





SKA AFRICA
SQUARE KILOMETRE ARRAY

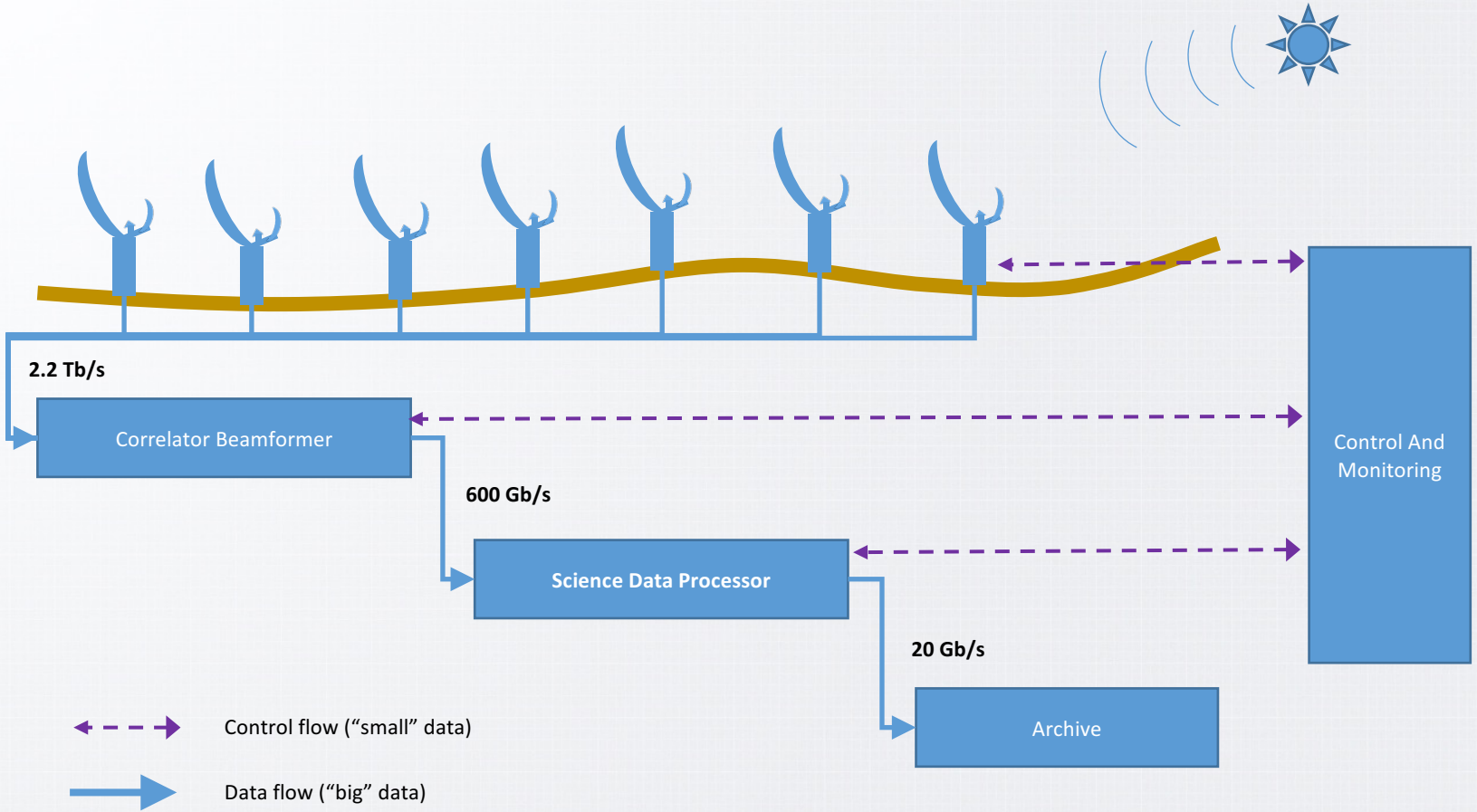
MeerKAT & Mesos

ORCHESTRATING MEERKAT'S DISTRIBUTED SCIENCE DATA PROCESSING PIPELINES

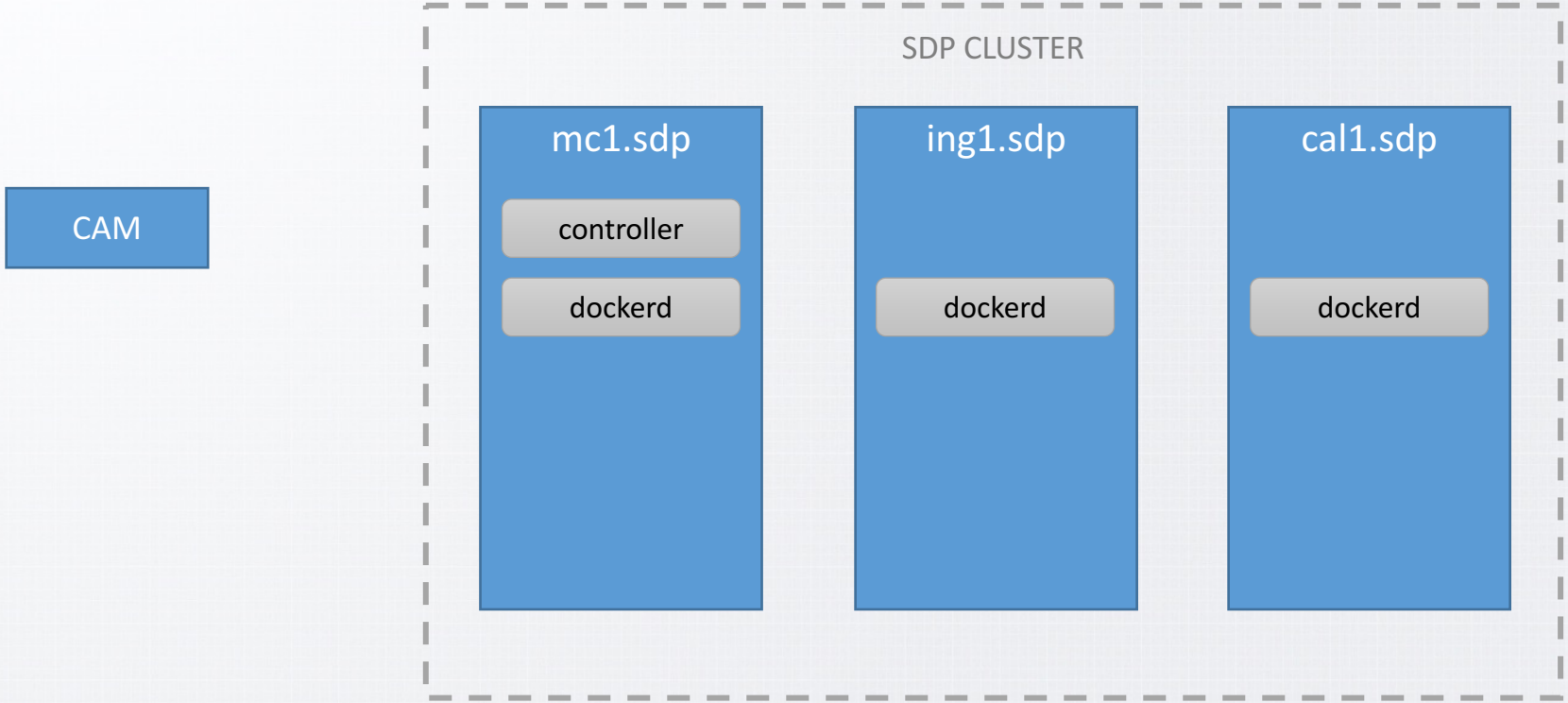
ICALEPCS2017
Barcelona · Spain, October 8-13 · Palau de Congressos de Catalunya

PRESENTER: Anton Joubert
CO-AUTHOR: Bruce Merry

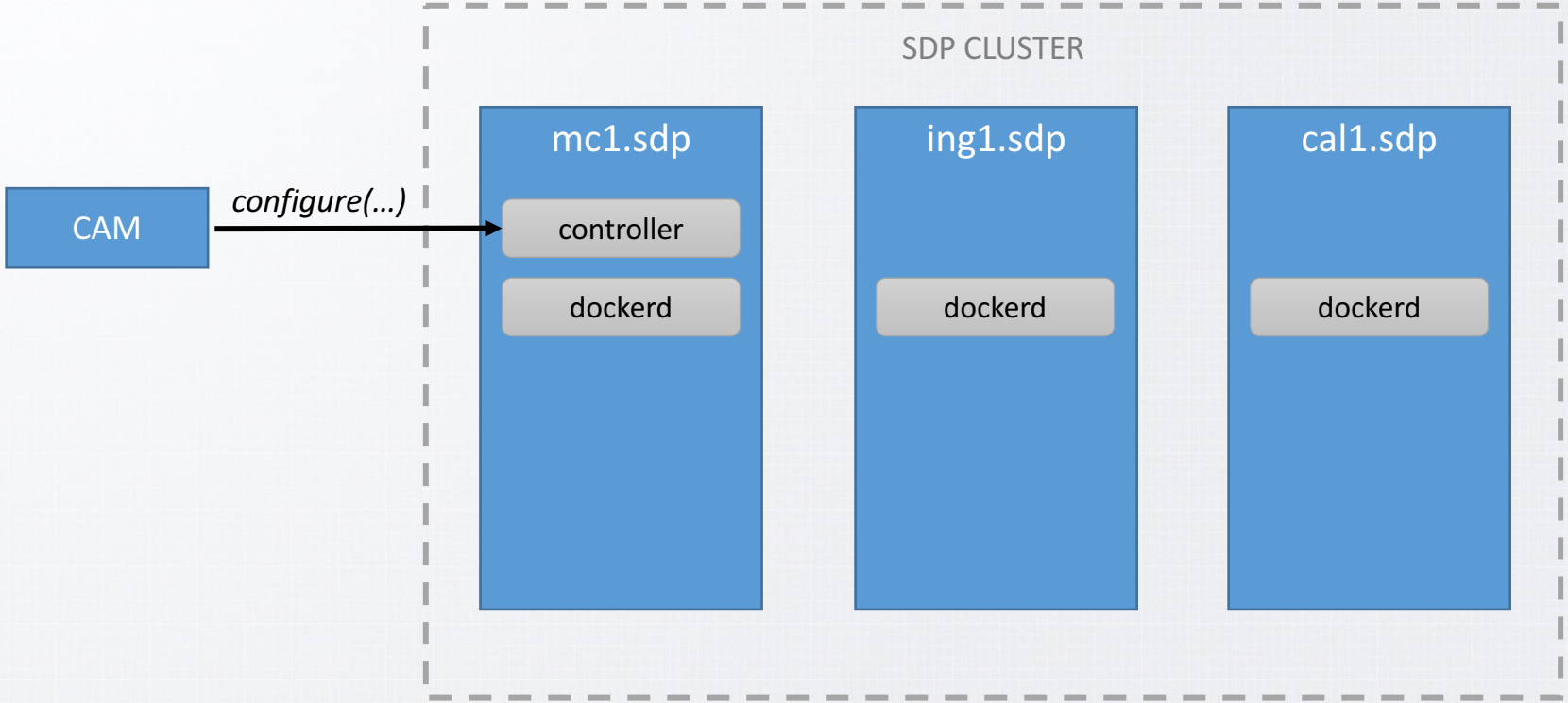
THBPA04


2.2 Tb/s
Correlator Beamformer
600 Gb/s
Science Data Processor
20 Gb/s
Archive
Control And Monitoring
Control flow (“small” data)
Data flow (“big” data)

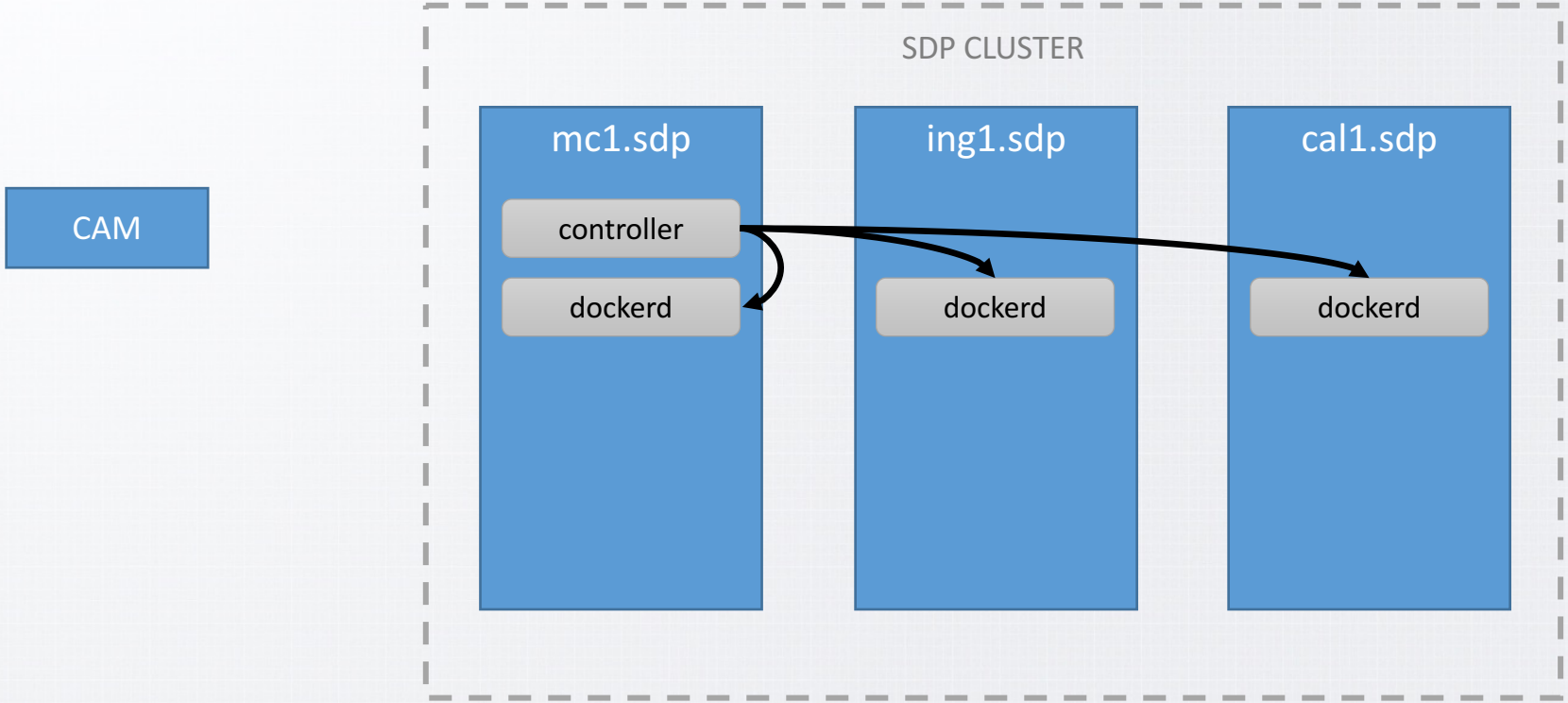
Where we were



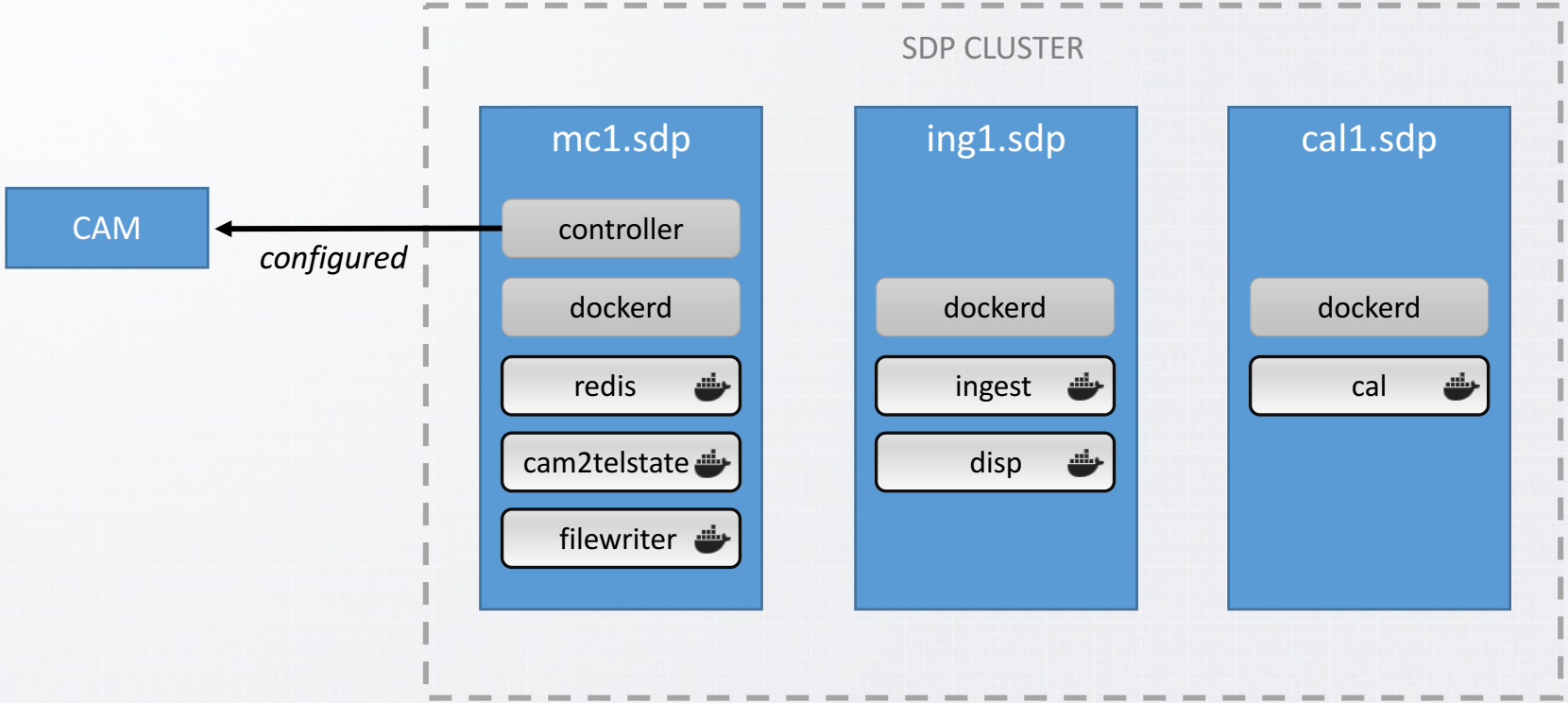
Where we were



Where we were



Where we were

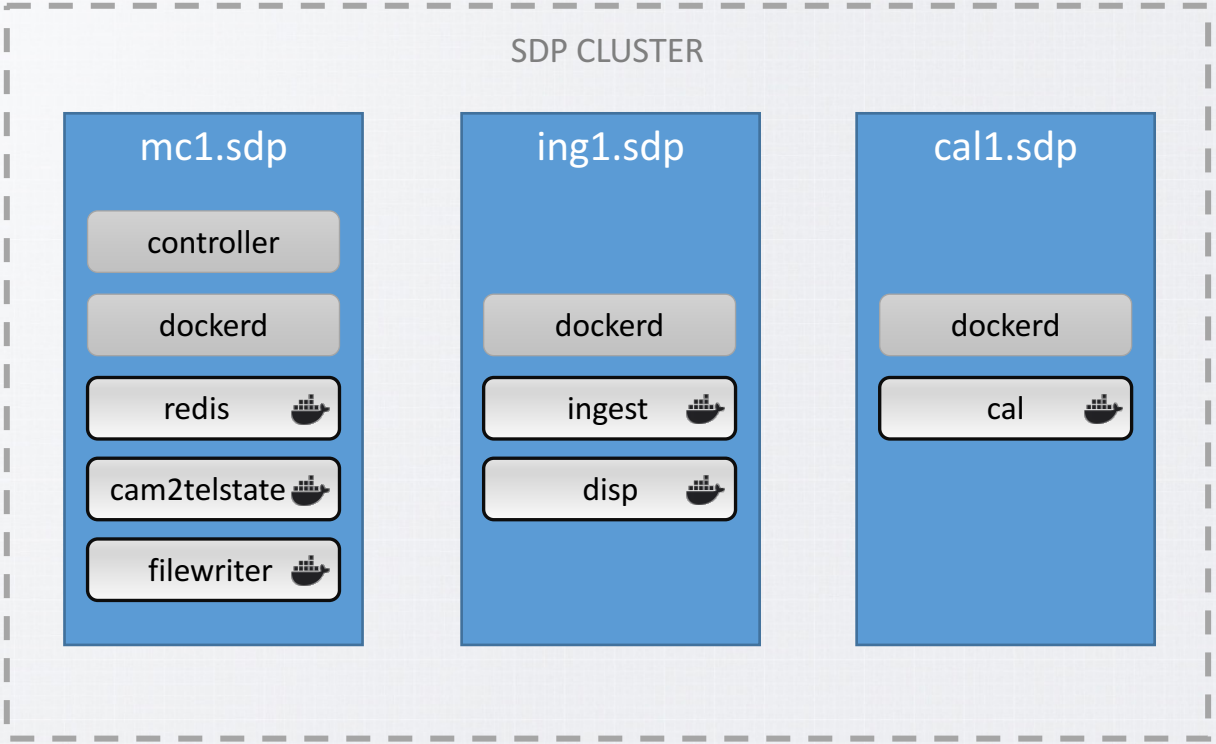


Where we were

Problems:

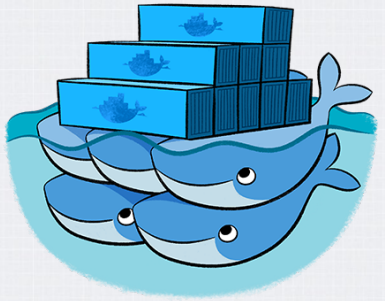
- Manual reconfiguration required if a host unavailable
- Host utilisation not well balanced
- Does not scale

How can we automate this?



Container orchestration tools

Docker Swarm

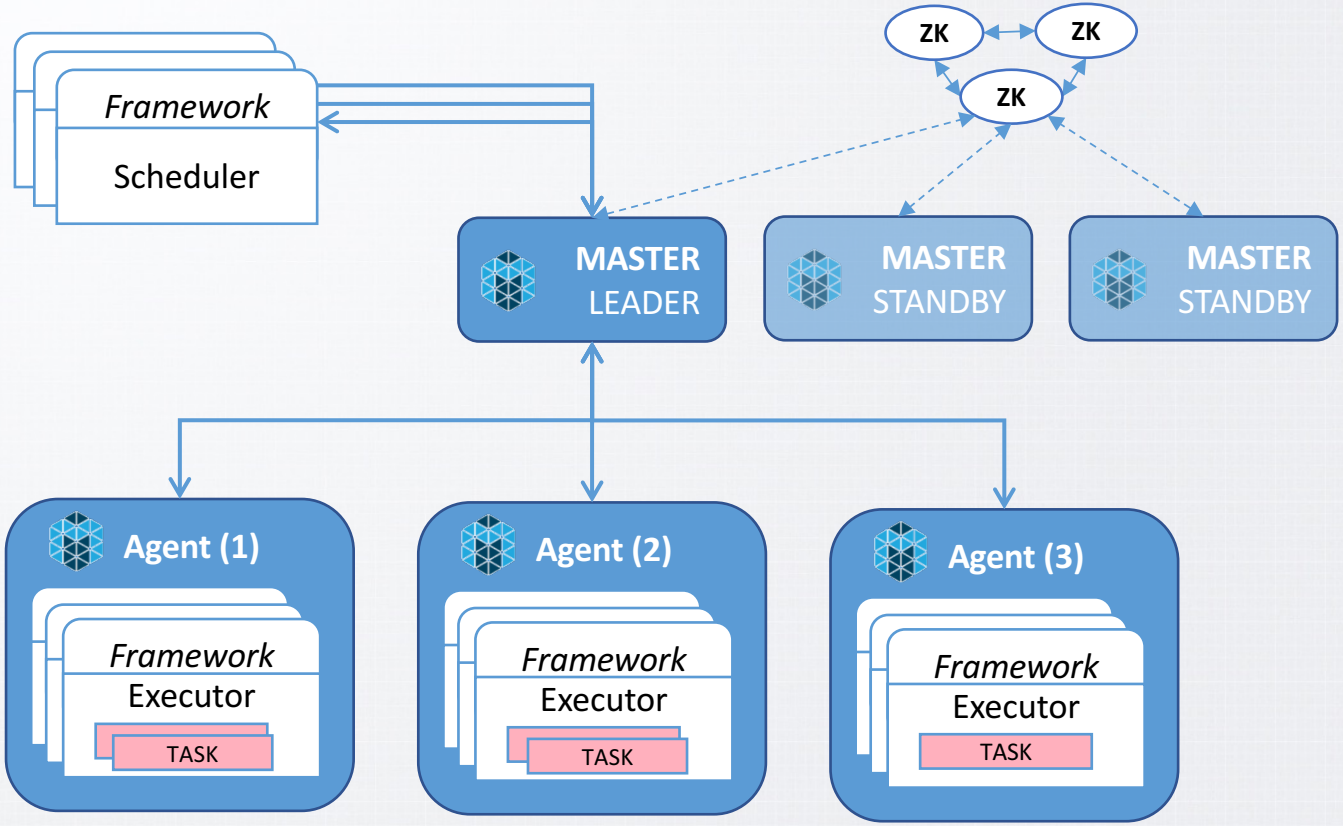


kubernetes

HashiCorp
Nomad

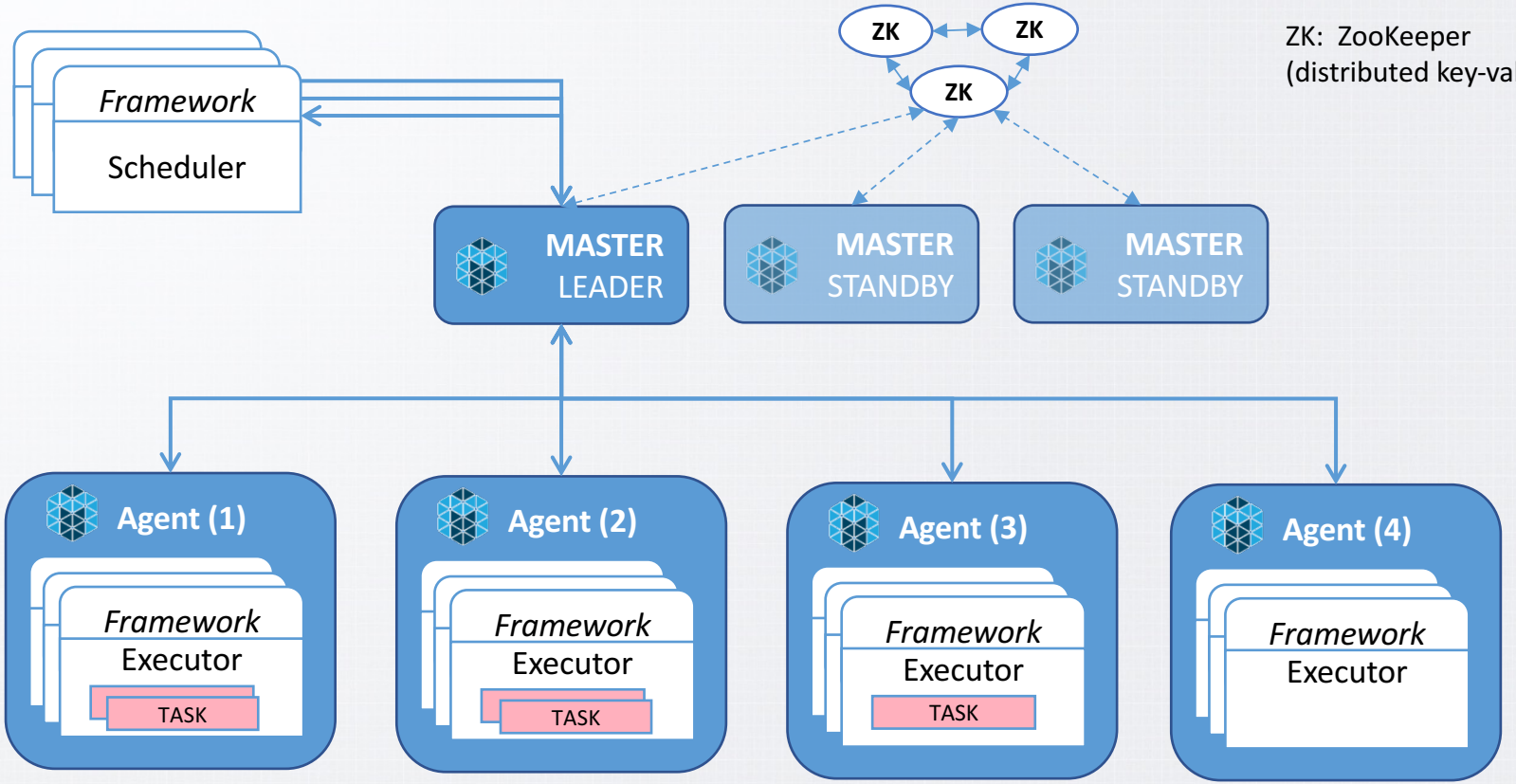
And many more…

Mesos Architecture



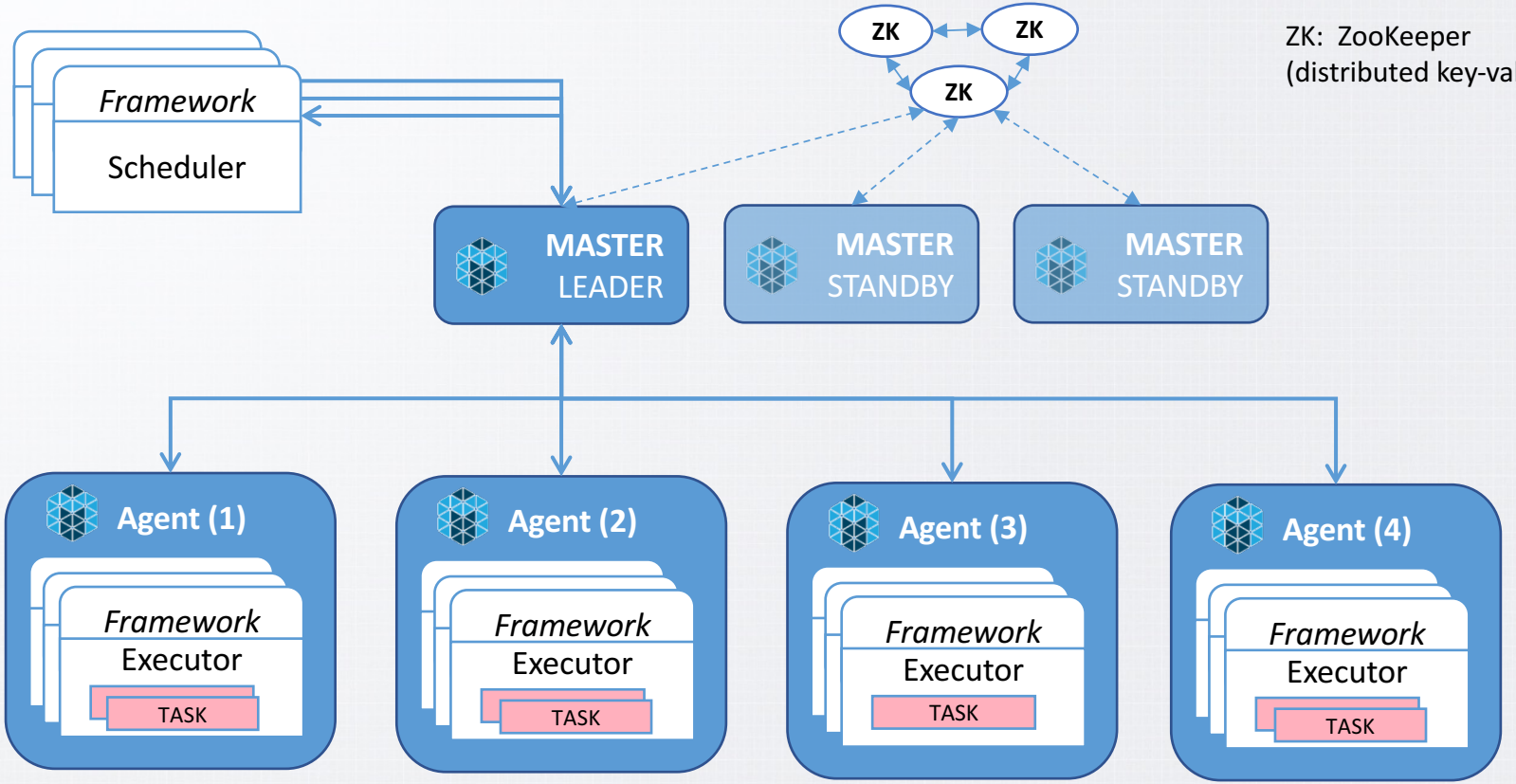
ZK: ZooKeeper
(distributed key-value store)

Mesos Architecture



ZK: ZooKeeper
(distributed key-value store)

Mesos Architecture



ZK: ZooKeeper
(distributed key-value store)

Why not use an existing framework?

Apache AURORA™

MARATHON

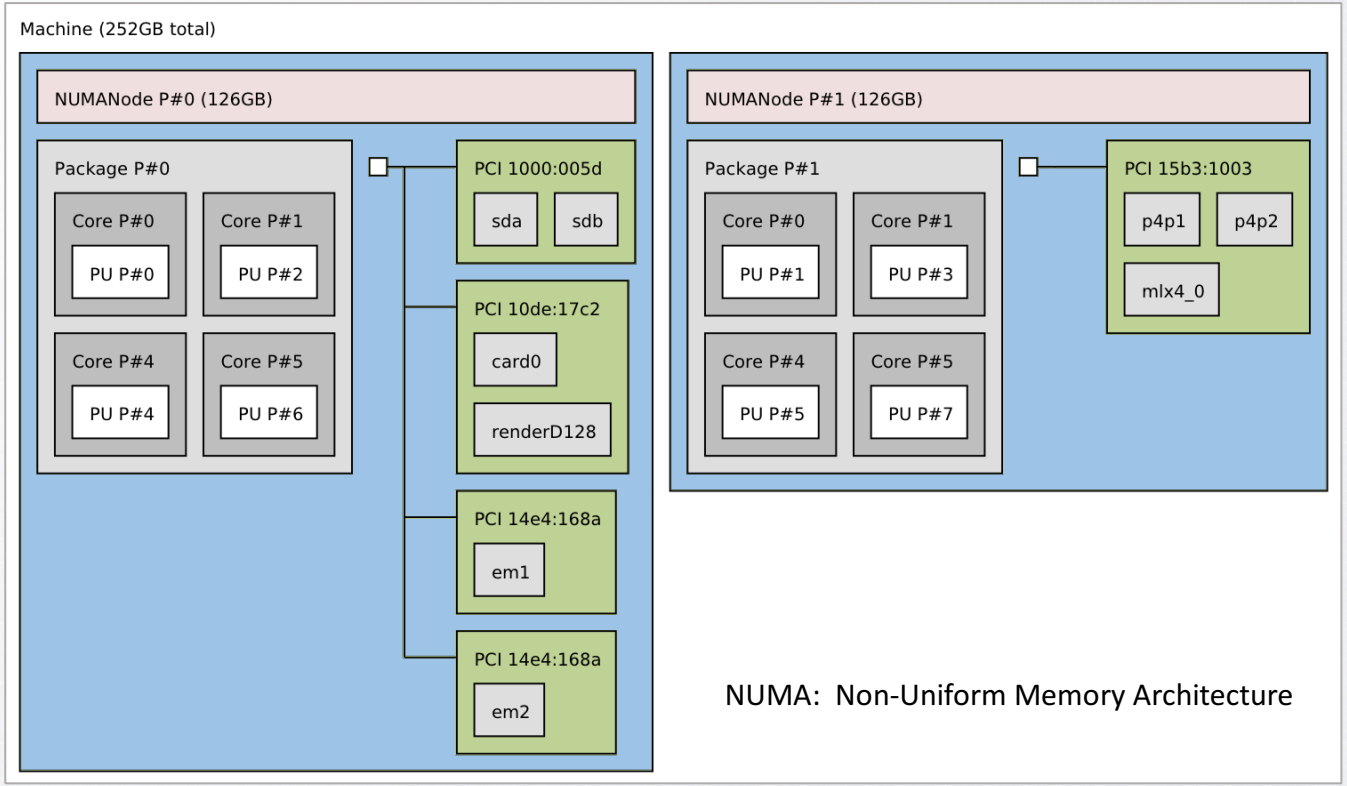
Spark

HubSpot / Singularity

Custom?

The world according to Mesos

The real world



NUMA: Non-Uniform Memory Architecture

NUMA support

E.g. run this task using 3 cores on the same NUMA node

Cores & Topology

Custom resources: `cores: [0-7]`

Custom attributes: `node.numa: [[0,2,4,6], [1,3,5,7]]`

Scheduler

Scheduler assigns cores from same node for NUMA-sensitive tasks

Docker

Use option:

`--cpuset-cpus`

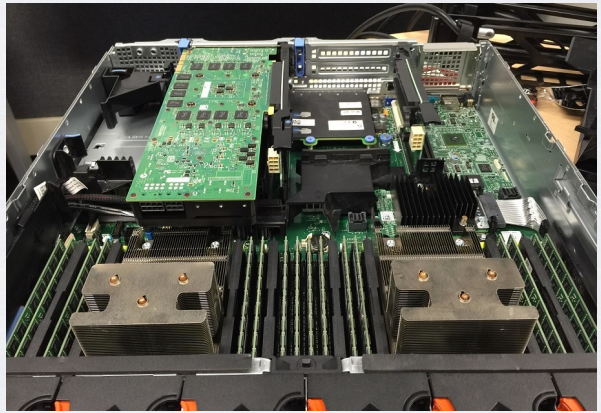


Image from: http://en.community.dell.com/

GPU support

E.g. run this task on a Titan X GPU, using 25% compute and 3GB RAM, on same NUMA node as CPU cores

GPU details

Custom resources:
```
node.gpu.0.compute: 1.0
node.gpu.0.mem: 8192
```

Custom attributes: NUMA node, GPU type, `/dev` entries, CUDA info, etc.

Scheduler

Scheduler assigns node with correct GPU and capacity, and cores from same NUMA node

Docker

E.g. pass in `/dev/nvidia0` as Docker argument
Also have Docker images optimised for some GPUs

Image from: https://www.geforce.com/

Note: Resource limits are not enforced by operating system

Network support

E.g. run this task on an ibverbs-capable NIC on the CBF network, allowing 27 Gb/s in-bound bandwidth, and using a single NUMA node

NIC details

Custom resources:
```
node.interface.0.bandwidth_in: 40e9
node.interface.0.bandwidth_out: 40e9
```

Custom attributes: IF name, `/dev` entries, network segment, NUMA node

Scheduler

Scheduler assigns node with correct NIC and capacity, and cores from same NUMA node

Docker

E.g. pass in `/dev/em1` as Docker argument

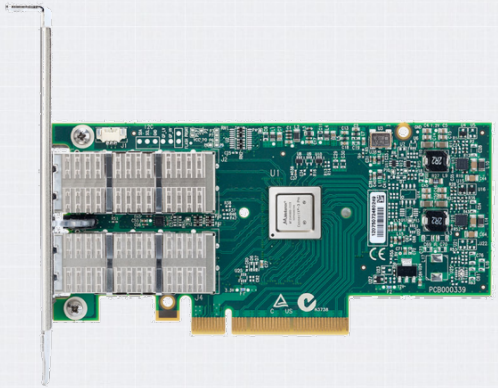


Image from: http://www.mellanox.com/

Note: Resource limits are not enforced by operating system

The good

- *PyMesos* Python package hides complexity of the HTTP API
- Mesos cleans up after framework crashes
- Mesos UI handy for debugging
- Mesos developers are friendly and responsive
- Describing placement policies with code makes for ultimate extensibility
- Only 5k lines of Python code for our custom scheduler

The not so good

- Attribute values can only contain: `A-Za-z0-9_/.-` (so had to use base64 encoding)
- Changing attributes of an agent requires recovery step (kill tasks and restart)
- No GUI support for custom resources
- Mesos not as rock-solid as expected (during our development)
- Fault-tolerance is still hard

Conclusion

- SDP pipelines are now very scalable
- Easy to recover from a node failure
- Custom scheduler allows us to maintain optimal performance
- Could be improved for much higher availability

Questions?

Thank you

More details in the paper: THBPA04

science & technology
Department:
Science and Technology
REPUBLIC OF SOUTH AFRICA

NRF National Research Foundation

SKA SQUARE KILOMETRE ARRAY Africa