Hadoop Distributed Batch Processing For Gaia

Marco Riello
IoA - Cambridge

DPCI - IoA Cambridge

- One of the 6 DPCs taking care of the Gaia data processing
- We are a small team
  - Francesca De Angeli (DPC manager)
  - Marco Riello (Deputy DPC manager, Tech-Lead, Processing)
  - Greg Holland (Software Architecture)
  - Patrick Burgess - Data Management, System Monitoring
  - Paul Osborne - Build Engineer, Test Manager
  - Sue Cowell & Neil Miller (sysadmin)

What do we do ?

- Data **Processing**
  - Photometric calibration (integrated photometry, BP/RP spectra)
  - Source Environment Analysis (not active yet)
- **Software Development**
  - Core software infrastructure for the calibration algorithms
  - Release, Validation, Integration & Operation of PhotPipe
- **Hardware**
  - procurement
  - maintenance

Challenges

Challenges I : the instrument

- Gaia has a complex payload comprised of several instruments
  - The payload instruments can be configured in different modes -> lots of parameters that have to be tracked
  - TDI mode, gating strategy to mimic different exposure times
  - Downlink is not sufficient to download full images: windowing scheme, compression and highly optimised representation of the data packets -> leads to a complex data stream (we consume more than 250 data types)
    - windowing scheme, hardware & software binning
  - Processing is distributed : communications are important !

Gaia DPAC

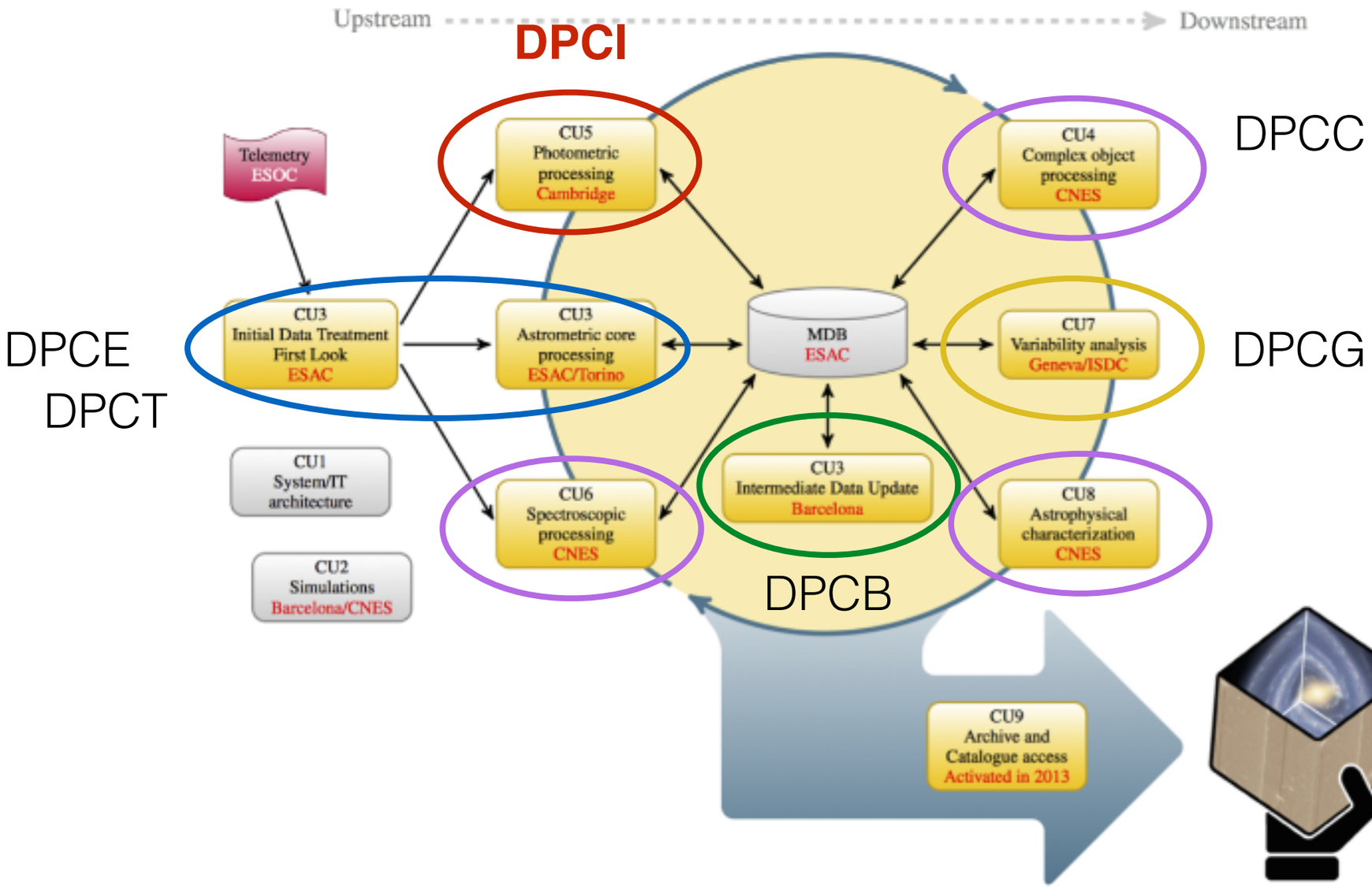
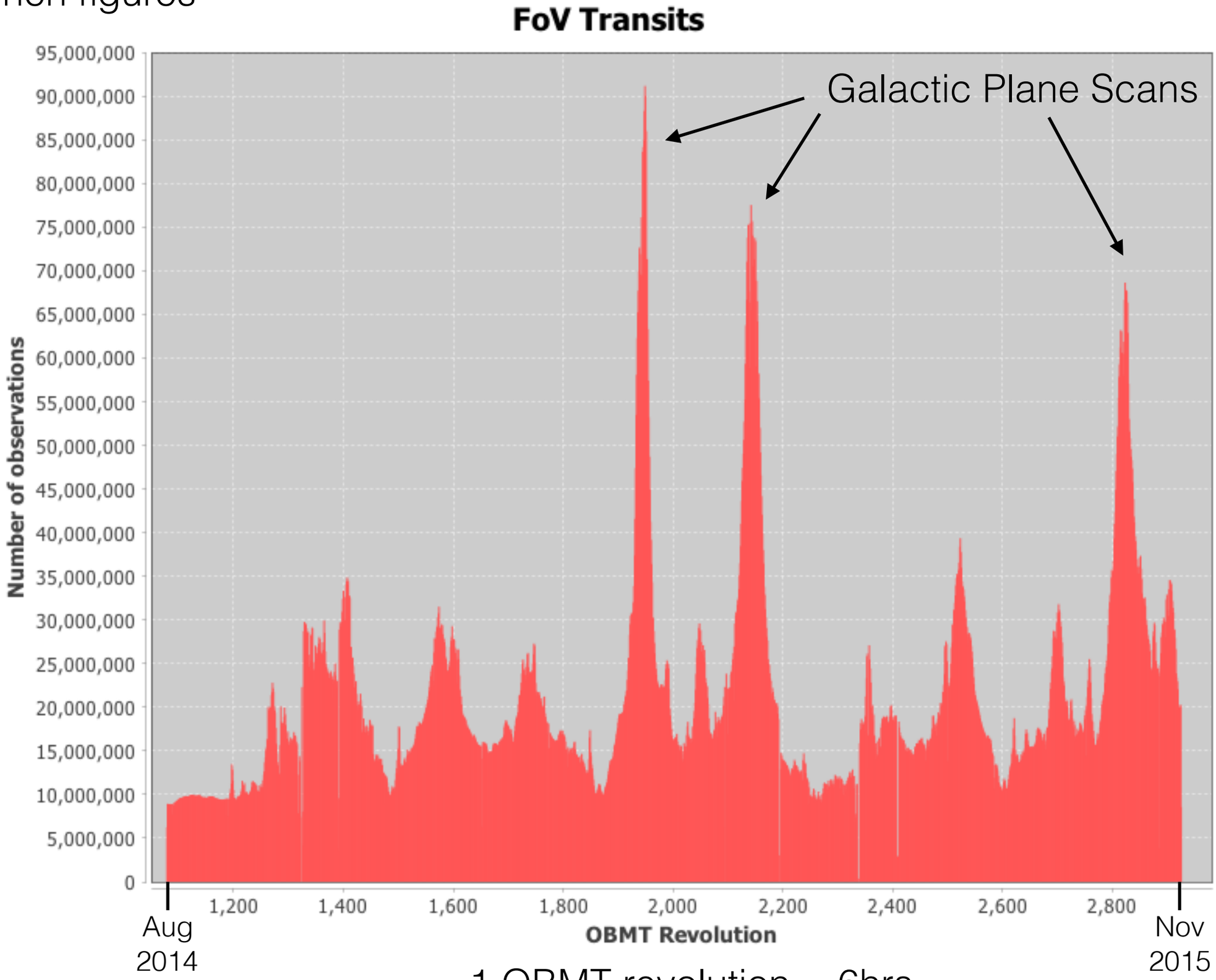


Figure Courtesy A. Brown, DPAC

Challenges II : the processing

- Mission is self-calibrating:
  - iterations within a processing system
  - iterations between DPAC systems (processing cycles)
  - quality improves with time -> also algorithms become more complex
  - **scalability** of processing systems is critical
- Data rate can vary by one order of magnitude
- All sub-systems, from data delivery and management to processing pipelines need to be able to scale with the increasing data volume.
- Resilience is also critical: 1/million failures is still producing > 1000 failures

~90 million transits/day
2x pre-launch figures



1 OBMT revolution = 6hrs

DPCI System

50000ft overview

Cerberus Processing Cluster

Messaging Service

NFS S. Alerts

NFS Staging

NFS Backup

QLOGIC

Jedi
DTA

Jedi
REST

LOG

ESAC

aspera

internet - 300Mbps

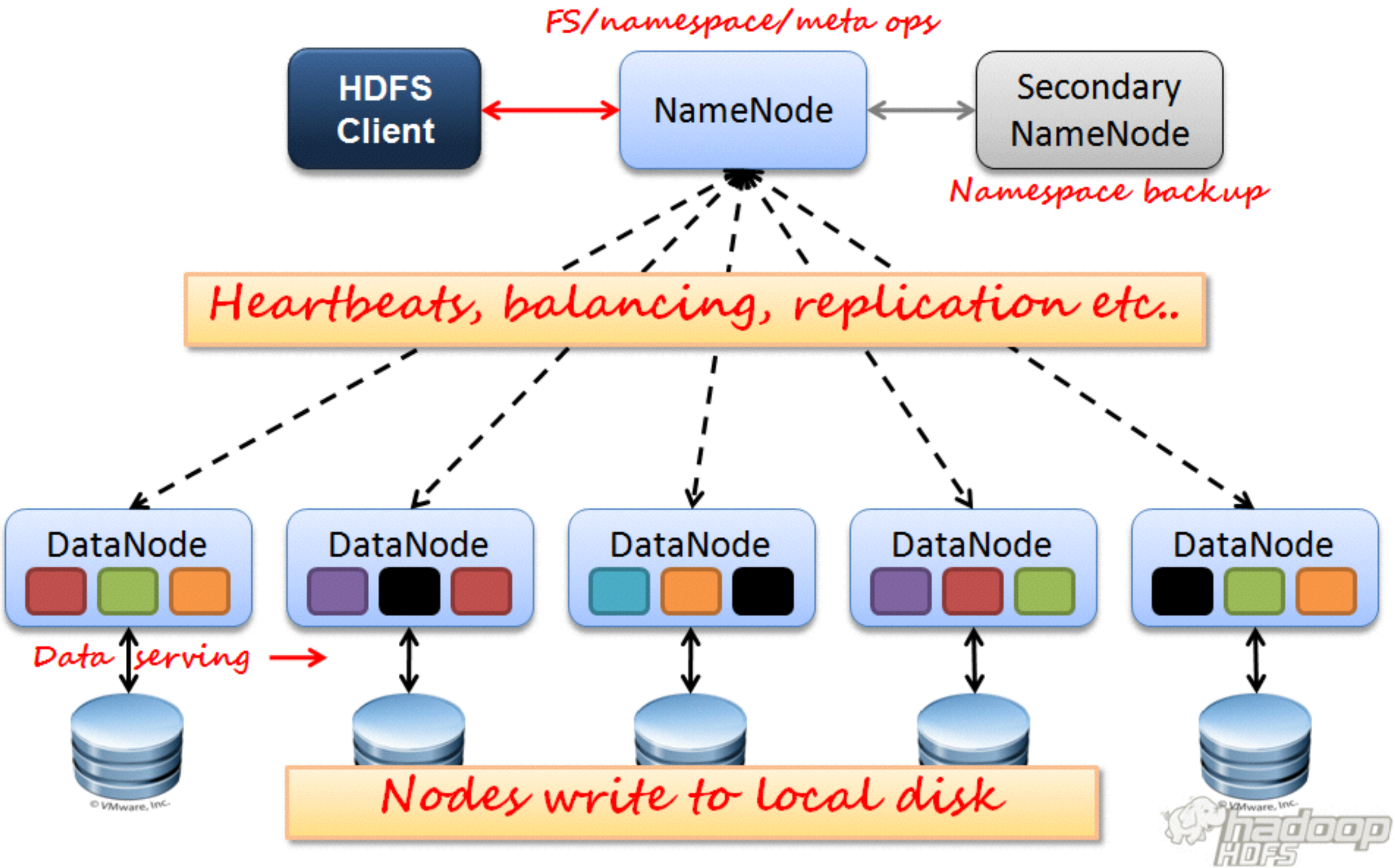
Cambridge

Jedi
Aspera


hadoop

Hadoop in a nutshell - I

- Distributed filesystem (HDFS)
  - Data is replicated at the block level on different nodes
  - *Namenode* keeps a registry of block locations and map to files
  - Nodes can be added/removed without downtime
  - Self-healing
    - Heartbeat mechanism to monitor node status
    - HDFS balancer ensure even distribution of data across the cluster
    - *Namenode* checks for under/over-replicated blocks
- Scales horizontally by adding more nodes


FS/namespace/meta ops
HDFS Client
NameNode
Secondary NameNode
Namespace backup
Heartbeats, balancing, replication etc..
DataNode
DataNode
DataNode
DataNode
DataNode
Data serving
Nodes write to local disk
© VMware, Inc.
© VMware, Inc.
hadoop HDFS

Hadoop in a nutshell - II

- Distributed filesystem (HDFS)
- Distributed processing
  - Application & Resource Management - YARN
  - Map/Reduce, a simple parallelisation abstraction

Hadoop in a nutshell - II

- Distributed filesystem (HDFS)
- Distributed processing
  - application & resource management - YARN
  - Map/Reduce, a simple parallelisation abstraction (Google, inspired by Lisp)

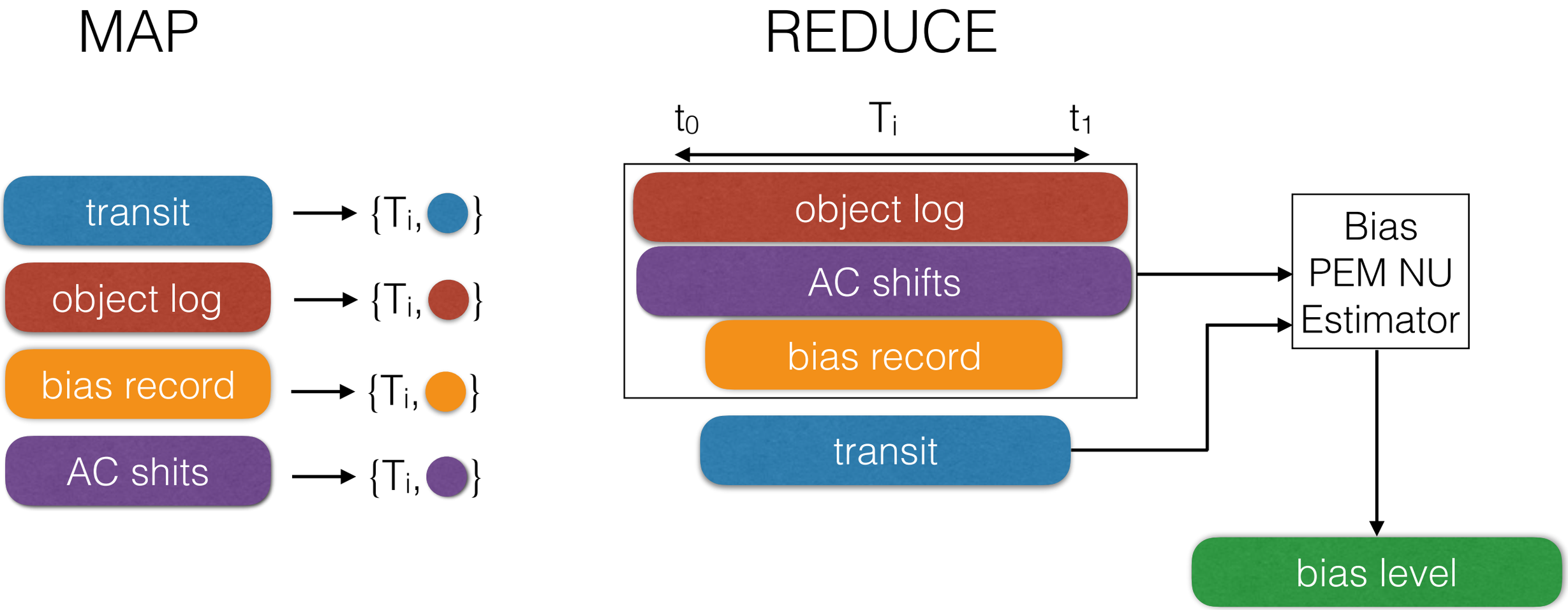
```
map (k1, v1) -> list (k2, v2)

reduce (k2, list(v2)) -> list (k3, v3)
```

Hadoop in a nutshell - II

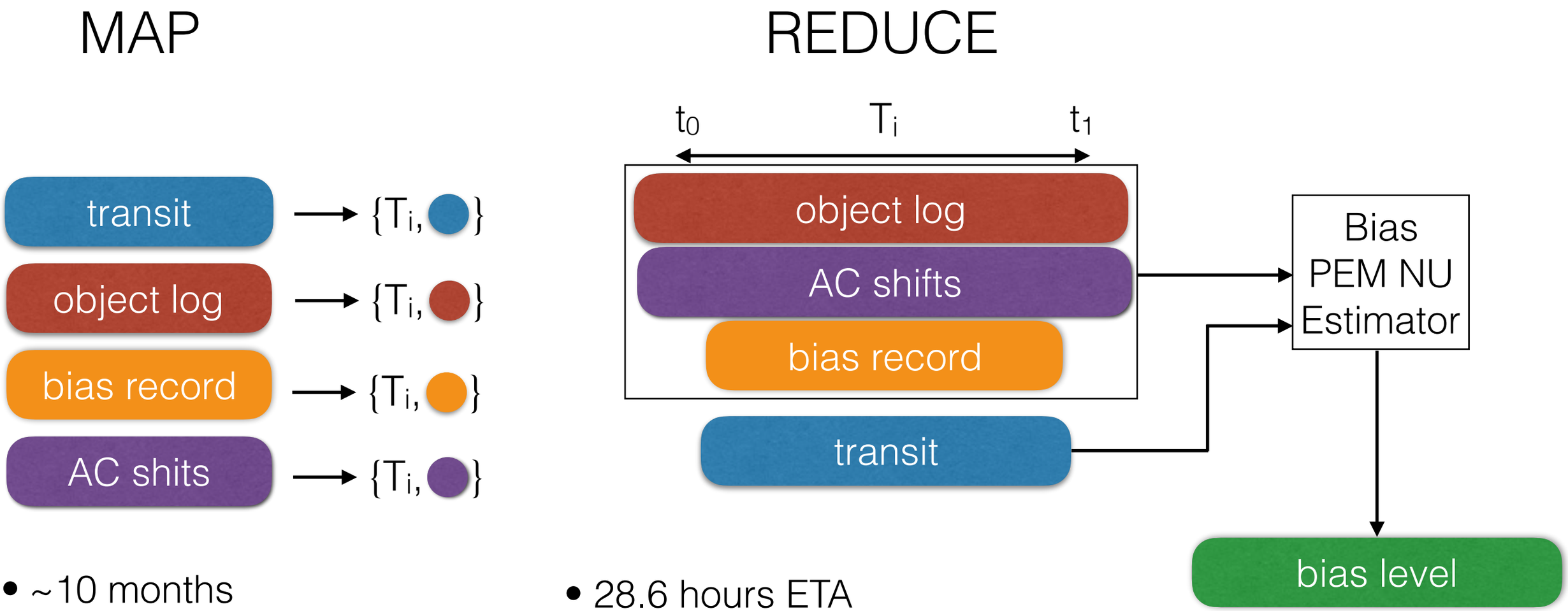
- Distributed filesystem (HDFS)
- Distributed processing
  - application & resource management - YARN
  - Map/Reduce, a simple parallelisation abstraction
  - SPARK (machine learning, iterative processes, better API)
- Resilient to failures & glitches
- **Data locality** : bring the processing to the data
  - read at the speed of your local disk
  - scales horizontally by adding more nodes:
    - process data faster
    - process more data in same amount of time

- **108** Hadoop data/proc nodes
  - Dual 6 cores @ 2.4GHz
  - 48GB memory
  - 9x1TB (enterprise class) disks, a few upgraded to 4TB
  - 40Gb IB/1Gb eth/service
  - Centos Linux
- Providing altogether
  - **1296** cores
  - 5.1TB memory
  - ~0.9PB HDFS storage
    - **1.6PB** including NFS

Bias PEM NU correction for BP/RP spectra



Bias PEM NU correction for BP/RP spectra



- ~10 months
- 22.5 billion transits
- 45 billion spectra
- ~5TB

- 28.6 hours ETA
- ~370,000 spectra/second
- Used only 60% of cluster

Photometric chain : 10months

Total Jobs	700
Elapsed	8.84 days
CPU Elapsed	5.71 yrs
Total maps	1481265 [98.8%]
Total reducers	40267
Total read	1.529 PB
Total written	1.701 PB

Conclusions

- A sound and scalable processing architecture is the key
- Hadoop proved to be a viable and extremely robust solution
  - Map reduce is a solid workhorse for distributed batch processing
  - On some workflows, Spark can provide 10x-100x w.r.t. plain old Map/Reduce
- I wish I had time to tell you more about the software implementations
  - Data modelling and formats
  - Functional programming
- I hope you will all enjoy the first Gaia Data Release in summer 2016 !

Questions ?