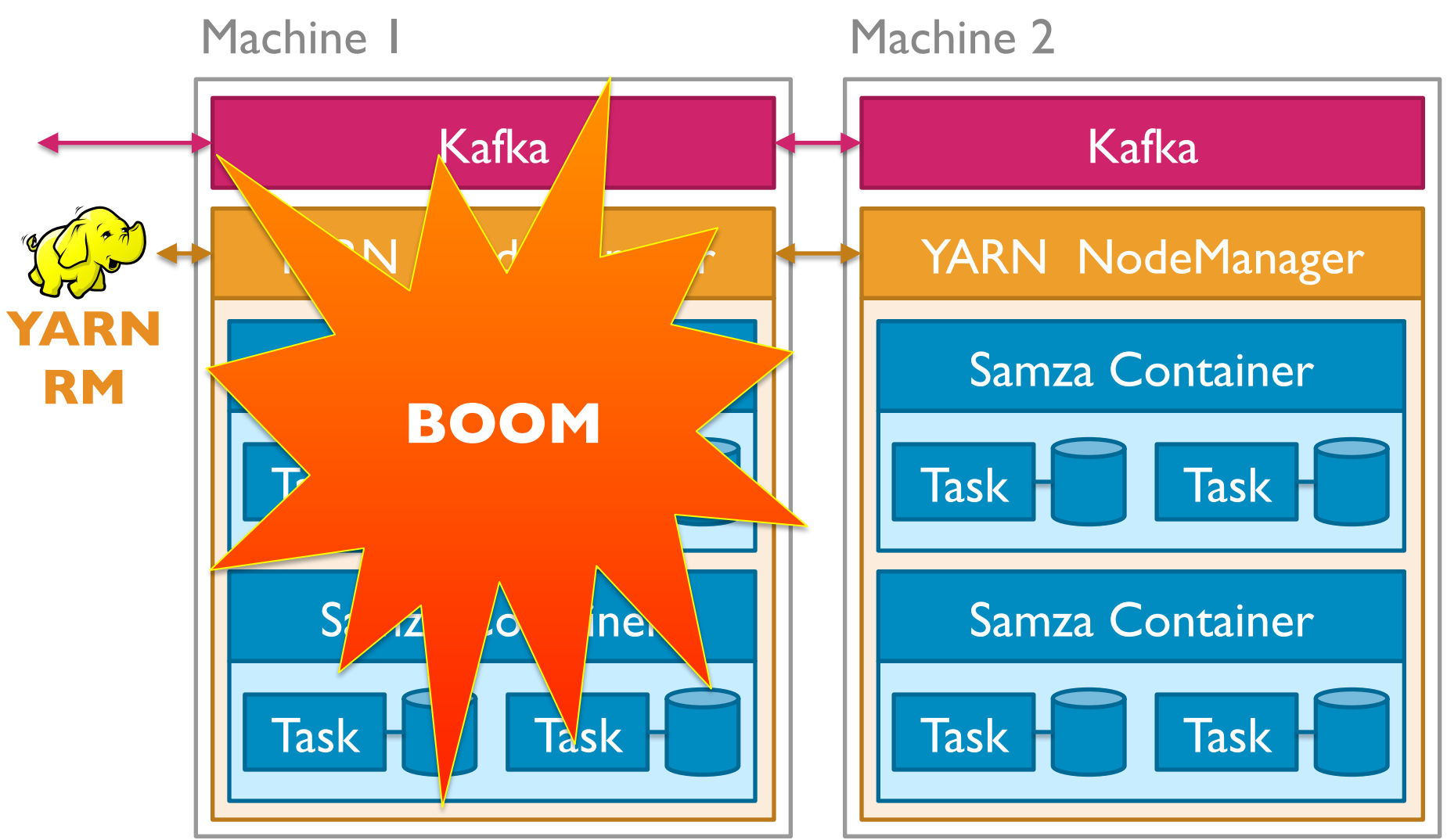
Newsfeed & following

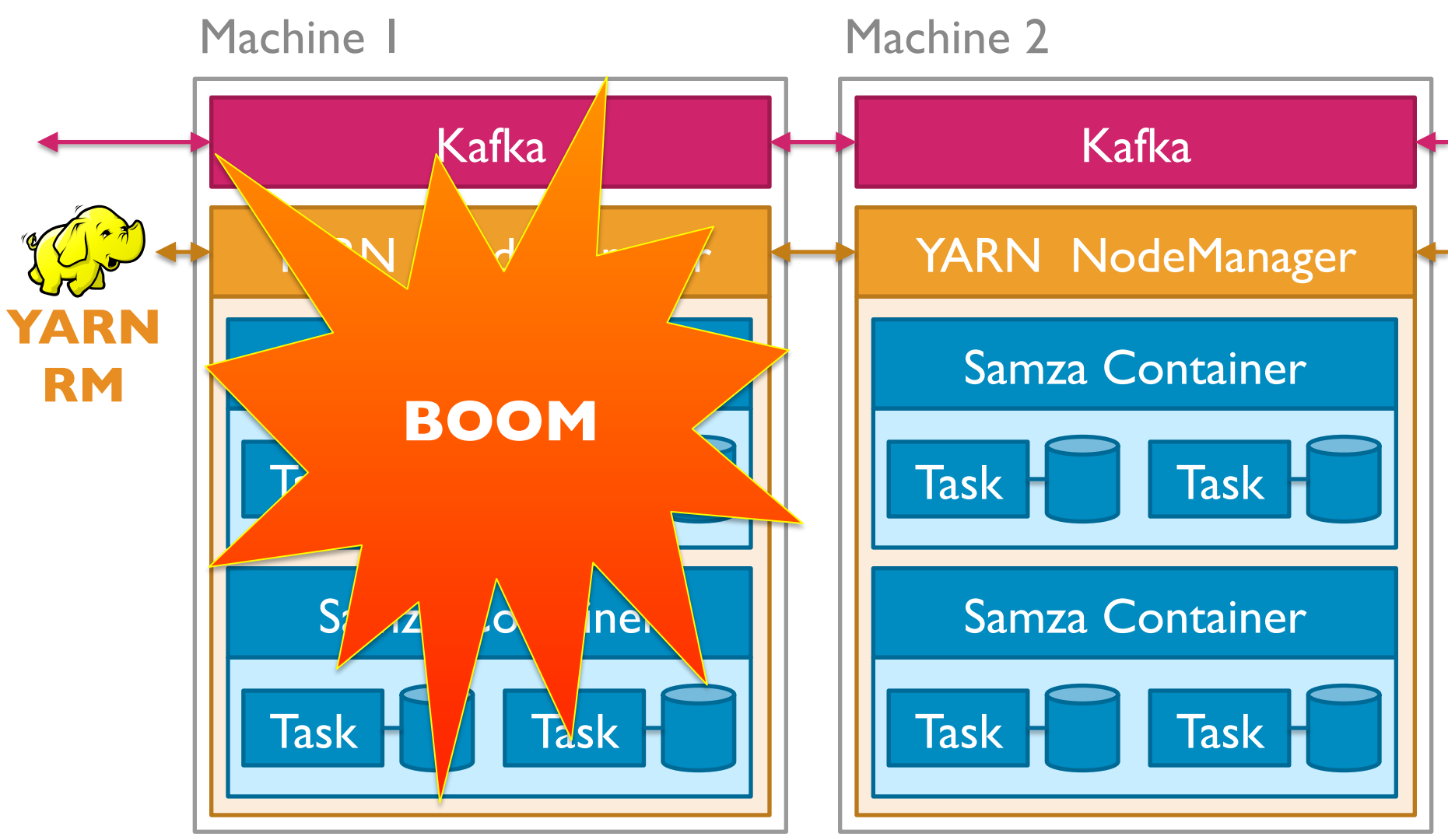


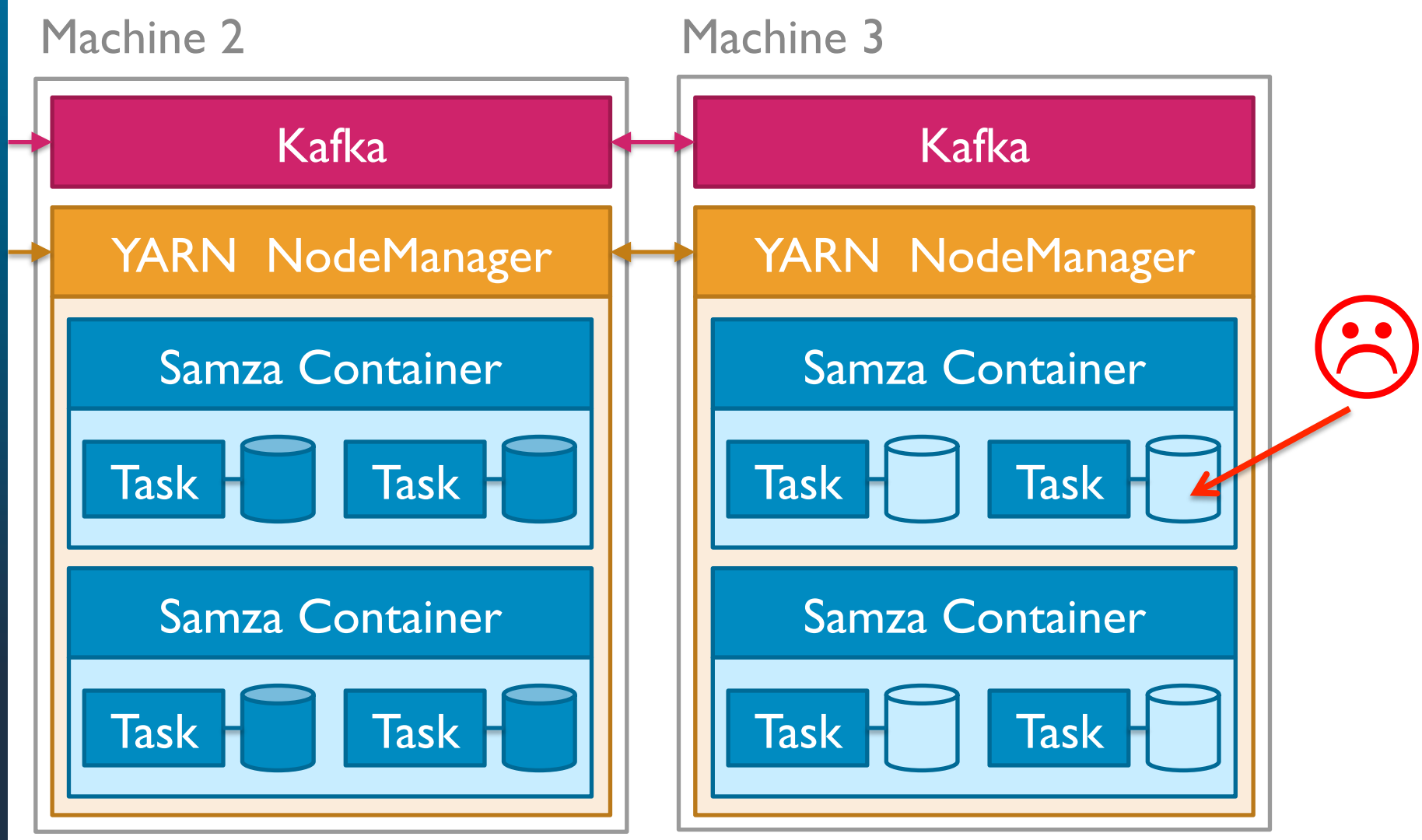
Local state:

Bring **computation** and **data**
together in one place

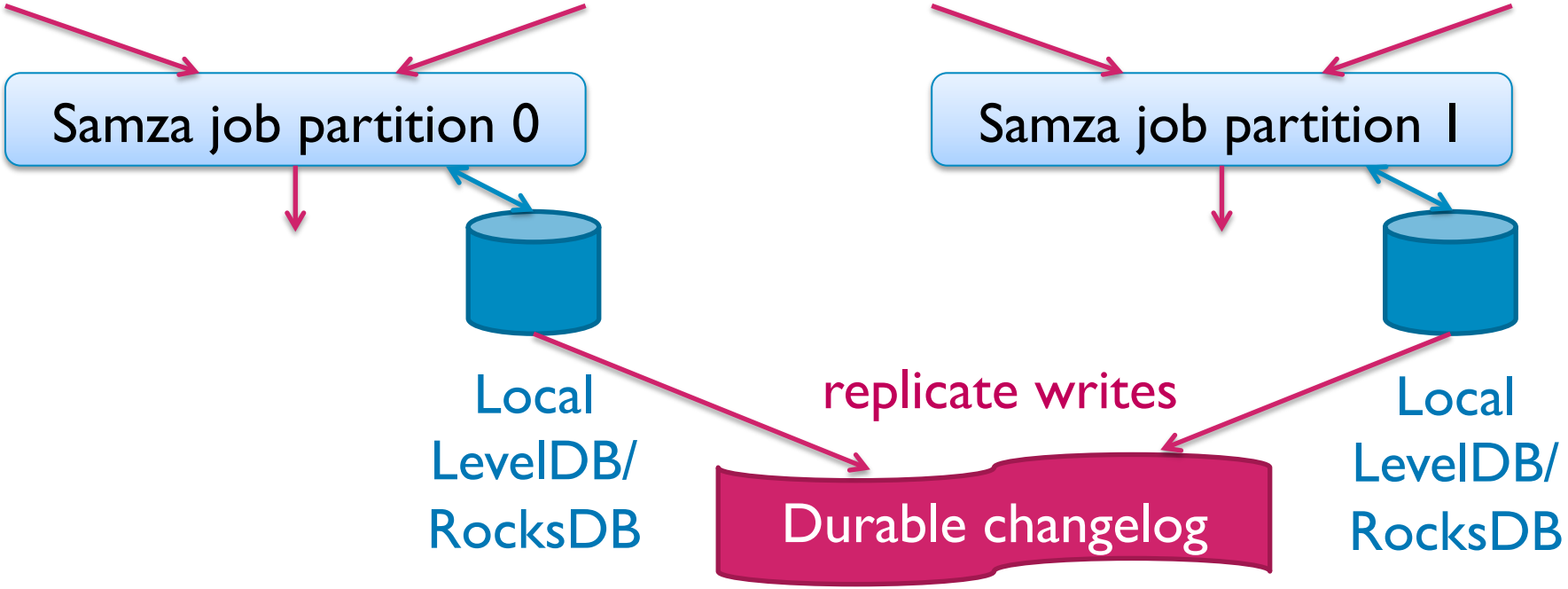
Fault tolerance

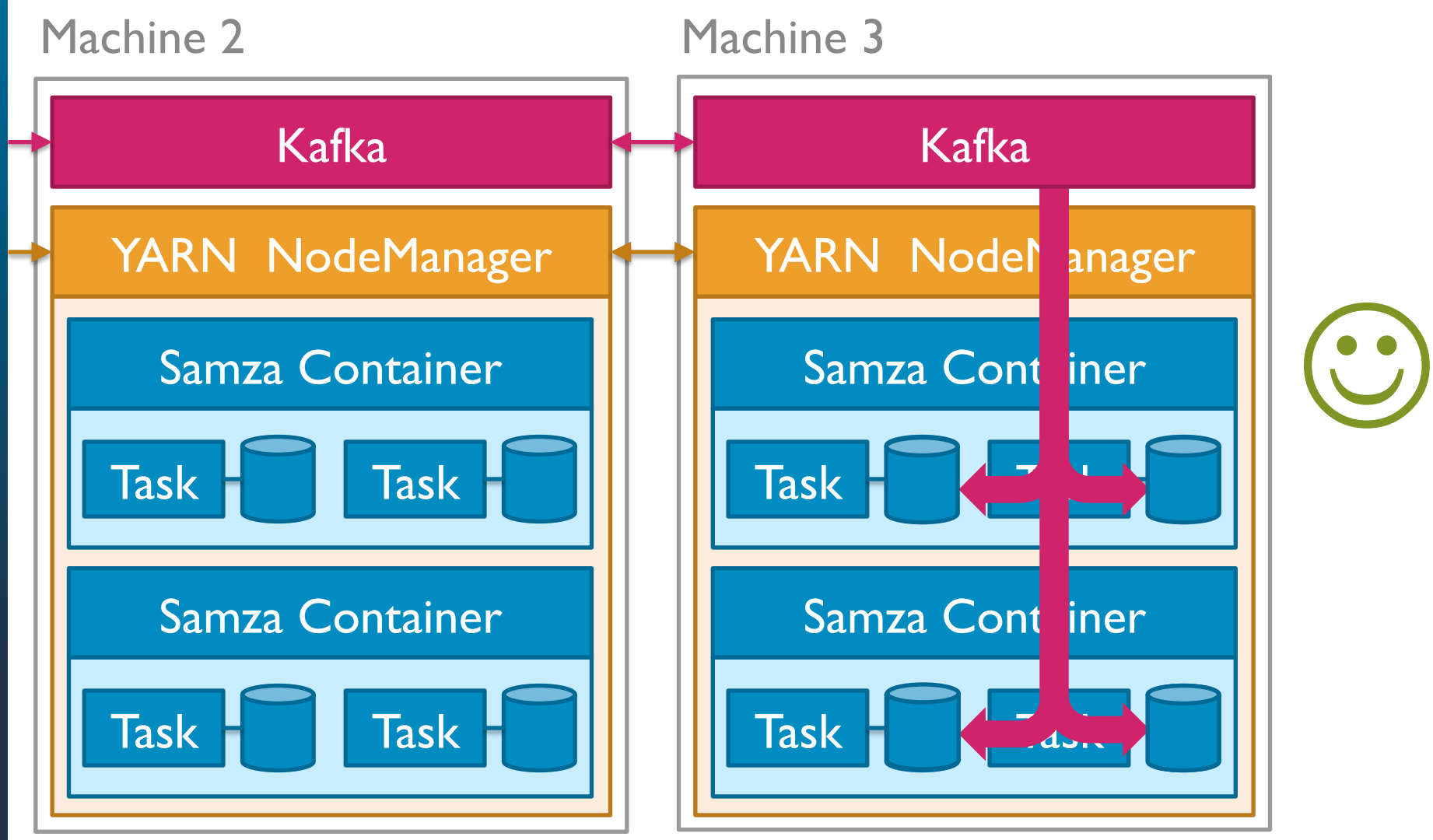

Machine 1
Machine 2
Kafka
Kafka
YARN NodeManager
YARN
RM
BOOM
Samza Container
Task
Task
Samza Container
Task
Task
Task
Task


Machine 1
Machine 2
Kafka
Kafka
YARN
RM
YARN NodeManager
BOOM
Samza Container
Task
Task
Samza Container
Task
Task
Task
Task


Machine 2
Machine 3
Kafka
Kafka
YARN NodeManager
YARN NodeManager
Samza Container
Task
Task
Samza Container
Task
Task
Samza Container
Task
Task
Samza Container
Task
Task

Fault-tolerant local state




Machine 2
Machine 3
Kafka
Kafka
YARN NodeManager
Samza Container
Task
Task
Samza Container
Task
Task
Task
Task

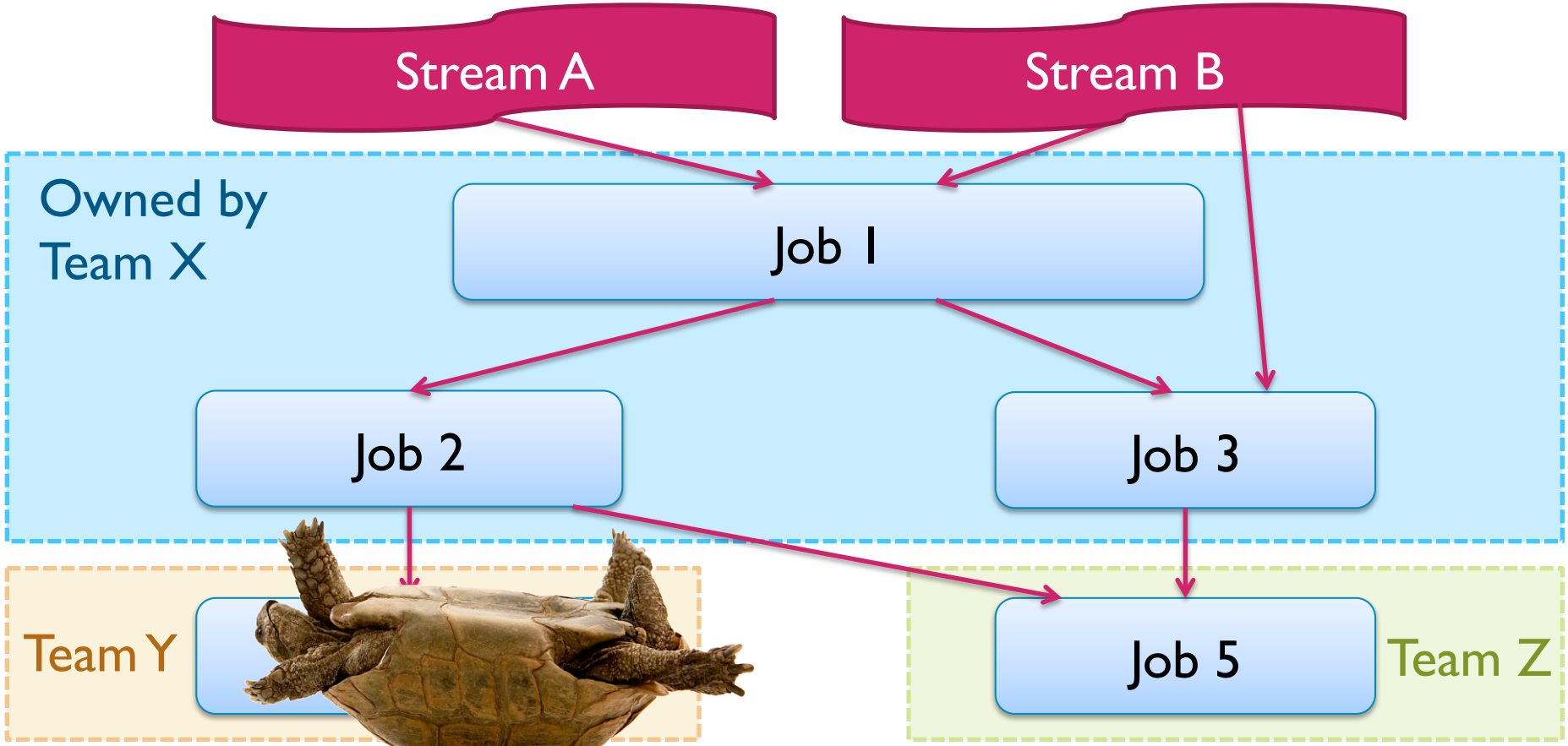
Samza's fault-tolerant local state

- Embedded key-value: **very fast**
- Machine dies ⇒ local key-value store is lost

- Solution: replicate all writes to Kafka!
- Machine dies ⇒ restart on another machine
- Restore key-value store from changelog
- Changelog compaction in the background (Kafka 0.8.1)

When things go slow…

Cascades of jobs



Consumer goes slow

Drop data

No thanks ☹

Backpressure

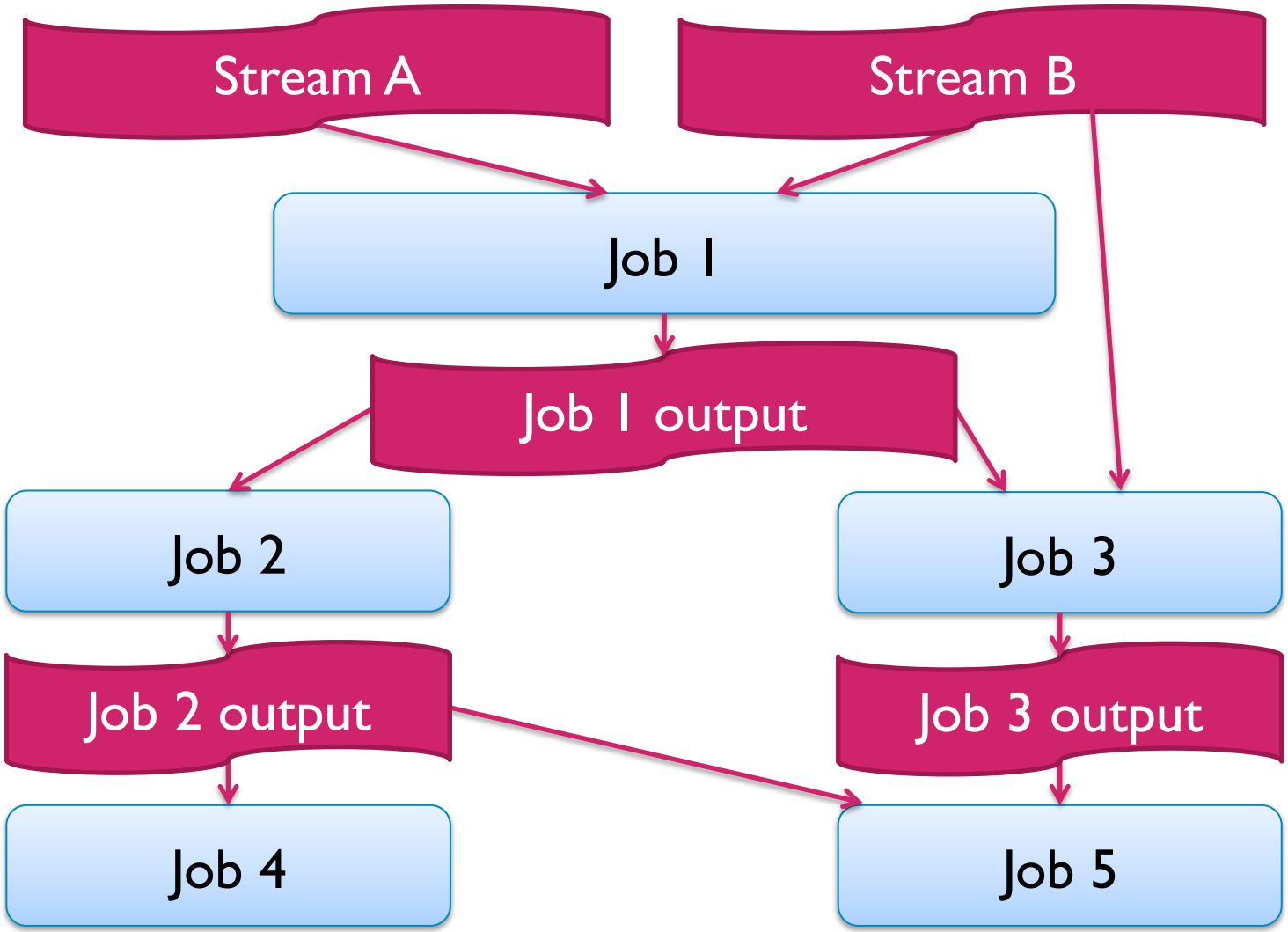
Other jobs grind to a halt ☹

Queue up

Run out of memory ☹

Spill to disk

Oh wait… Kafka does this anyway!


Stream A
Stream B
Job 1
Job 1 output
Job 2
Job 3
Job 2 output
Job 3 output
Job 4
Job 5

Samza always writes
job output to **Kafka**
MapReduce always writes
job output to **HDFS**

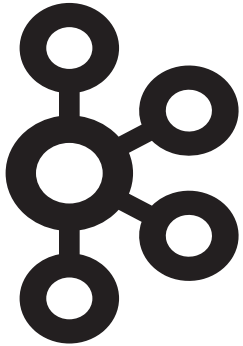
Every job output is a named stream

- **Open:** Anyone can consume it
- **Robust:** If a consumer goes slow, nobody else is affected
- **Durable:** Tolerates machine failure
- **Debuggable:** Just look at it
- **Scalable:** Clean interface between teams
- **Clean:** loose coupling between jobs

Recap

Problem	Solution
Need to buffer job output for downstream consumers	Write it to Kafka!
Need to make local state store fault-tolerant	Write it to Kafka!
Need to checkpoint job state for recovery	Write it to Kafka!

Apache **Kafka**



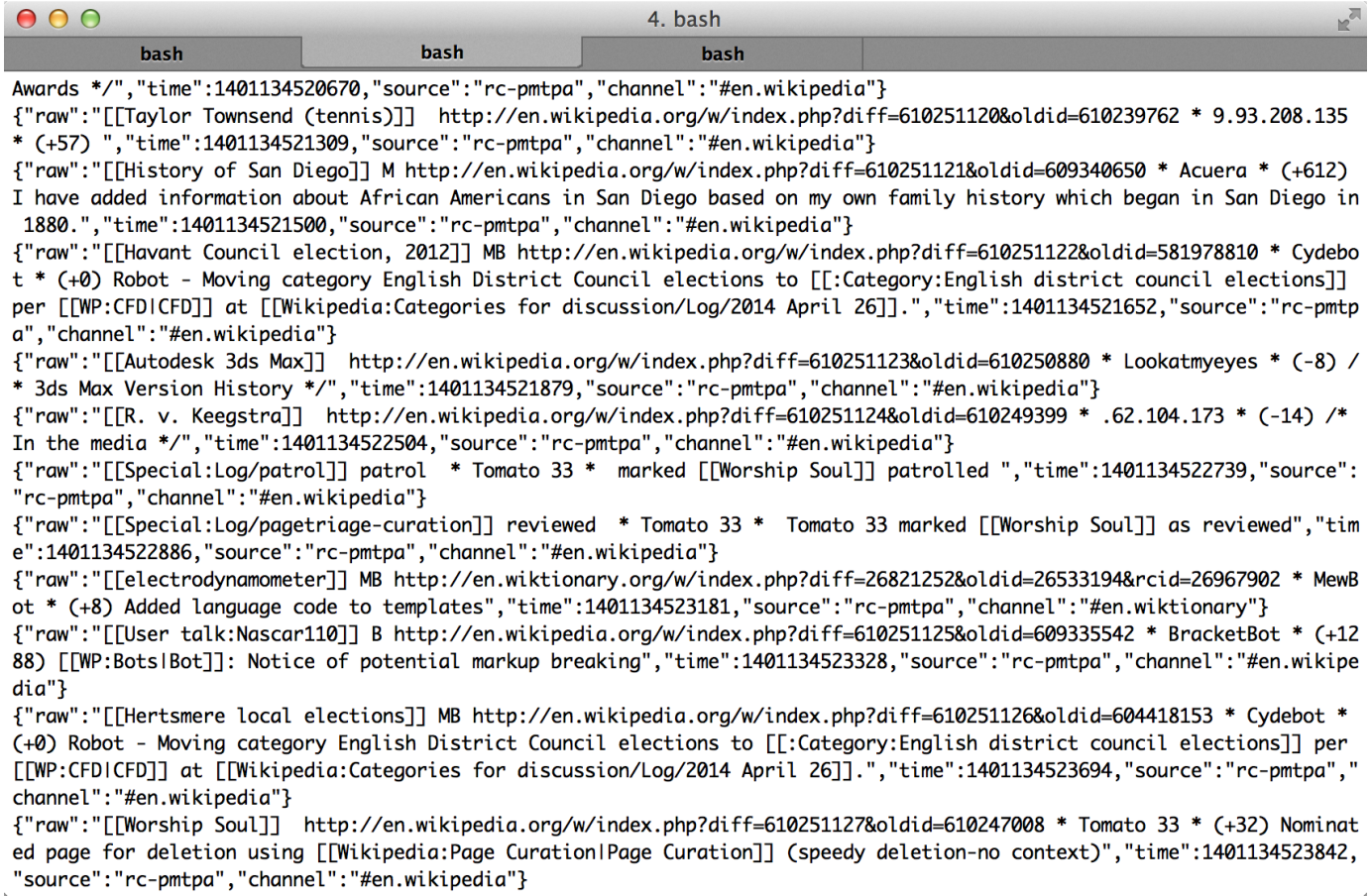
kafka.apache.org

Apache **Samza**

samza

samza.incubator.apache.org

Hello Samza (try Samza in 5 mins)




```
Awards */","time":1401134520670,"source":"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[Taylor Townsend (tennis)]]  http://en.wikipedia.org/w/index.php?diff=610251120&oldid=610239762 * 9.93.208.135
* (+57) ","time":1401134521309,"source":"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[History of San Diego]] M http://en.wikipedia.org/w/index.php?diff=610251121&oldid=609340650 * Acuera * (+612)
I have added information about African Americans in San Diego based on my own family history which began in San Diego in
 1880.","time":1401134521500,"source":"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[Havant Council election, 2012]] MB http://en.wikipedia.org/w/index.php?diff=610251122&oldid=581978810 * Cydebo
t * (+0) Robot - Moving category English District Council elections to [[:Category:English district council elections]]
per [[WP:CFD|CFD]] at [[Wikipedia:Categories for discussion/Log/2014 April 26]].","time":1401134521652,"source":"rc-pmtp
a","channel":"#en.wikipedia"}
{"raw":"[[Autodesk 3ds Max]]  http://en.wikipedia.org/w/index.php?diff=610251123&oldid=610250880 * Lookatmyeyes * (-8) /
* 3ds Max Version History */","time":1401134521879,"source":"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[R. v. Keegstra]]  http://en.wikipedia.org/w/index.php?diff=610251124&oldid=610249399 * .62.104.173 * (-14) /*
In the media */","time":1401134522504,"source":"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[Special:Log/patrol]] patrol  * Tomato 33 *  marked [[Worship Soul]] patrolled ","time":1401134522739,"source":
"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[Special:Log/pagetriage-curation]] reviewed  * Tomato 33 *  Tomato 33 marked [[Worship Soul]] as reviewed","tim
e":1401134522886,"source":"rc-pmtpa","channel":"#en.wikipedia"}
{"raw":"[[electrodynamometer]] MB http://en.wiktionary.org/w/index.php?diff=26821252&oldid=26533194&rcid=26967902 * MewB
ot * (+8) Added language code to templates","time":1401134523181,"source":"rc-pmtpa","channel":"#en.wiktionary"}
{"raw":"[[User talk:Nascar110]] B http://en.wikipedia.org/w/index.php?diff=610251125&oldid=609335542 * BracketBot * (+12
88) [[WP:Bots|Bot]]: Notice of potential markup breaking","time":1401134523328,"source":"rc-pmtpa","channel":"#en.wikipe
dia"}
{"raw":"[[Hertsmere local elections]] MB http://en.wikipedia.org/w/index.php?diff=610251126&oldid=604418153 * Cydebot *
(+0) Robot - Moving category English District Council elections to [[:Category:English district council elections]] per
[[WP:CFD|CFD]] at [[Wikipedia:Categories for discussion/Log/2014 April 26]].","time":1401134523694,"source":"rc-pmtpa","
channel":"#en.wikipedia"}
{"raw":"[[Worship Soul]]  http://en.wikipedia.org/w/index.php?diff=610251127&oldid=610247008 * Tomato 33 * (+32) Nominat
ed page for deletion using [[Wikipedia:Page Curation|Page Curation]] (speedy deletion-no context)","time":1401134523842,
"source":"rc-pmtpa","channel":"#en.wikipedia"}
```

Thank you!

Samza:

- Getting started: *samza.incubator.apache.org*
- Underlying thinking: *bit.ly/jay_on_logs*
- Start contributing: *bit.ly/samza_newbie_issues*

Me:

- Twitter: *@martinkl*
- Blog: *martinkl.com*