CorrelX: A Cloud-Based VLBI Correlator

V. Pankratius, A. J. Vazquez, P. Elosegui

Massachusetts Institute of Technology
Haystack Observatory

pankrat@mit.edu, victorpankratius.com

1st NEROC Symposium: Radio Science and Related Topics
November 4, 2016

Speaker: A. J. Vazquez (slides from IVTW2016)

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Scalability is Key

- Examples: Current & Future Radio Arrays

  - 10s of Tbit/sec during cross-correlation
  - 1 terabyte/day typical output data rate

Raw Data		Cost	Sensors
~72 Tbit/s	ASKAP (under construction)	$160M	96-pixel phased array feeds on 36 12m dishes
~4 Tbit/s (10 PB science-ready data / year)	MeerKAT (under construction)	$100M	64 13.5m antennas, wideband single-feed pixels
~10 Tbit/s	SKA Phase 1 (construction starting 2017)	$900M	low-freq. component: 262144=2^{18} dipoles
~2.5 Pbit/s	SKA Phase 2 (construction ?)	?	3-4 million antennas

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

CorrelX

New architecture

- Breaks with current paradigms that aim to mimic hardware correlators in software
- Based on chunk streaming model
- Enables offline, out-of-order correlation of chunks

Key Features

- Cloud scalability
- Elastic correlator deployments for variable workloads, burst capability
- Reliability on large number of nodes through built-in online testing mechanisms (injected control chunks)
- Plugin-based. Enables quick prototyping and extensions for research and teaching
- Separate concerns: core correlation code vs. parallel computing code
- Leverages open-source: Map-Reduce, parallel file systems
- Enables machine-learning-based performance optimization

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

CorrelX

- Proof of concept prototype, incremental feature additions
- Core: Map-Reduce paradigm
  - Popularized by Google
  - Programming model: two types of workers:
    - Map(input_block) → set of [key,value] pairs
    - Reduce(set of [key,value] pairs) → set of [results]
  - Parallelism hidden by framework:
    - Framework launches one map per input block, one reducer per partition (i.e., all pairs identified by same key).
  - Apache Hadoop: mapper and reducer can be written in different languages (e.g., Python, C++, ...)
  - Parallel file system: Lustre
    - Max. volume size: “over 16 EB (theoretical)” (2.8.59, Oct 2016)
    - Used in over 50% of Top 100 supercomputers
    - “Presents all clients with a unified namespace for all of the files and data in the filesystem”

MIT HAYSTACK OBSERVATORY Massachusetts Institute of Technology

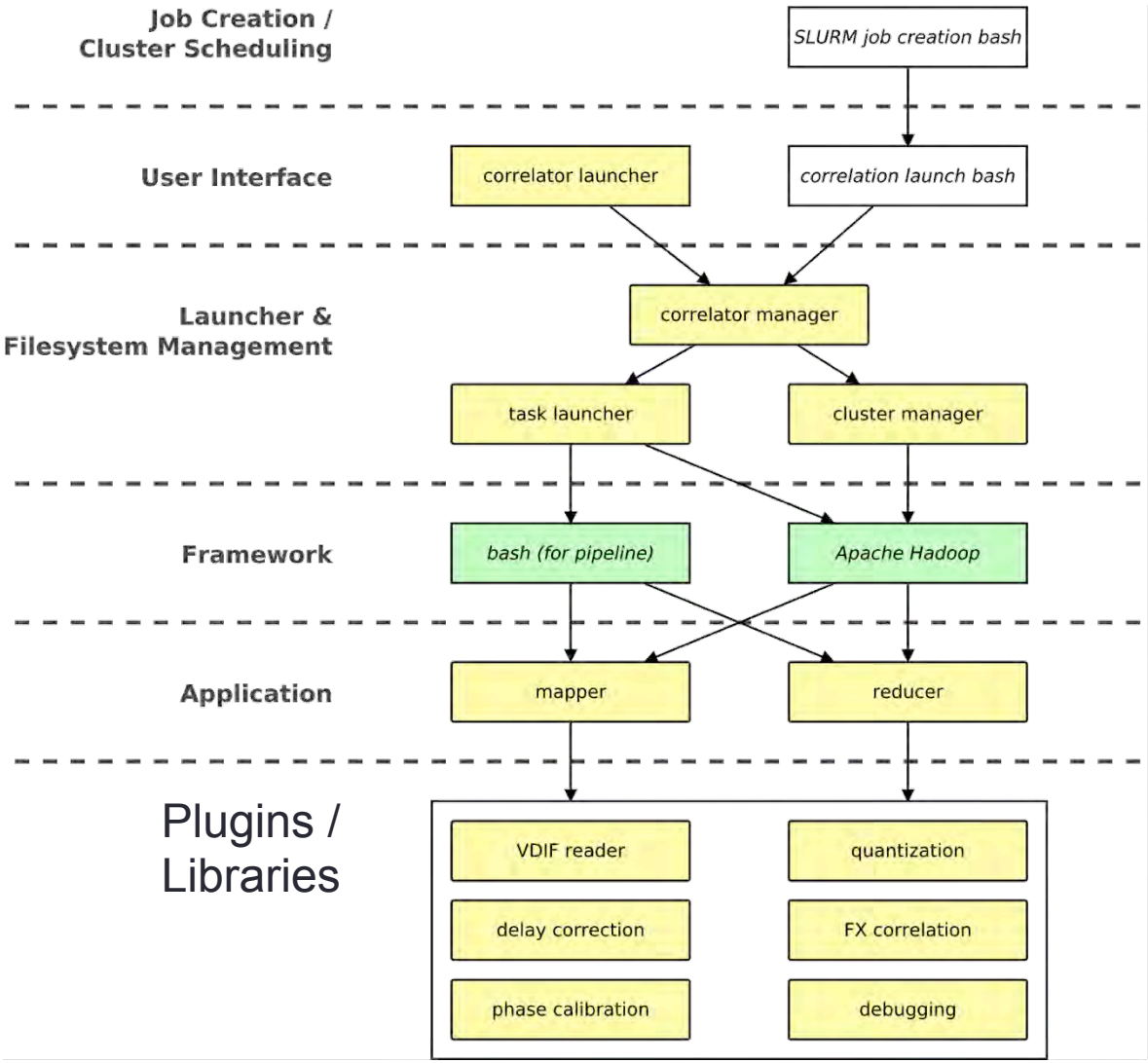
CorrelX

- CorrelX Correlation Configuration
  - Deployment and performance configuration: .xml files
  - Partitioner: define sub-keys for partitioning (what goes into each reducer) and sorting (to reconstruct “order” after“out-of-order” parallel computing)
- FX correlation
  - Parallelization in frequency bands and accumulation periods
  - Parallelization in baselines configurable: single-baseline-per-task, linear-scaling-stations and all-baselines-per-task (default is all-baselines-per-task)
- Delay correction using polynomials from CALC (vex2difx+calcif2).
- Phase calibration tone extraction
- Zoom bands (currently during post-processing)
- Interfaces:
  - Input format: VDIF (configuration read from frame header), complex or real, 1 or 2 bits per sample, multiple threads xor multiple channels per frame
  - Output format: text file with visibilities and phase calibration results

MIT HAYSTACK OBSERVATORY Massachusetts Institute of Technology

CorrelX Implementation Details

- Correlator manager
  - Deploy cluster
  - Process configuration files
  - Launch correlation
- Cluster manager
  - Methods for starting/stopping cluster
  - Methods for filesystem initialization
- Task launcher
  - Methods for correlation job launch
- Map
  - VDIF read and corner-turning
  - Initial integer-sample alignment
- Reduce
  - Fractional sample shift
  - Fringe rotation
  - Fourier transform
  - Fractional Sample Correction
  - Multiply and accumulate



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

CorrelX Implementation Details

Project	Language	Lines of code
CorrelX	Python	9k 2k comments
Libs.	Python	2M
Hadoop	Java	2M
Lustre	C	1M

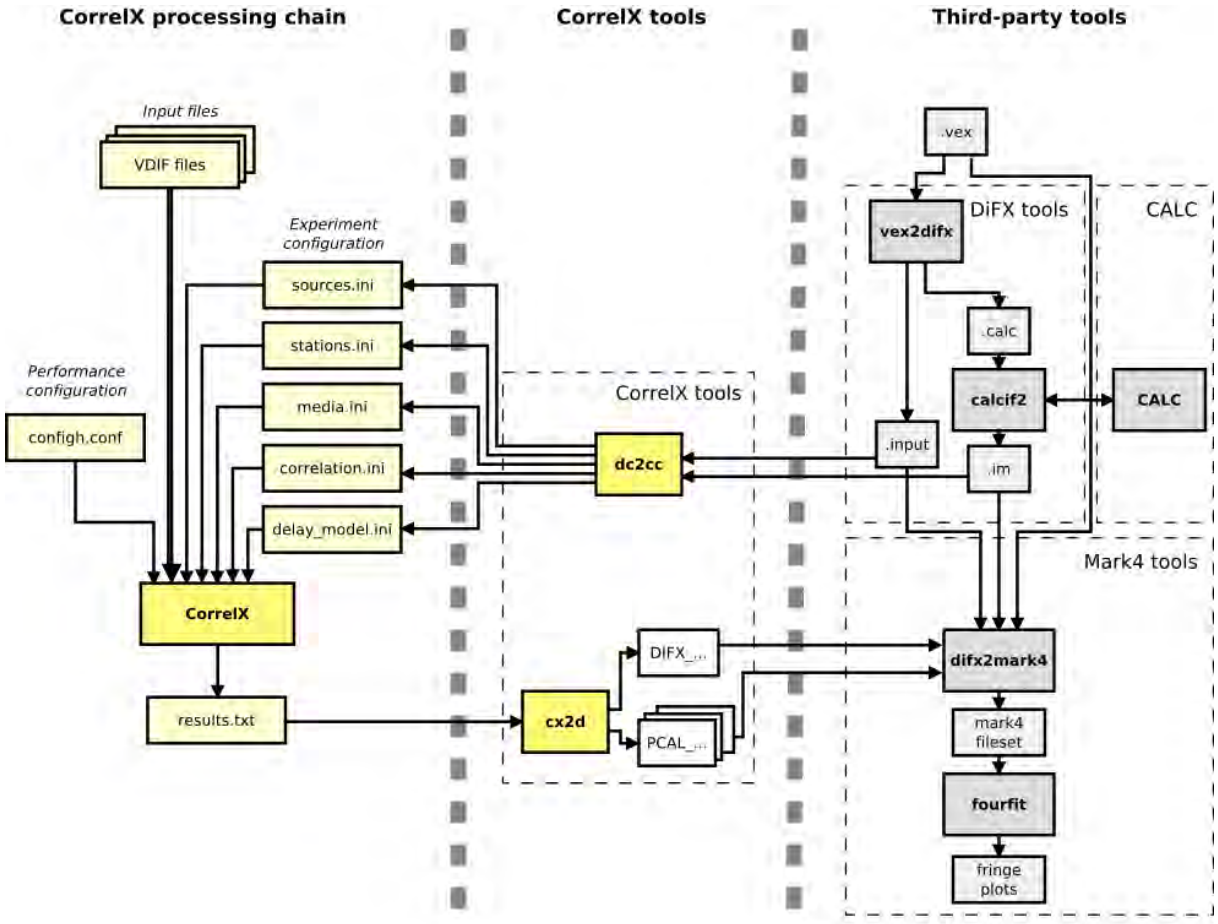
This is code maintained by open-source projects

CorrelX Layer	LOC (approx.)
Launcher and FS Management	2k
Application (map, reduce)	1.8k
Libraries (VDIF, quant., FX…)	4k
Tools (dc2cc, cx2d…)	1.2k

This is the code the VLBI community needs to maintain

MIT HAYSTACK OBSERVATORY Massachusetts Institute of Technology

Tool Chain Integration



- Tools for integration with existing processing chains:
  - Input file converter from .input and .im
  - Output file converter to SWIN and PCAL files
- Integration with Mark4 post-processing chain
  - dc2cc: configuration
  - cx2d: output conversion
- Experiment files
  - sources.ini
  - stations.ini
  - media.ini
  - correlation.ini
  - delay_model.ini
- Correlator configuration
  - configh.conf

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Results

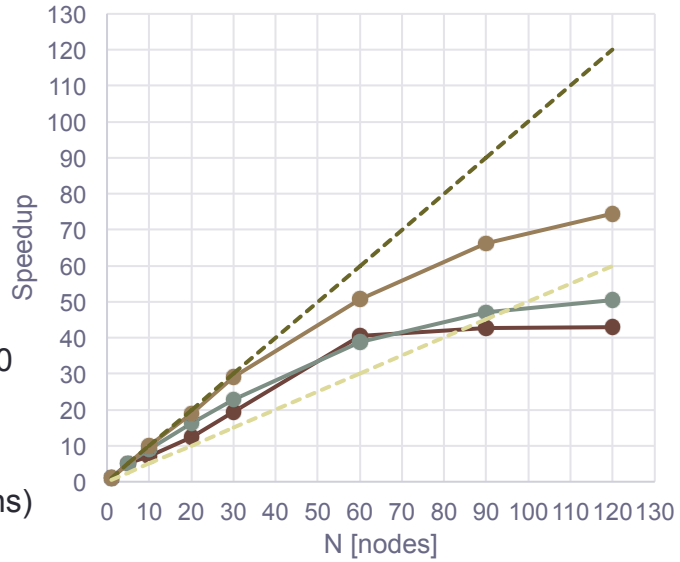
- Real-world datasets
  - Event Horizon Telescope (EHT)
  - VLBI Global Observing System (VGOS)

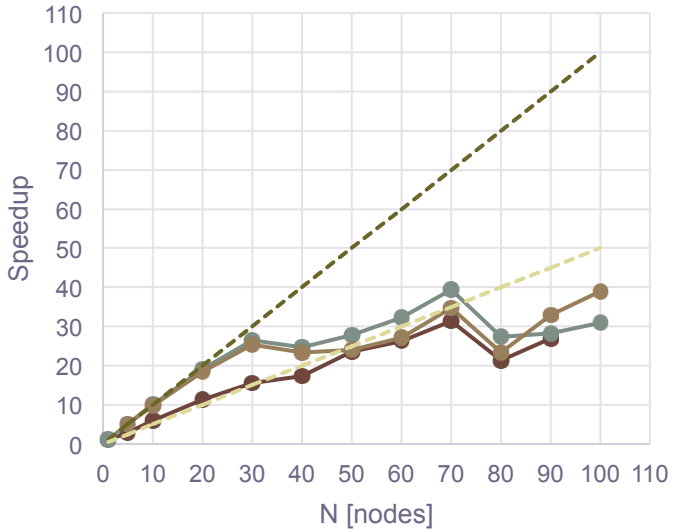
- Cloud environment
  - Massachusetts Green High Performance Computing Center
  - Nodes: ~200
  - Intel(R) Xeon(R) CPU E5-2660 v3 @ 2.60GHz, 20 cores/node, 32 GB RAM
  - FDR Infiniband network (~56 Gps)
  - NFS filesystem: ~1.2 Gbps
  - Lustre filesystem: ~2.2 Gbps
  - Job scheduling: SLURM

MIT HAYSTACK OBSERVATORY Massachusetts Institute of Technology

Results

- **Scalability benchmarking**
  - Cluster configuration:
    - 14 cores per node for map or reduce containers
    - 1 container (map or reduce) per core
    - 1 node for manager
    - From 1 to 120 nodes for MapReduce (up to 1680 cores)
  - Experiment configuration:
    - Duplicated stations and data (4, 8 and 16 stations)
    - No delay correction computations
    - No phase calibration, no zoom bands
  - Datasets:
    - EHT bm434a:
      - FFT 40960
      - Scan 300 s, acc. 0.33 s, 2 bands: 1818 reducers
      - Reducer scaling: 1818/14 ≈ 130 nodes
    - VGOS v15328:
      - FFT 128
      - Scan 30 s, acc. 1 s, 32 bands: 960 reducers
      - Reducer scaling: 960/14 ≈ 69 nodes

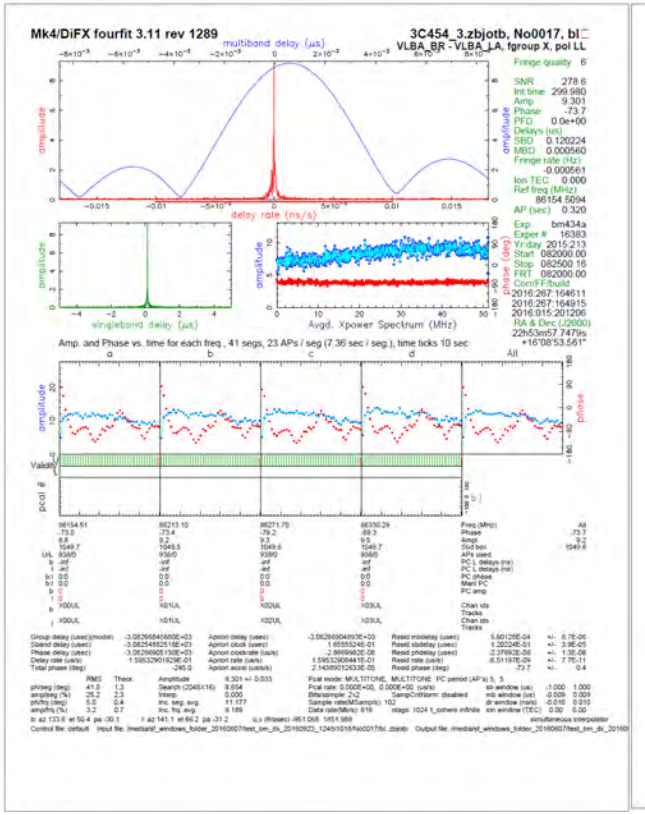


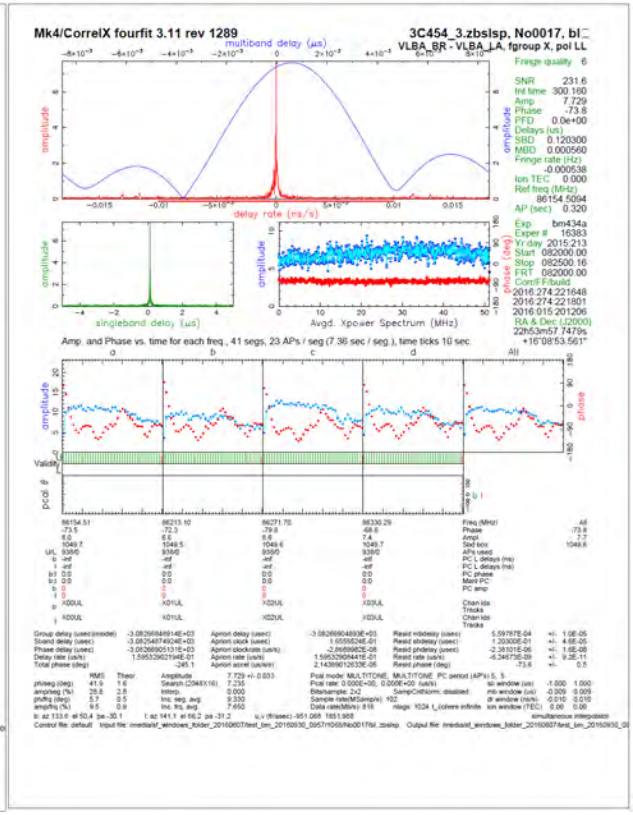


MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Results: EHT

- EHT bm434a:
  - VLBA BR-LA, 2015
  - Scan: 300 s
  - Acc.: 0.32 s
  - Sideband: USB
  - Sampling: 256 MHz
  - 2 bits real
  - Zoom bands 51.2 MHz
- Work in progress:
  - Details in next slide.

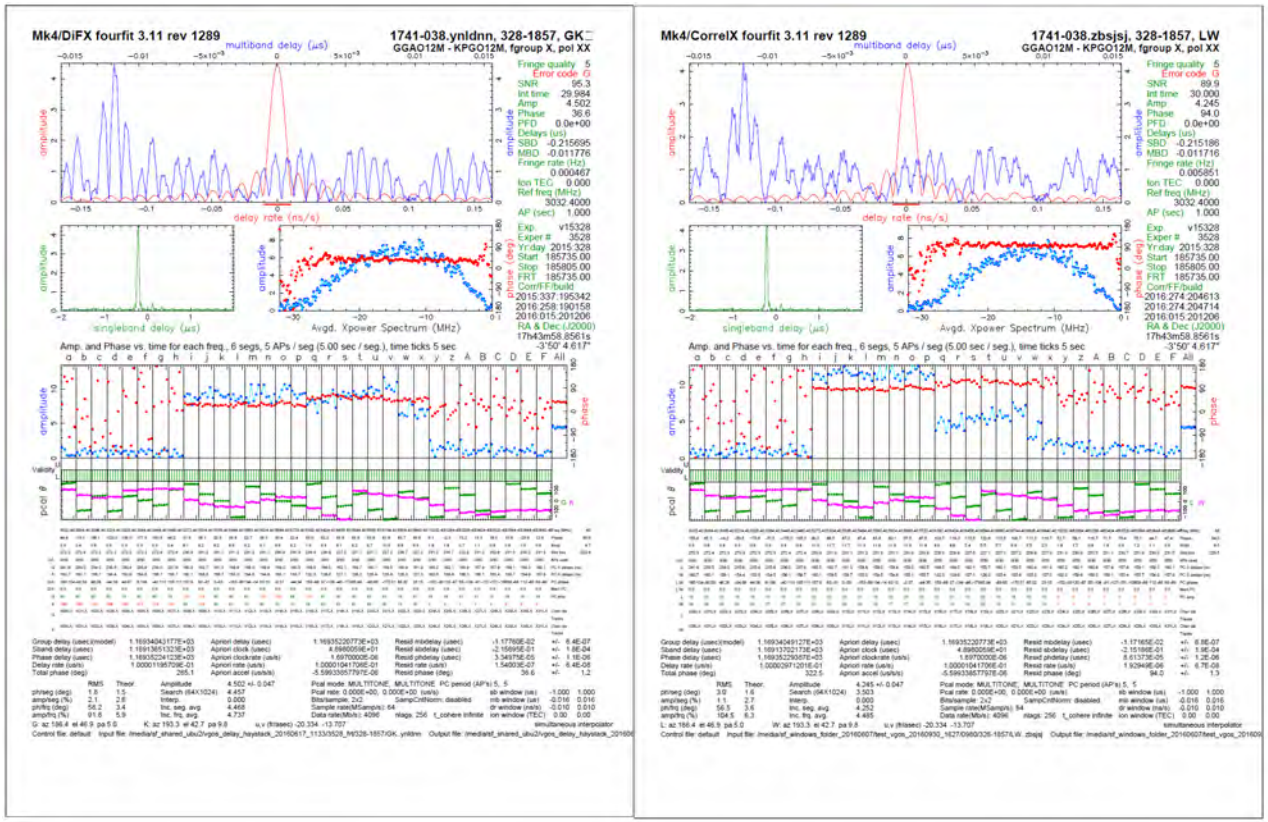




MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Results: VGOS

- VGOS v15328:
  - GGAO-KPGO, 2015
  - Scan: 30 s
  - Acc.: 1 s
  - Sideband: LSB
  - Sampling: 32 MHz
  - 2 bits complex
  - Phase calibration
- Work in progress:
  - Details in next slide.



MIT HAYSTACK OBSERVATORY

Massachusetts Institute of Technology

Open Source Release Soon

https://github.com/MITHaystack/CorrelX

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Conclusion

- CorreIX
  - Proof-of-concept functionality for VGOS and EHT correlations
    - VGOS: LSB 2-bit complex, phase calibration
    - EHT: USB 2-bit real, zoom bands
    - Integrated in state-of-the-art processing chain
    - Early prototype, scales with number of nodes, but still work to do (add more functionality, improve SNR, per-node performance)
  - **Flexibility, scalability, simplicity**
  - Plugins: Opportunity for research and teaching. Prototype new algorithms, mapper-reducers, execute on real data sets easily

MIT HAYSTACK OBSERVATORY Massachusetts Institute of Technology

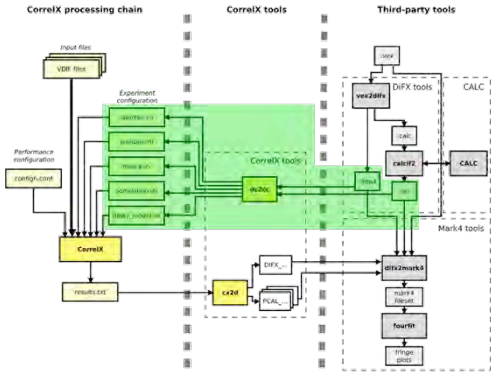
Thanks!

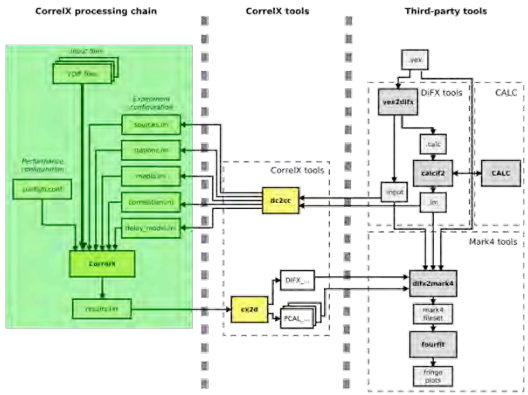
Questions? Comments?

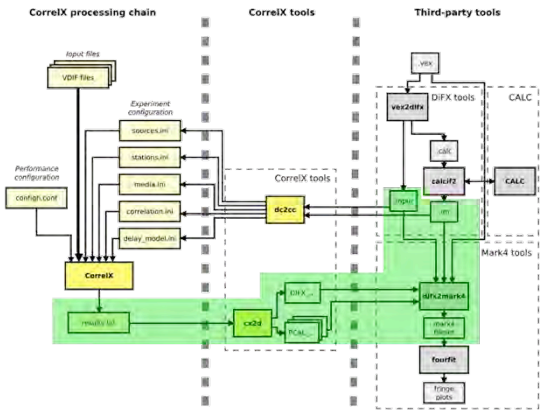
→ pankrat@mit.edu

Many thanks to: Roger Cappallo, Alan Rogers, and many others

Demos







- Configuration:

1. Show .im, .input files.
2. Run *dc2cc*.
3. Show .ini files.

- Correlation:

1. Show .ini files
2. Show .vdif files
3. Show .conf file
4. Run *correlx*

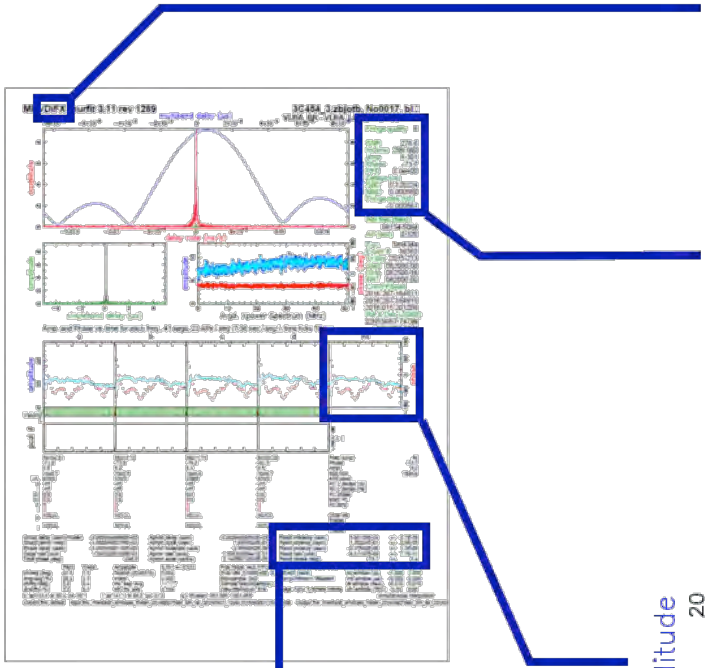
- Output conversion:

1. Show output file
2. Run *cx2d*
3. Run *difx2mark4*

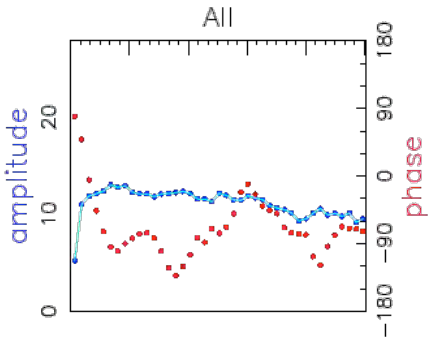
MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

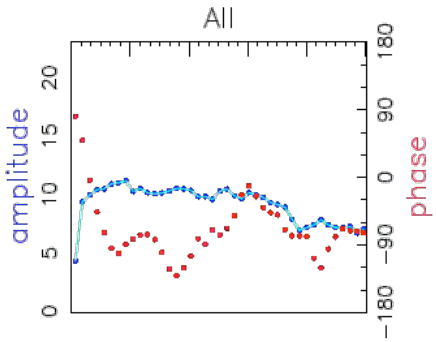
Results: EHT

- EHT bm434a:
  - Multiband delay approx. within tolerance
  - Lower SNR



	DiFX	CorrelX
Fringe quality	6	6
SNR	278.6	231.6
Int time	299.980	300.160
Amp	9.301	7.729
Phase	-73.7	-73.8
PFD	0.0e+00	0.0e+00
Delays (us)		
SBD	0.120224	0.120300
MBD	0.000560	0.000560
Fringe rate (Hz)	-0.000561	-0.000538



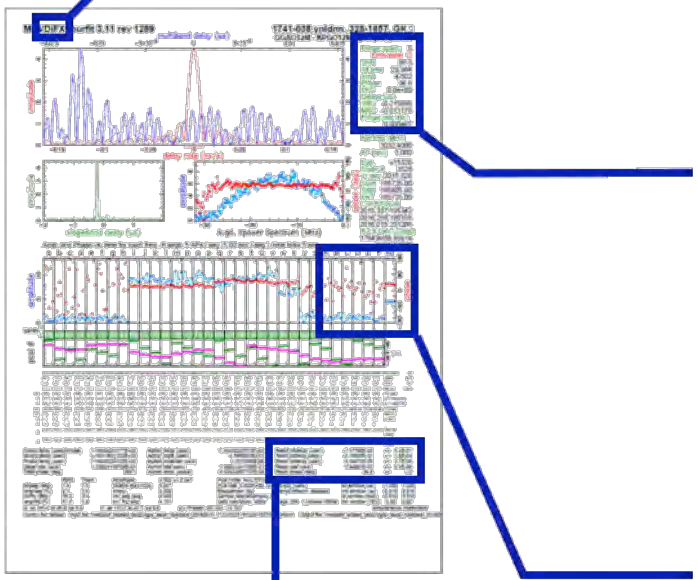


Resid mbdelay (usec)	5.60128E-04	+/-	8.7E-06	5.59787E-04	+/- 1.0E-05
Resid sbdelay (usec)	1.20224E-01	+/-	3.9E-05	1.20300E-01	+/- 4.6E-05
Resid phdelay (usec)	-2.37692E-06	+/-	1.3E-08	-2.38101E-06	+/- 1.6E-08
Resid rate (us/s)	-6.51197E-09	+/-	7.7E-11	-6.24673E-09	+/- 9.2E-11
Resid phase (deg)	-73.7	+/-	0.4	-73.8	+/- 0.5

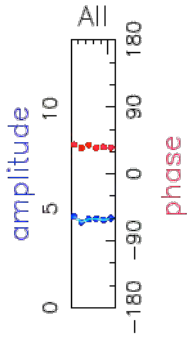
MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

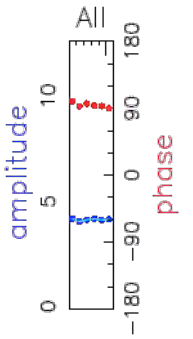
Results: VGOS

- VGOS v15328:
  - Errors still above tolerances
  - Lower SNR



	DiFX	CorreliX
Fringe quality	5	5
Error code	G	G
SNR	95.3	89.9
Int time	29.984	30.000
Amp	4.502	4.245
Phase	36.6	94.0
PFD	0.0e+00	0.0e+00
Delays (us)		
SBD	-0.215695	-0.215186
MBD	-0.011776	-0.011716
Fringe rate (Hz)	0.000467	0.005851



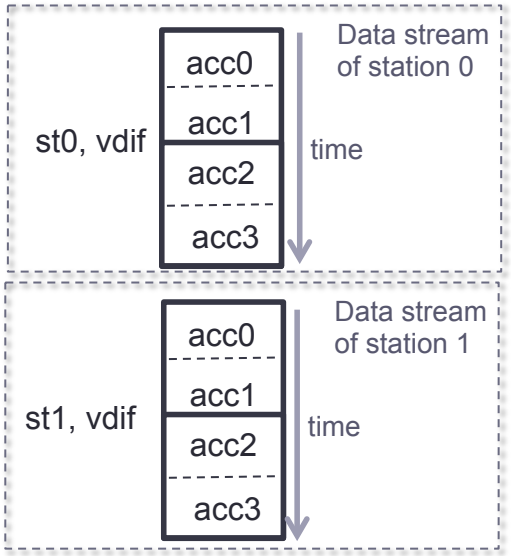


Resid mbdelay (usec)	-1.17760E-02	+/-	6.4E-07	-1.17165E-02	+/- 6.8E-07
Resid sbdelay (usec)	-2.15695E-01	+/-	1.8E-04	-2.15186E-01	+/- 1.9E-04
Resid phdelay (usec)	3.34975E-05	+/-	1.1E-06	8.61373E-05	+/- 1.2E-06
Resid rate (us/s)	1.54003E-07	+/-	6.4E-08	1.92949E-06	+/- 6.7E-08
Resid phase (deg)	36.6	+/-	1.2	94.0	+/- 1.3

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

- Correlation example:
  - Cluster:
    - 3 nodes.
    - 2 cores/node.
  - Experiment:
    - 2 stations.
    - 2 files, 2 parts/file
      => 2*2= 4 mappers
    - 2 bands.
    - 4 accumulation periods
      => 2*4=8 reducers



Mapreduce description

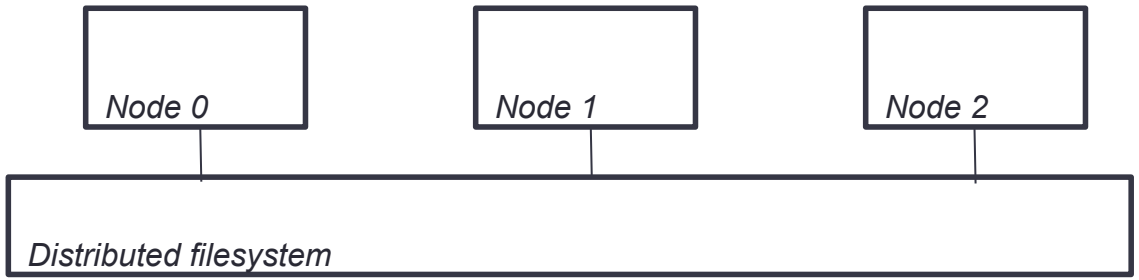
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

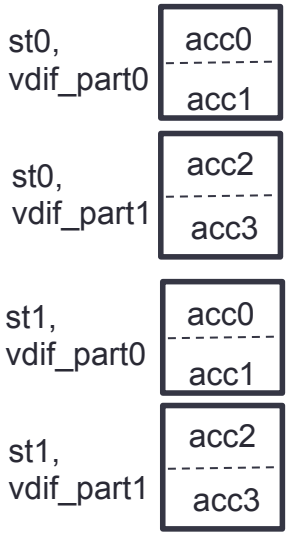
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.



Mapreduce description

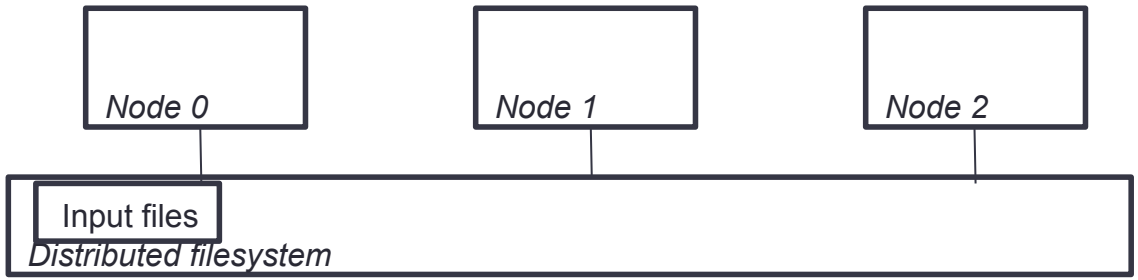
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

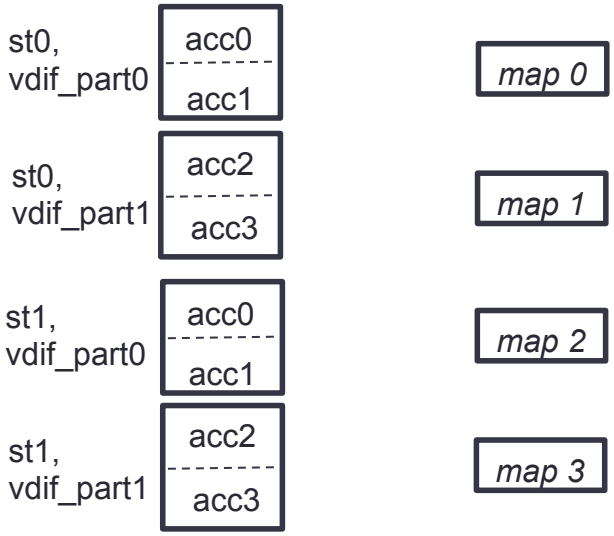
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.

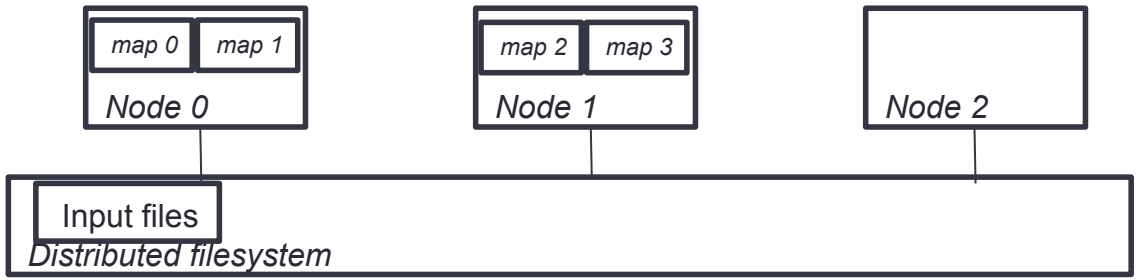


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

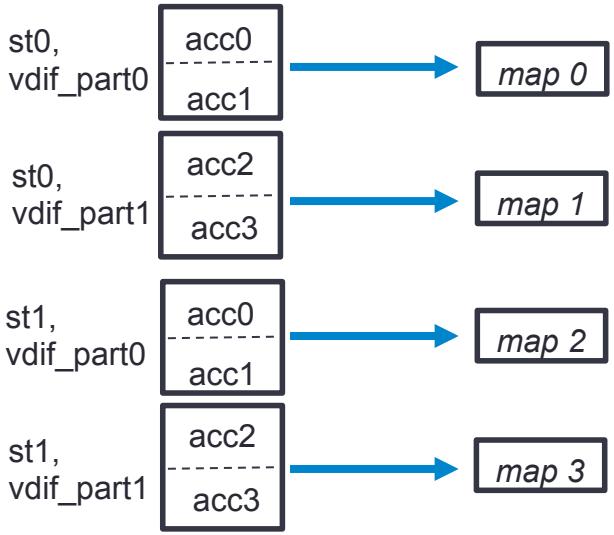
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.



Mapreduce description

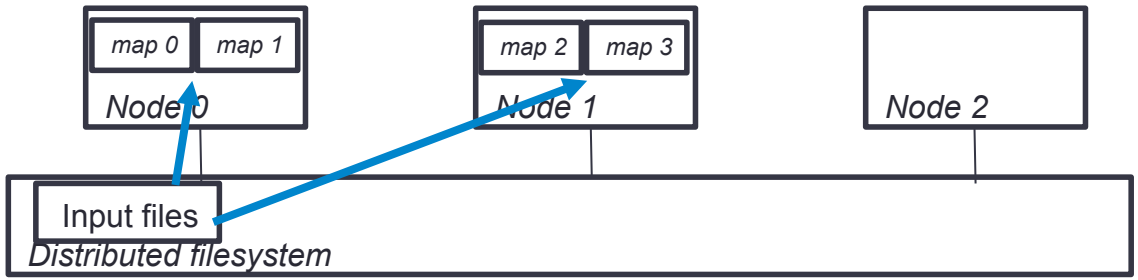
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

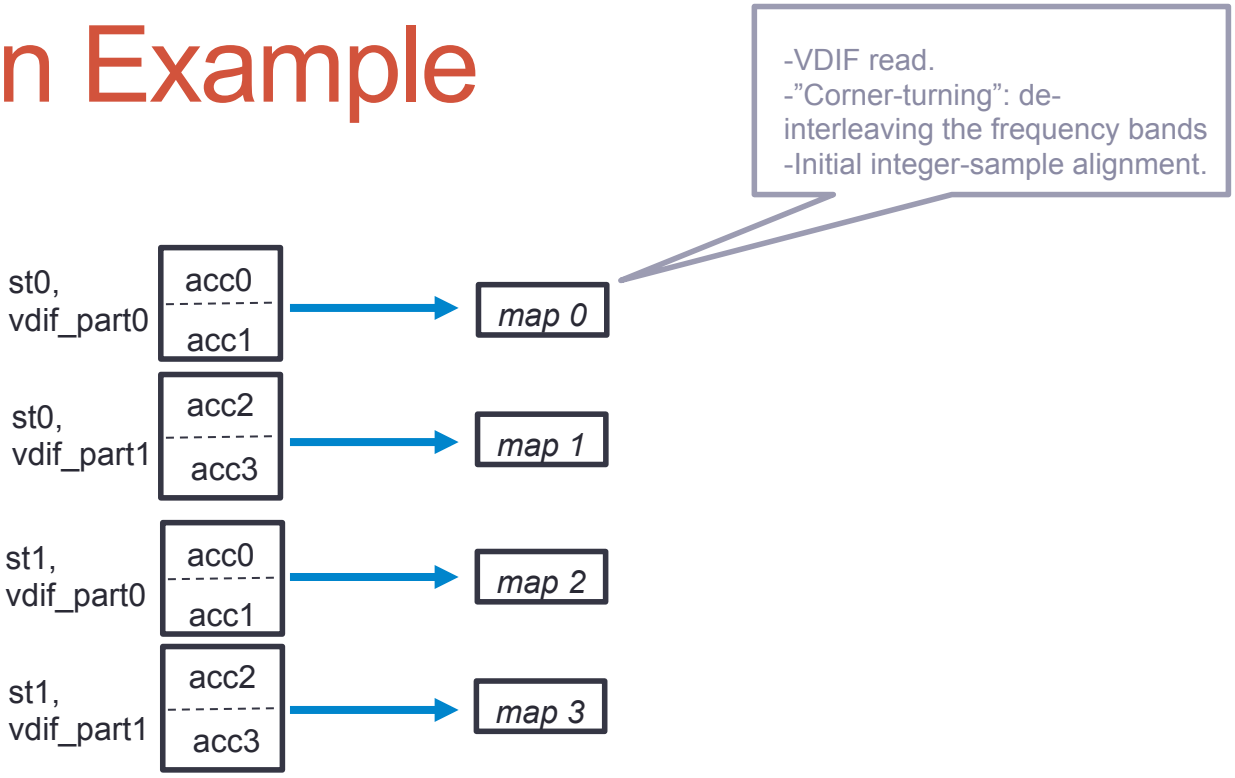
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.

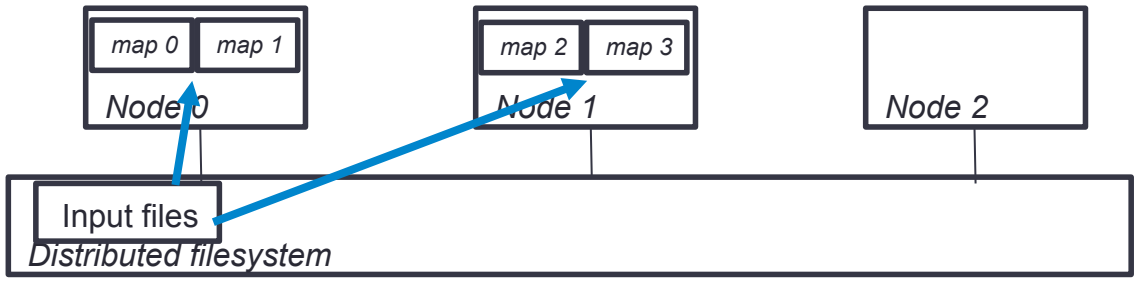


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

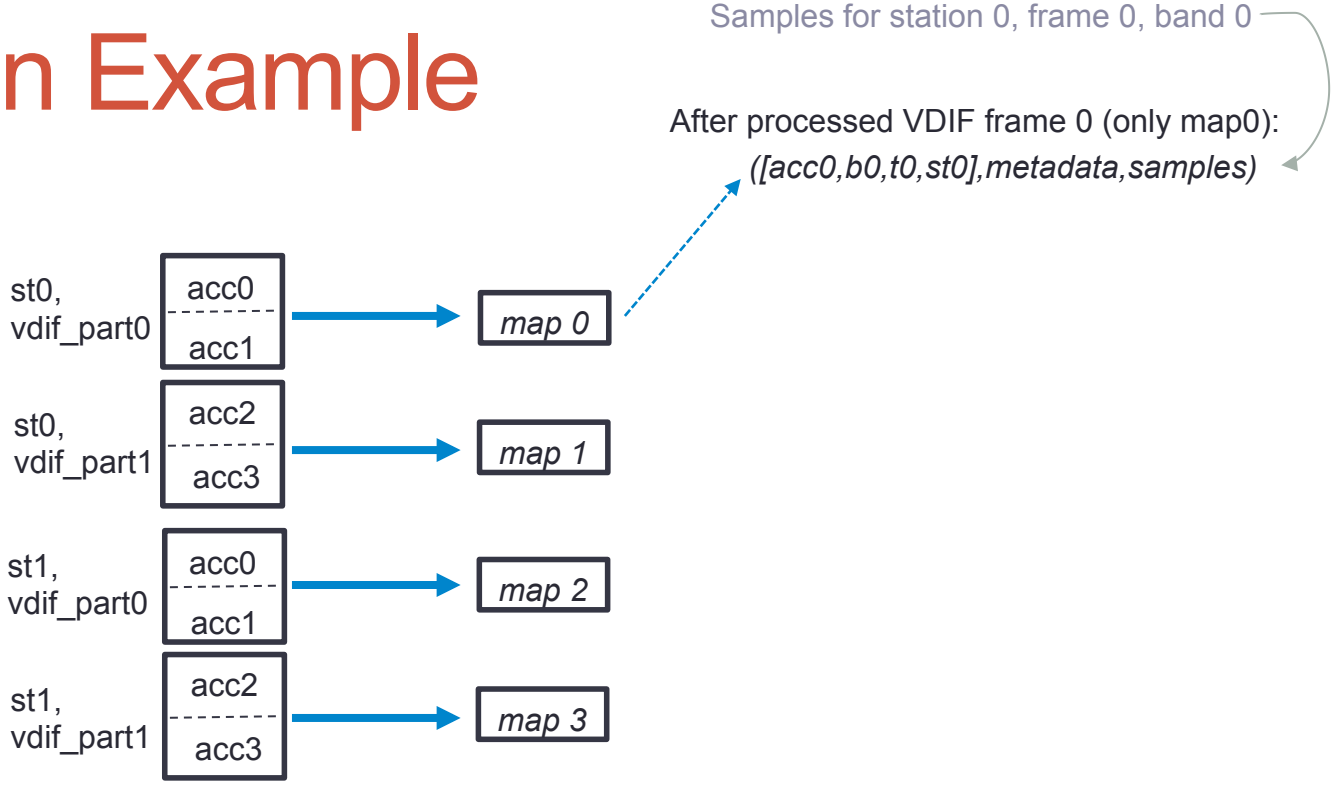
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

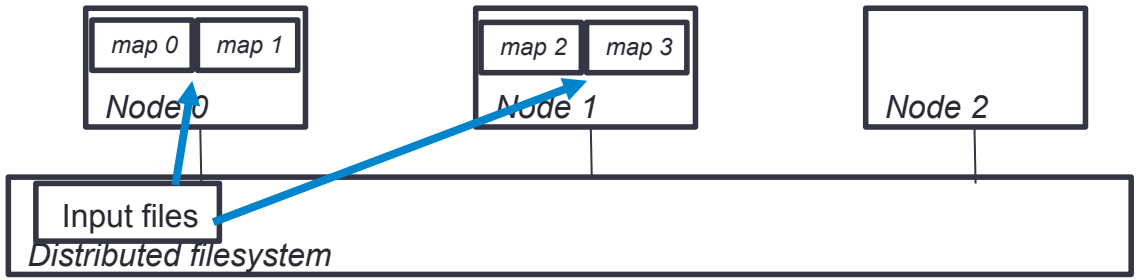


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

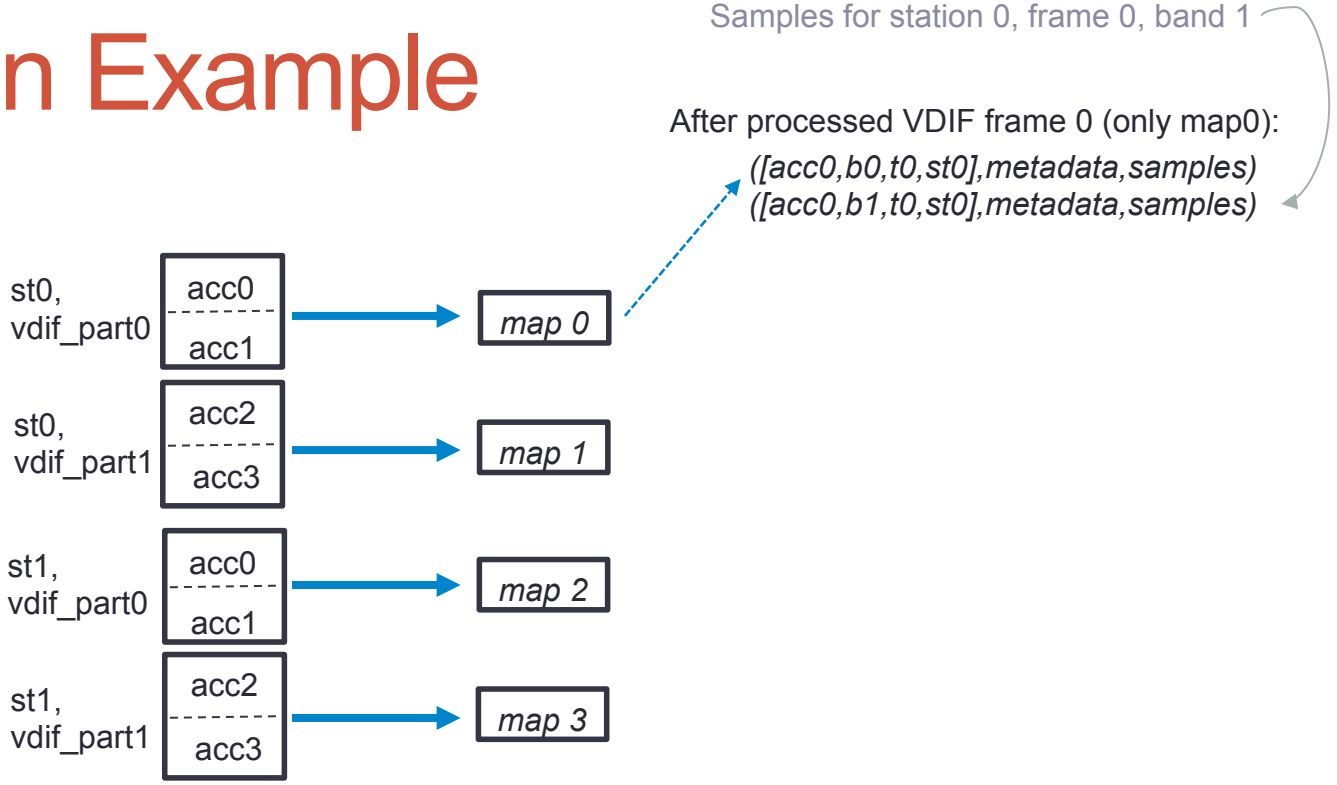
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

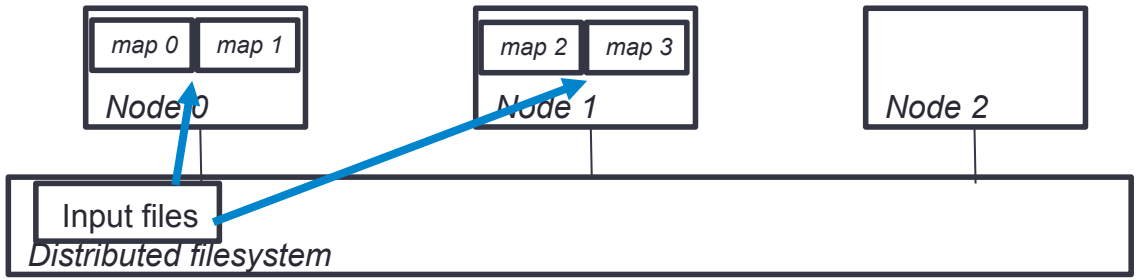


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

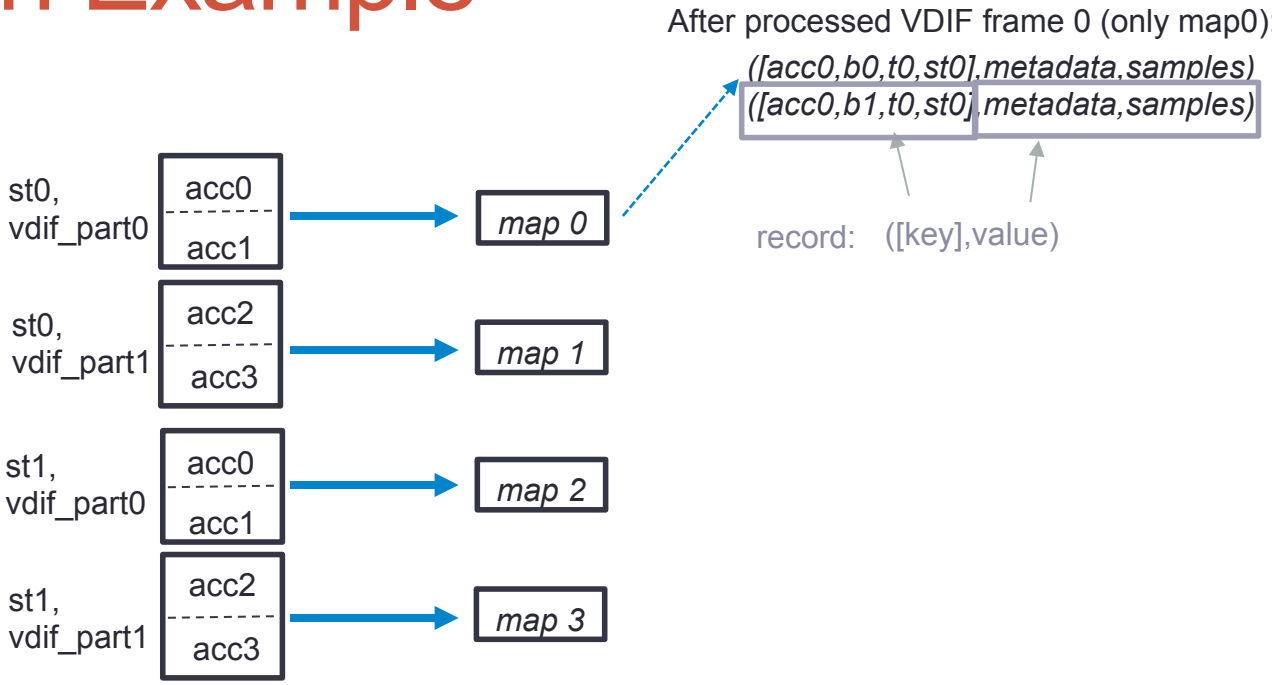
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

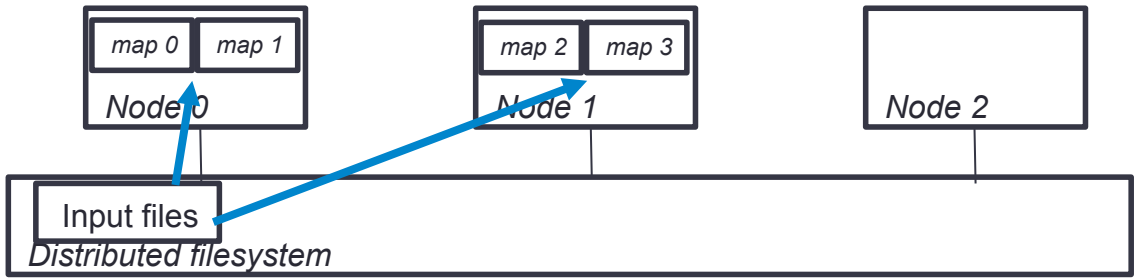


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

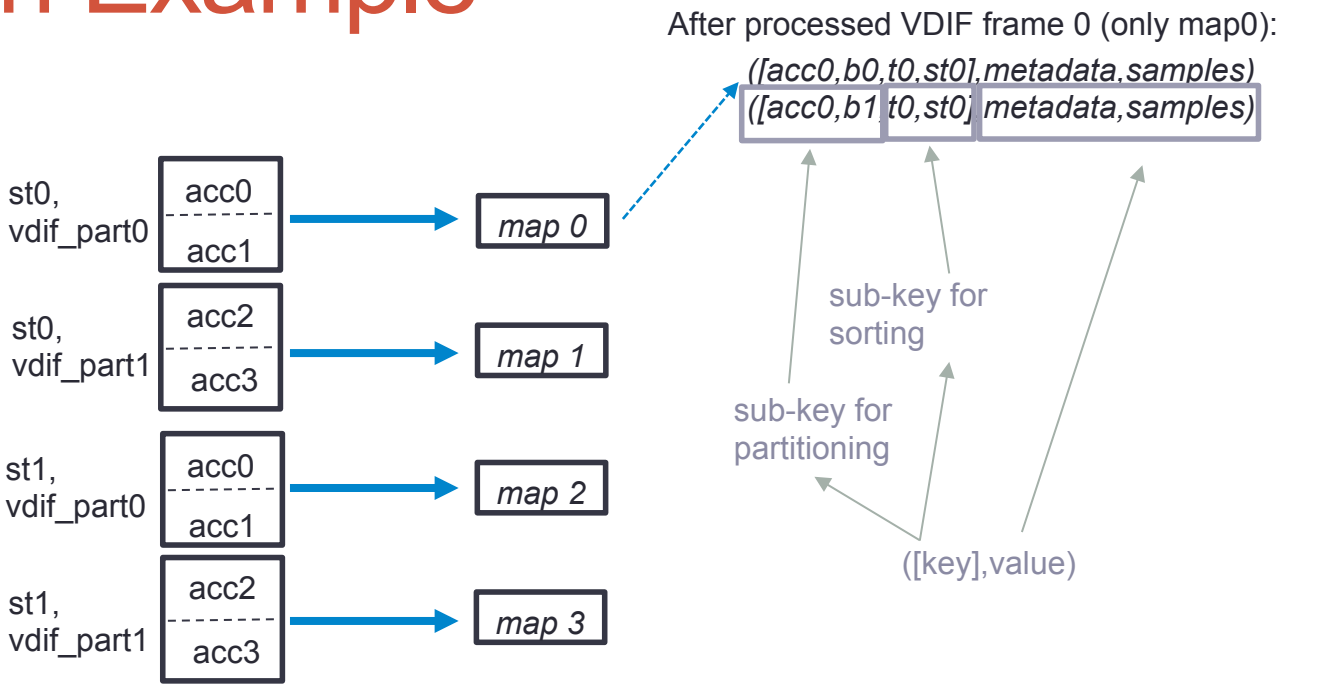
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.



Mapreduce description

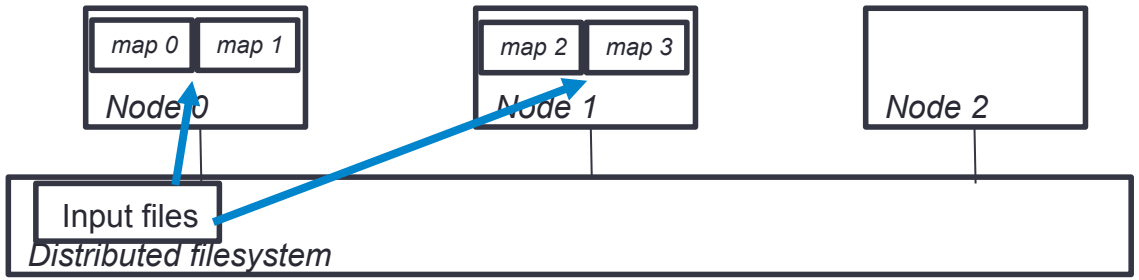
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

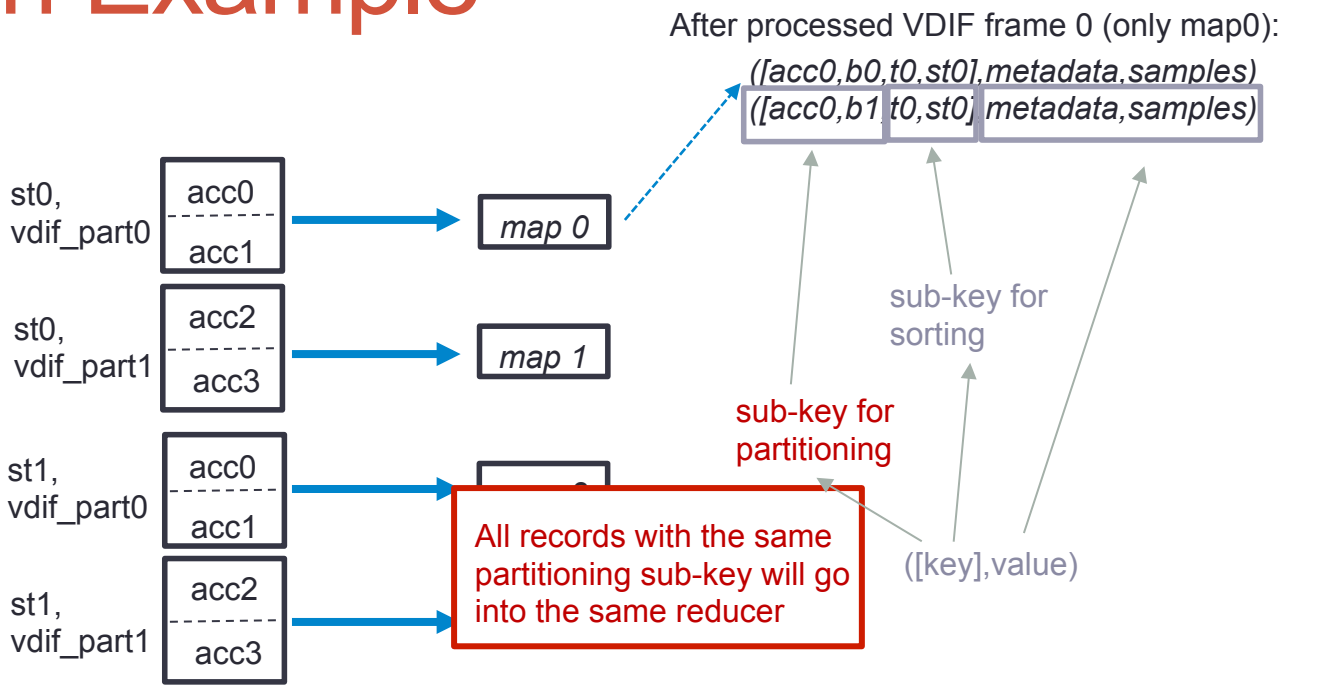
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

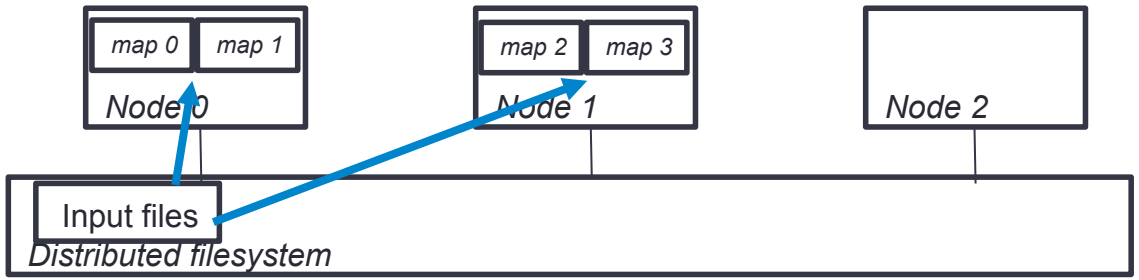


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

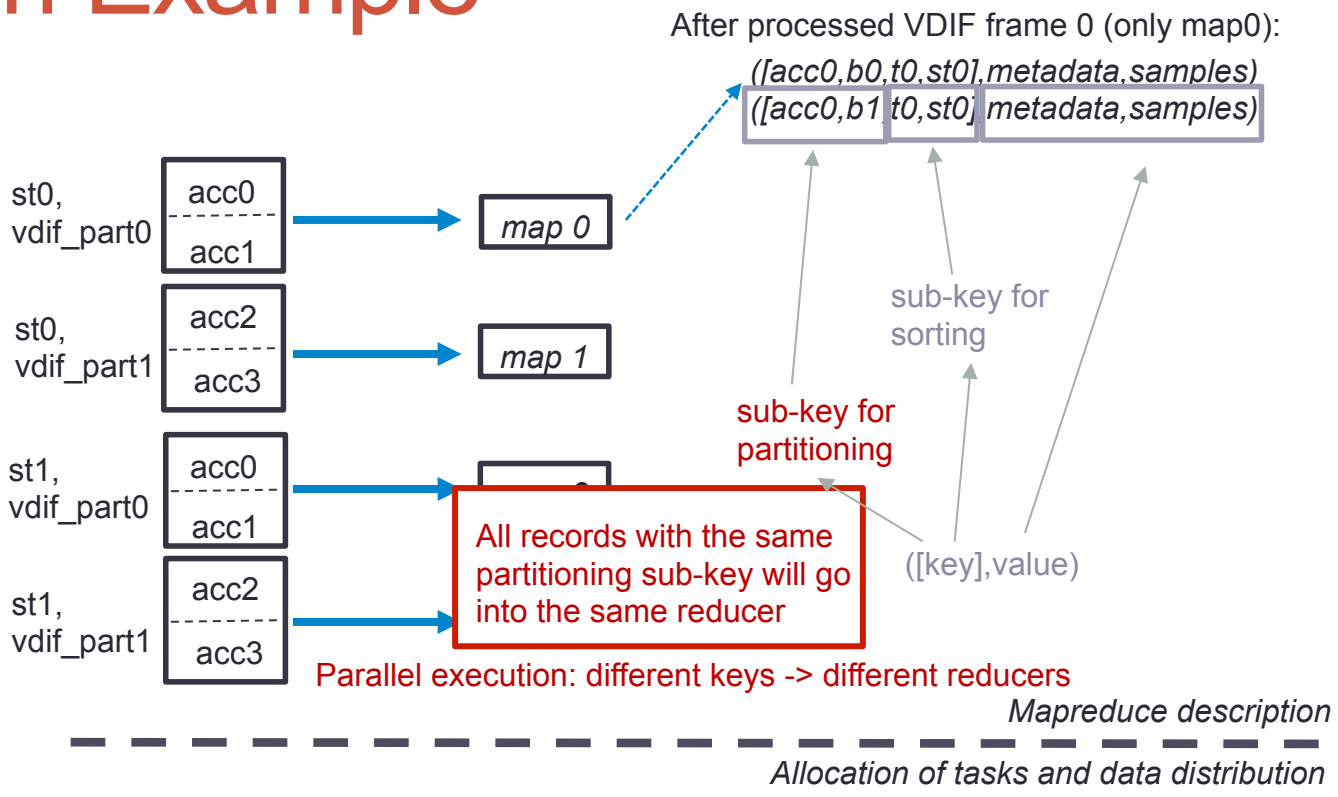
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

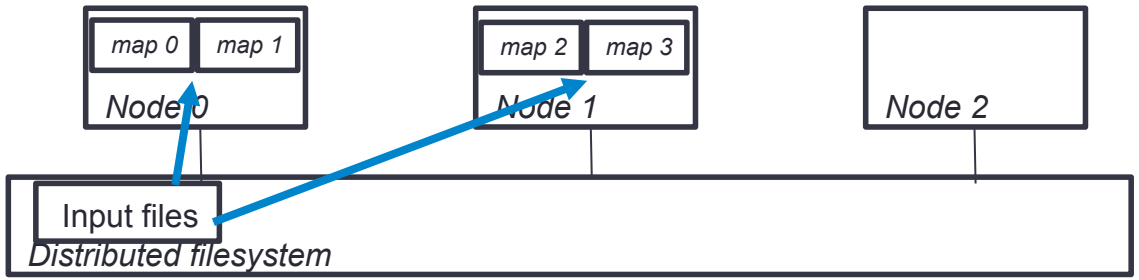


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

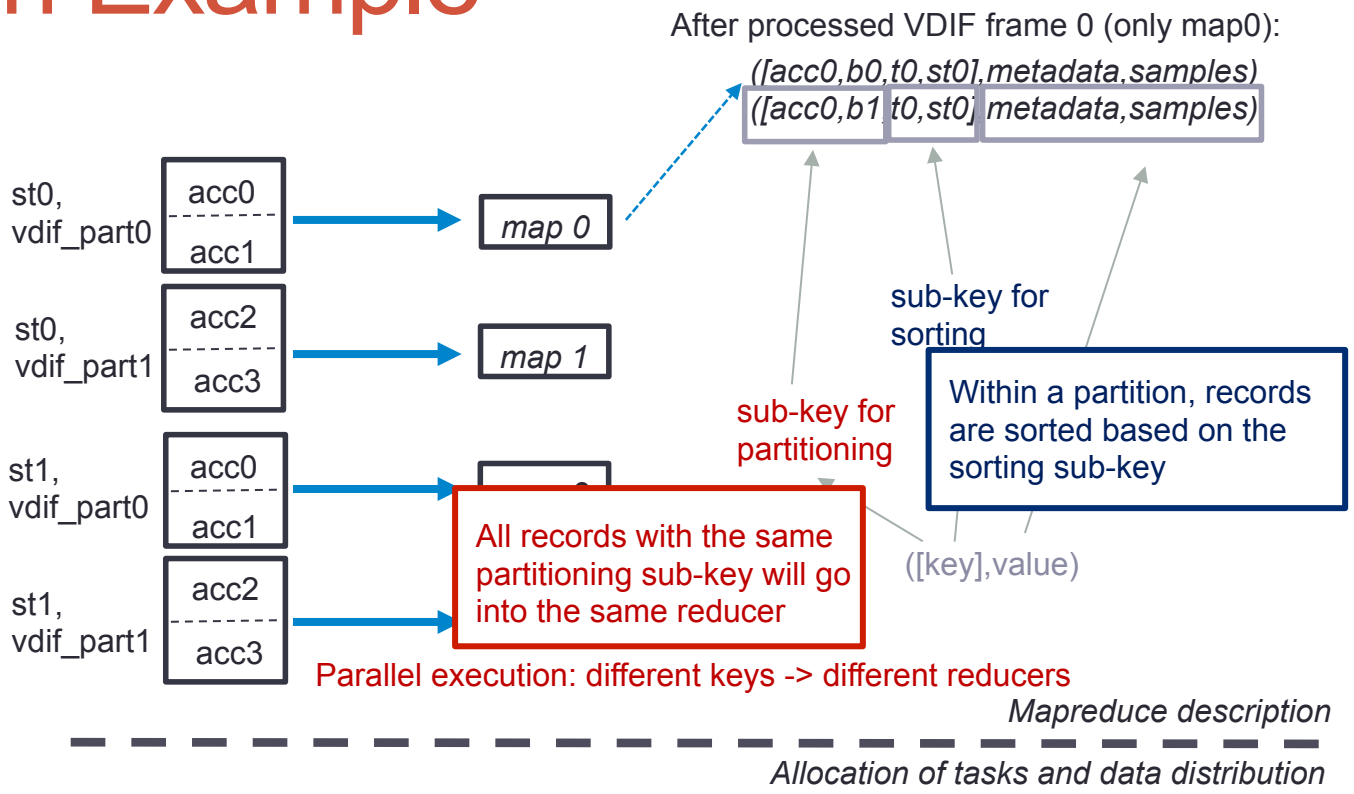
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

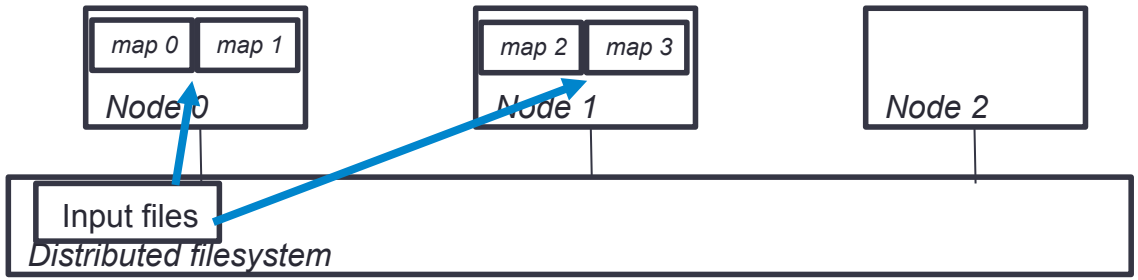


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

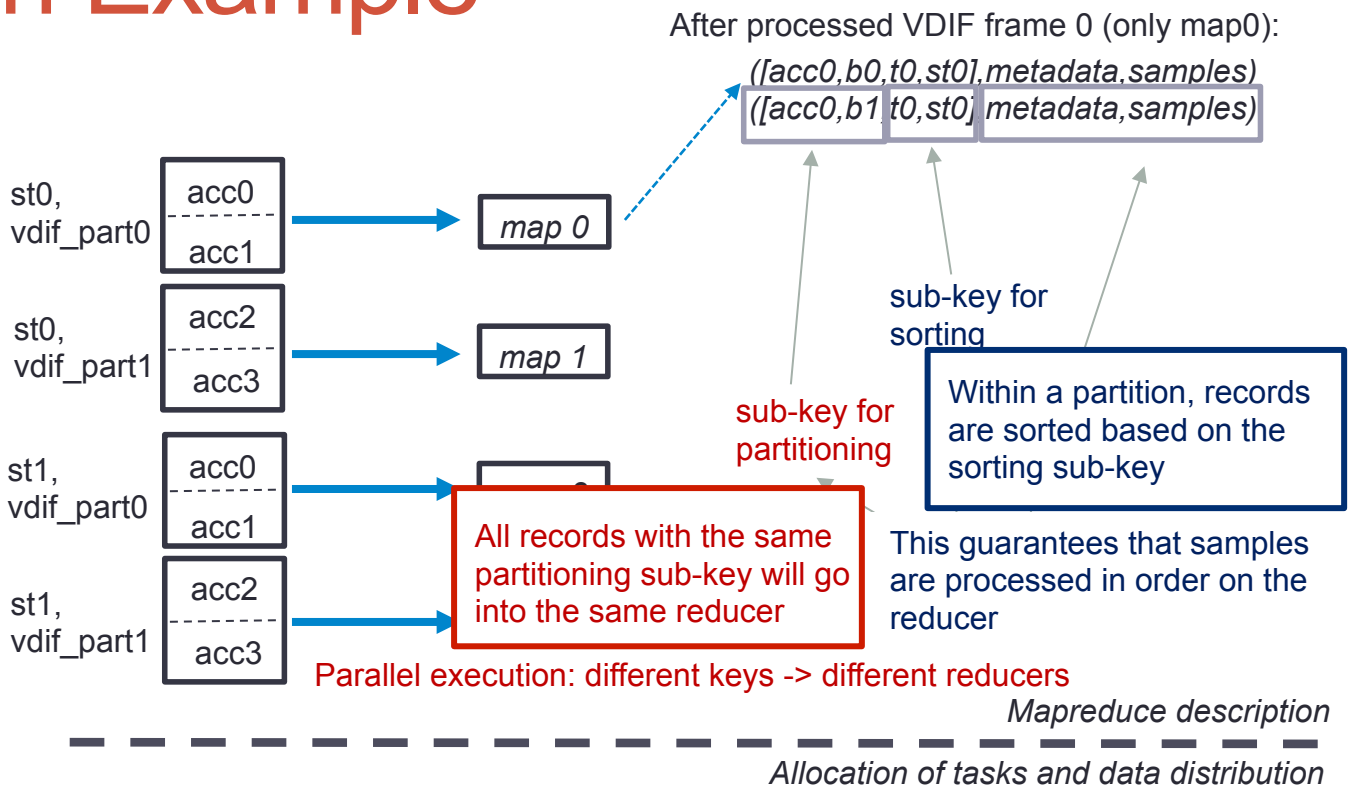
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

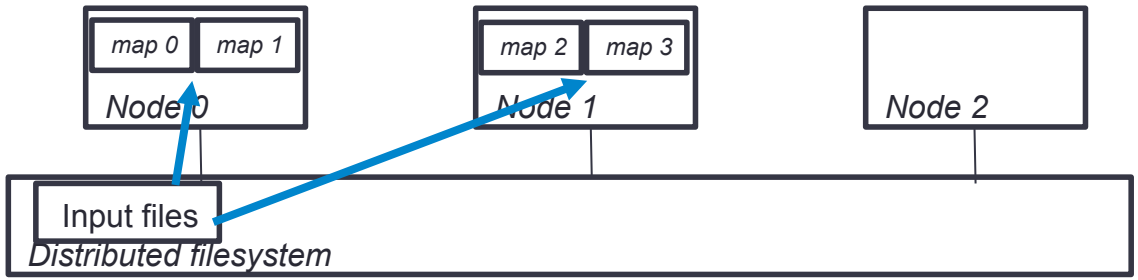


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

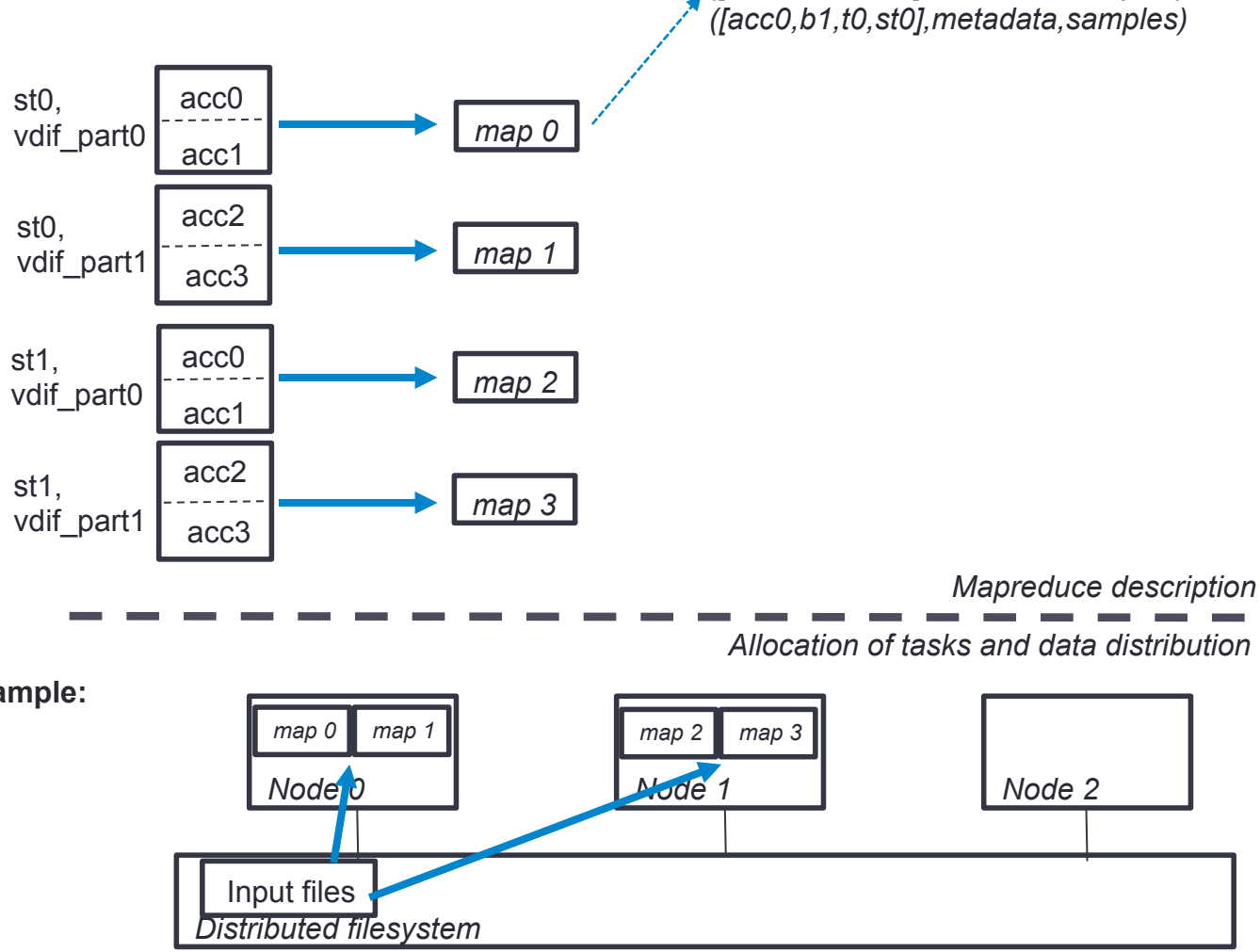
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.



MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

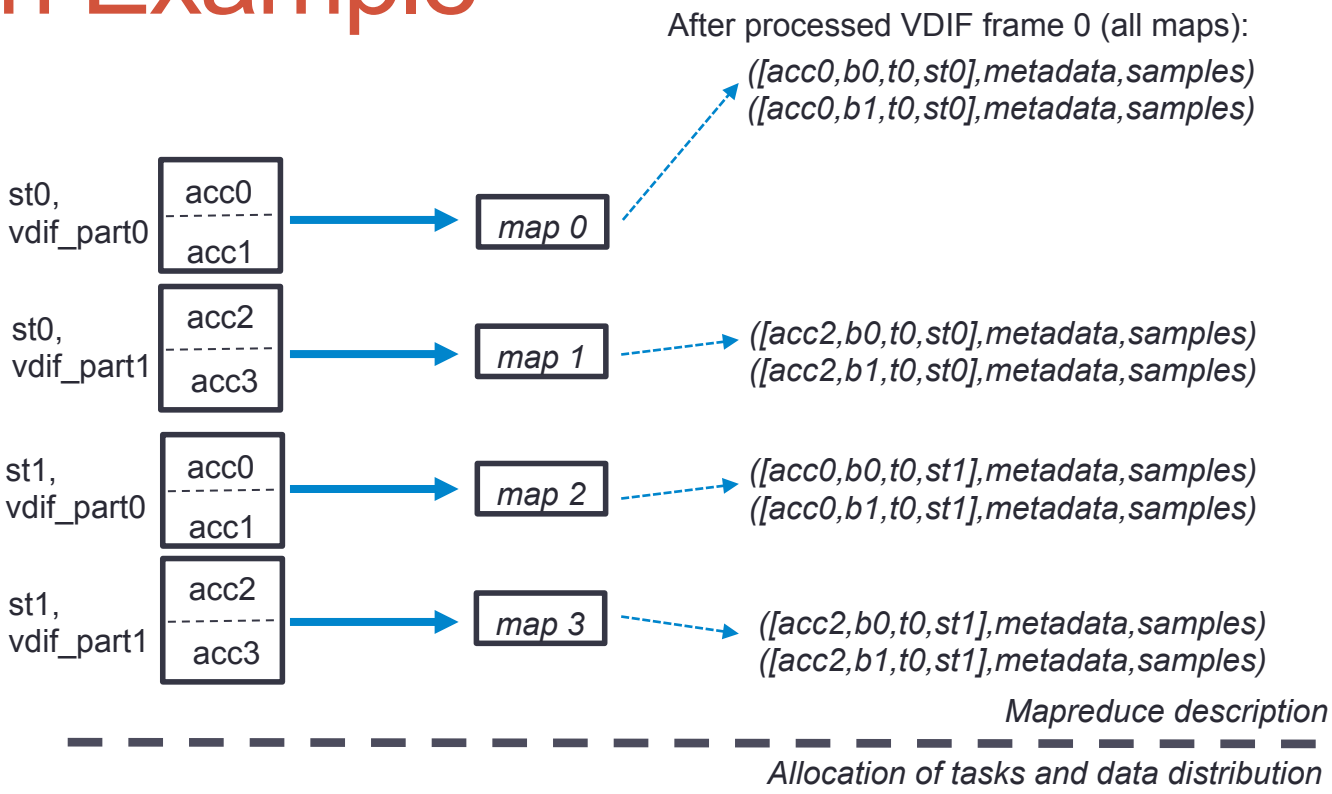
Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

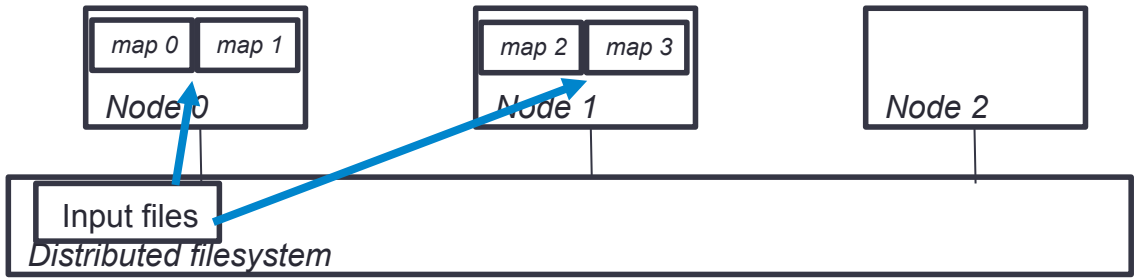


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

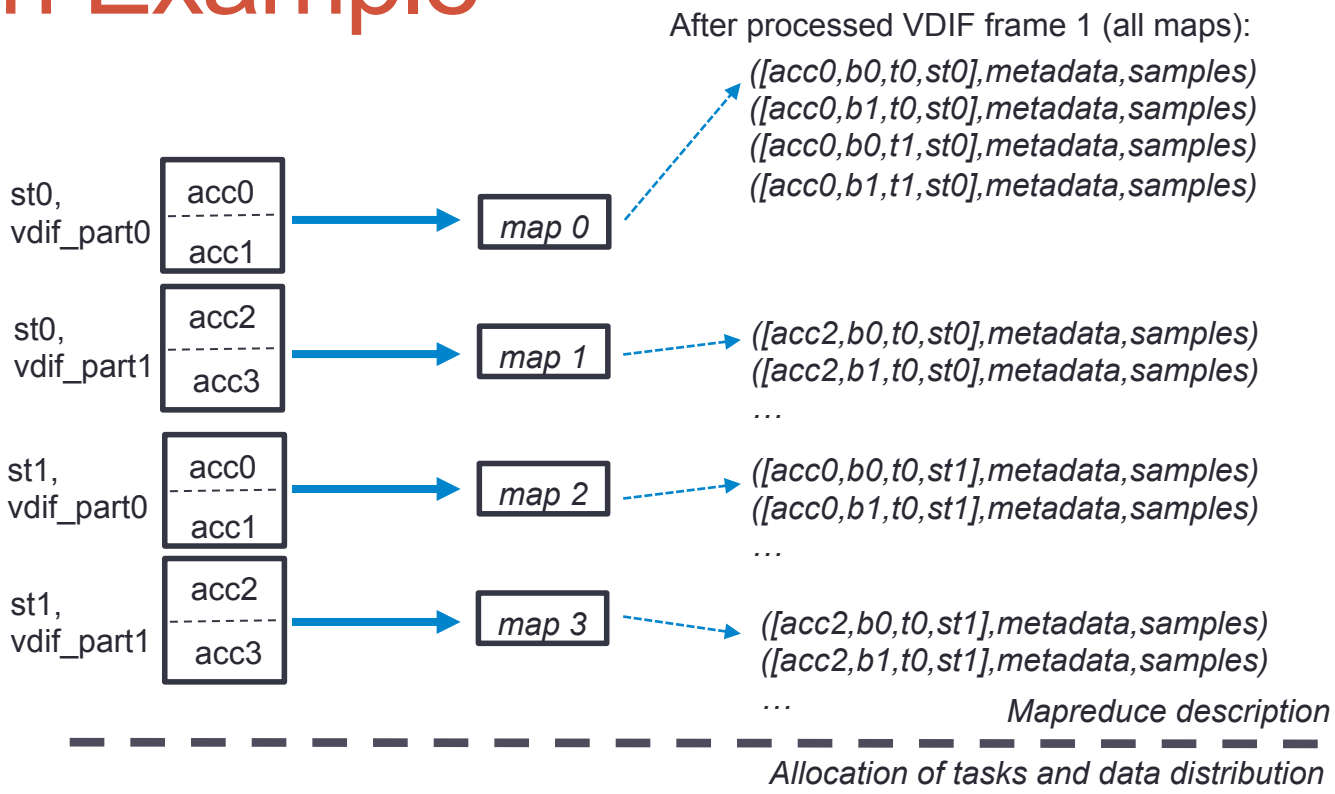
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

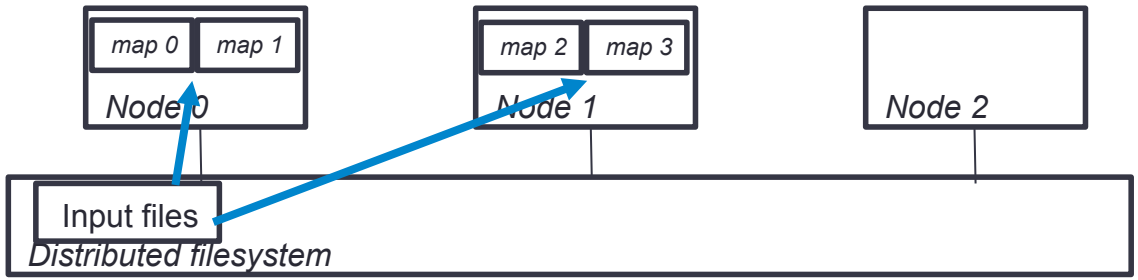


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

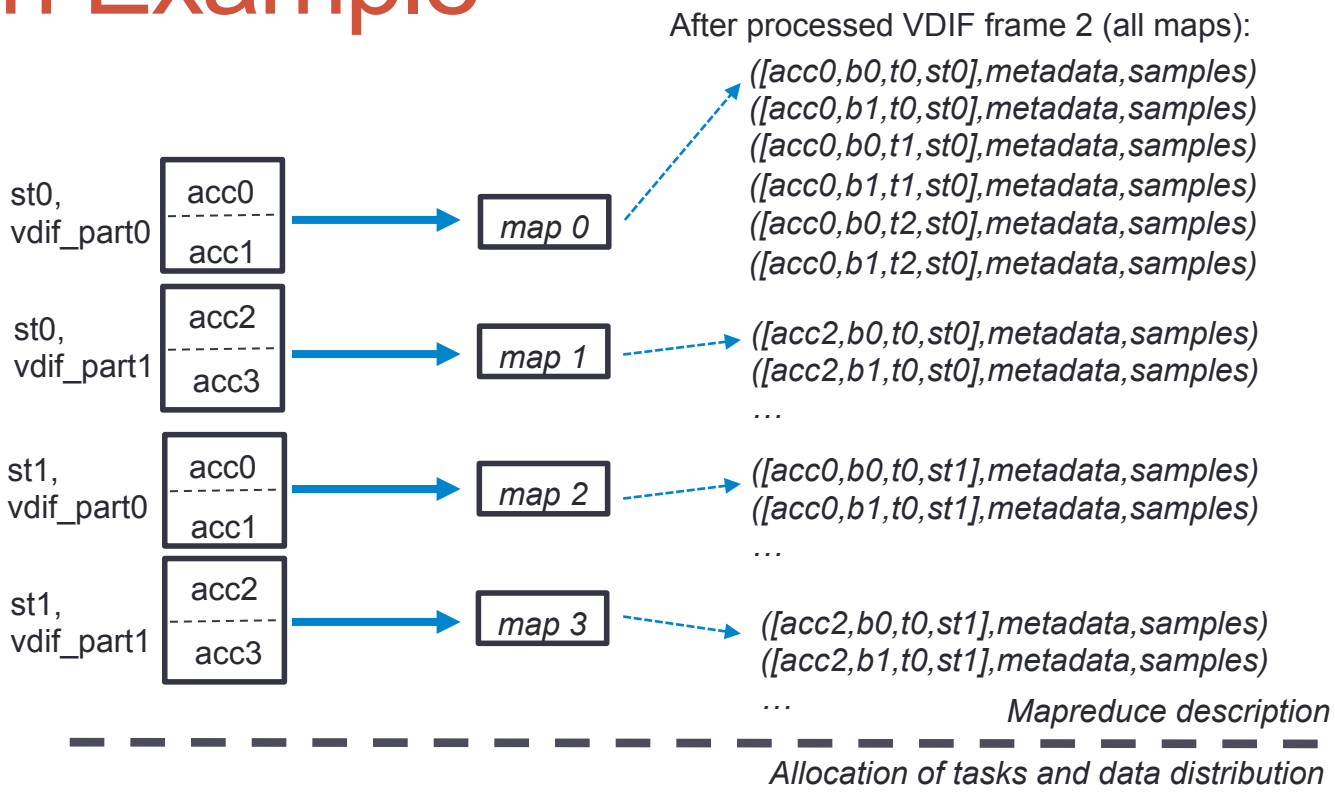
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

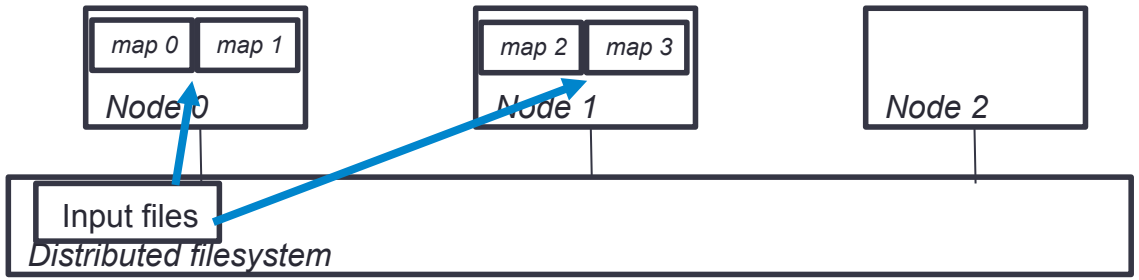


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

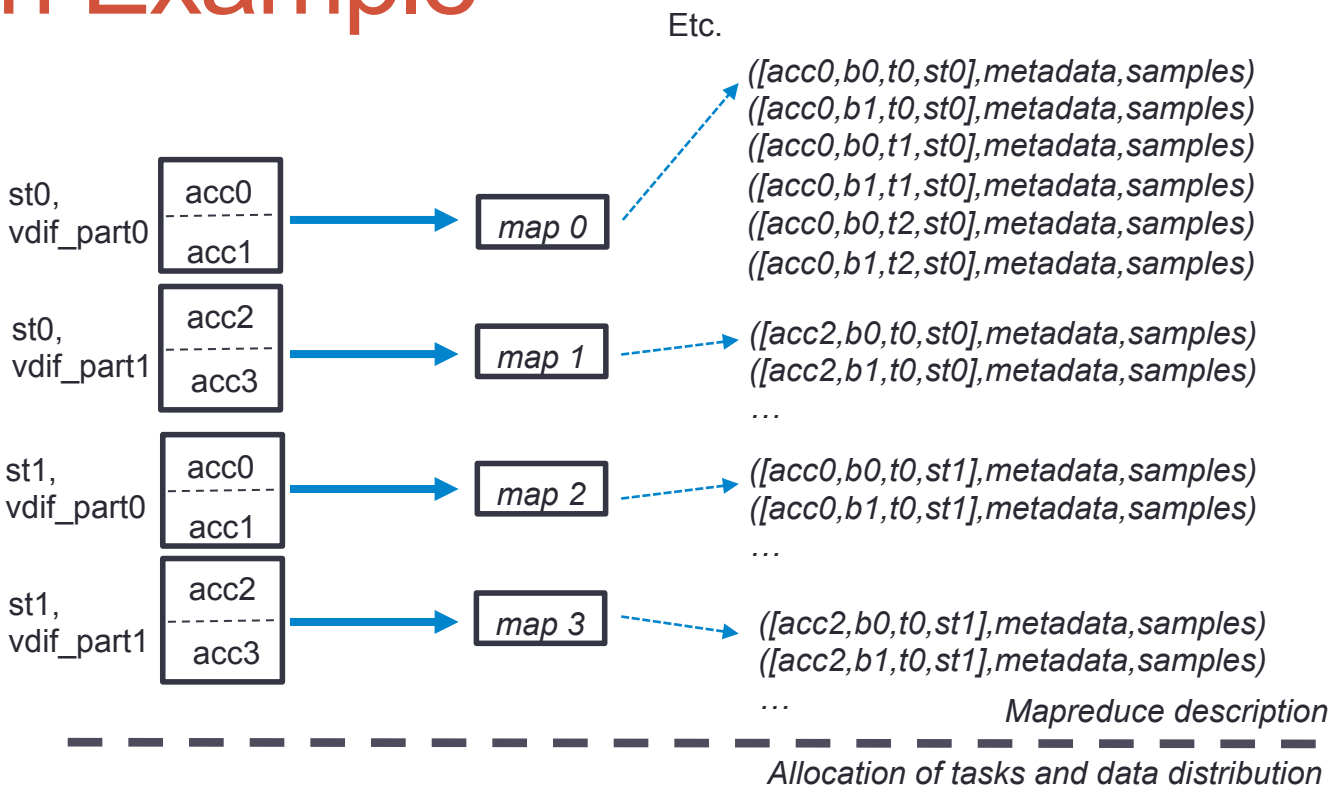
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.

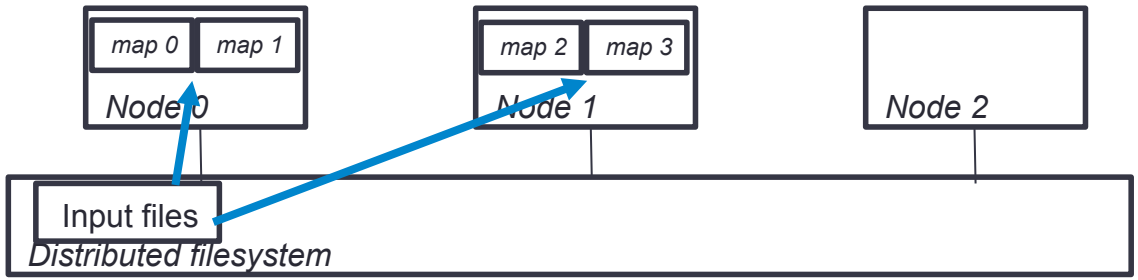


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.

Mapreduce description

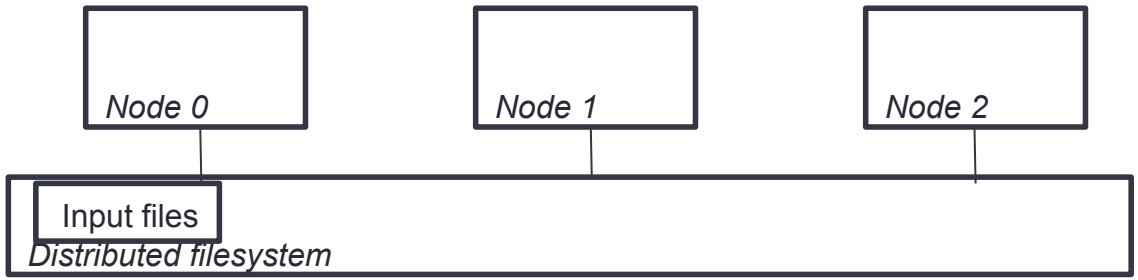
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

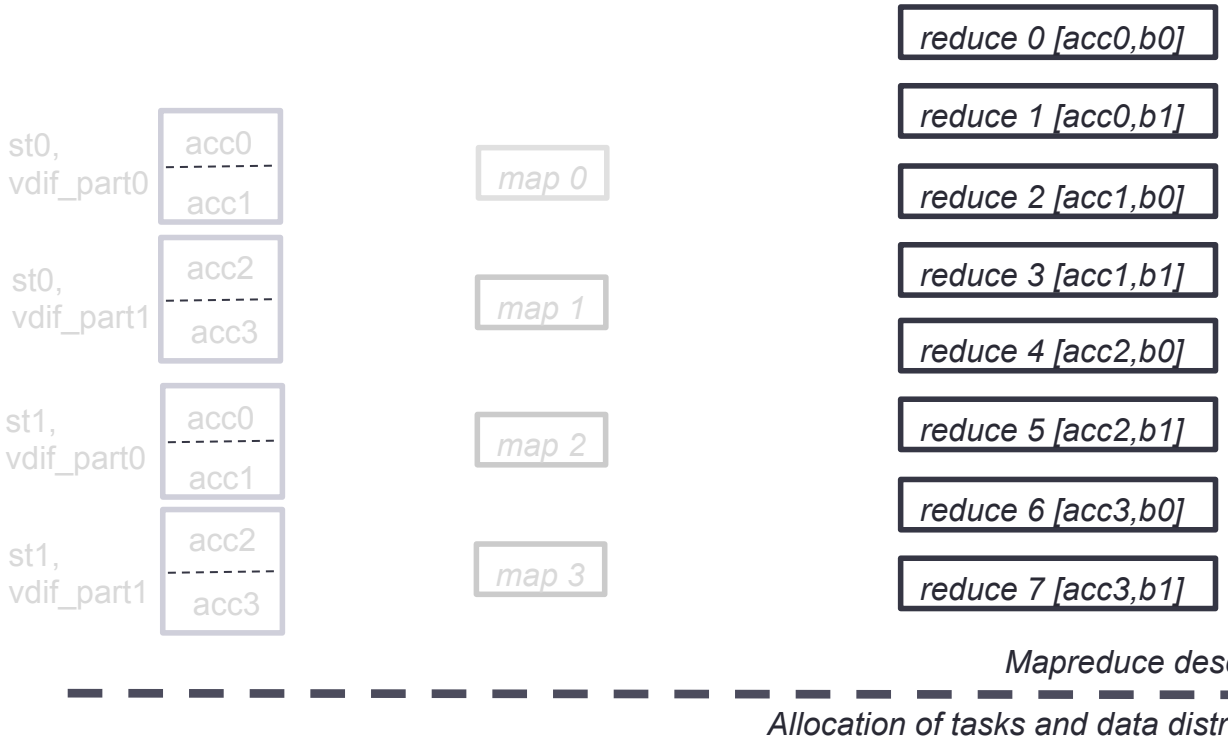
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.

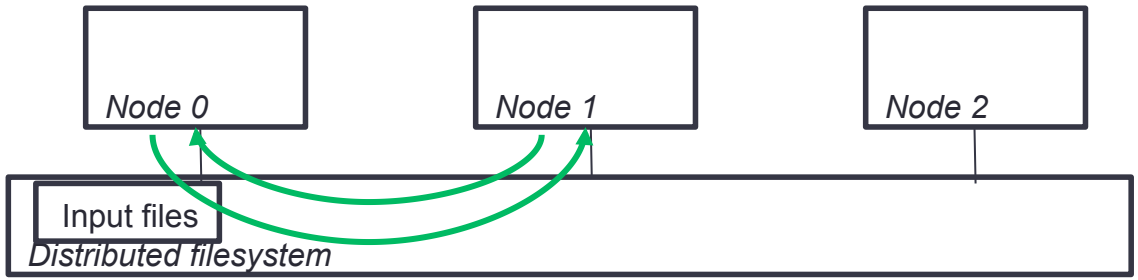


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

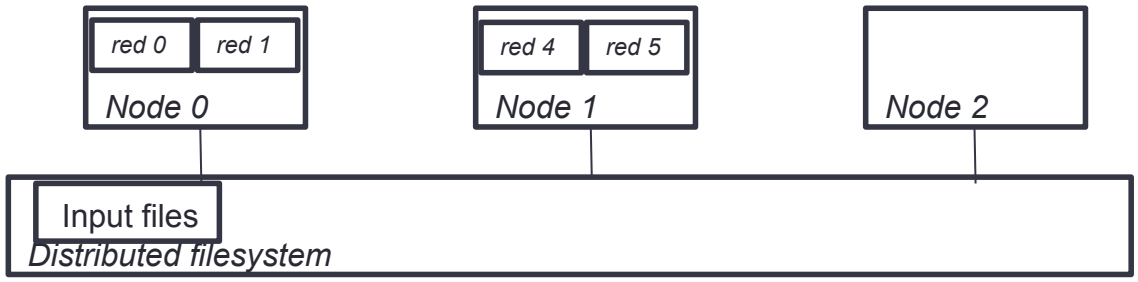
1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

-Fractional sample shift.
-Fringe rotation.
-Fourier transform.
-Fractional sample correction.
-Multiply and accumulate.

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.

Mapreduce description

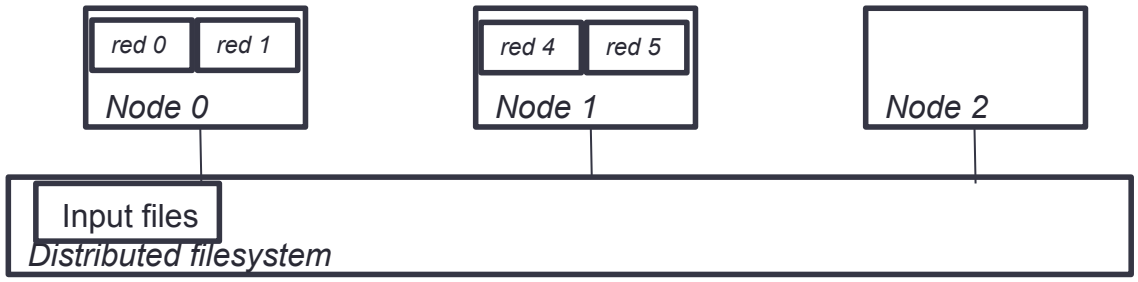
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8

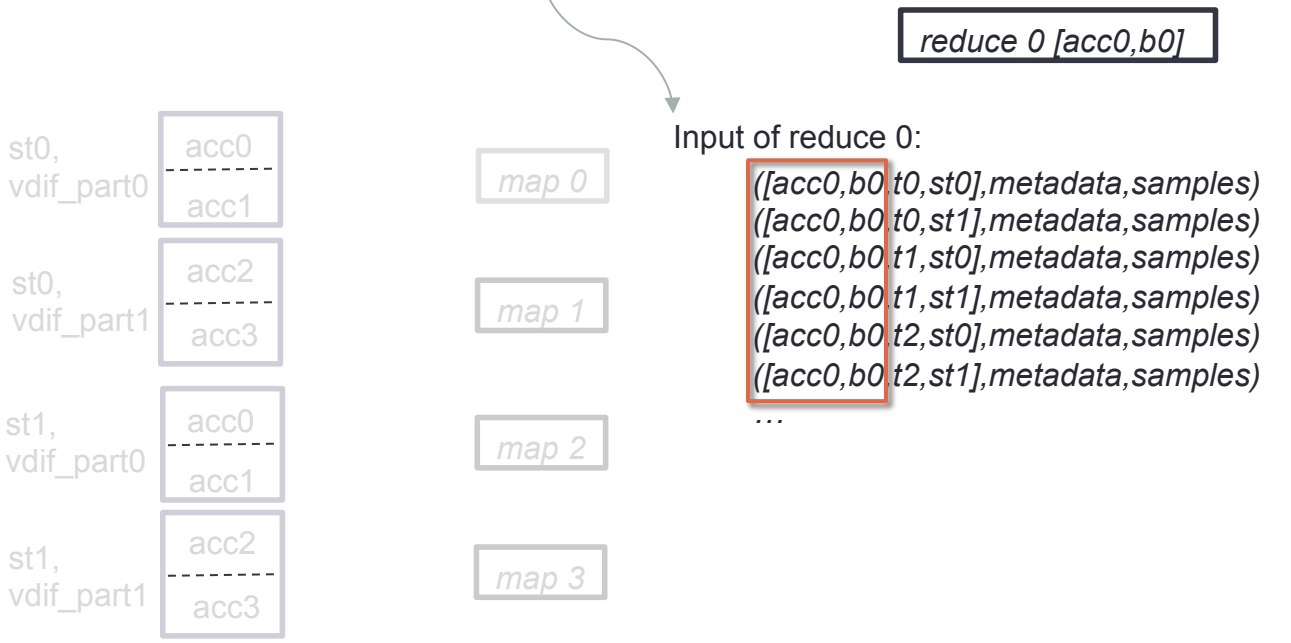


MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

Samples for all stations, acc. period 0, band 0

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.



Mapreduce description

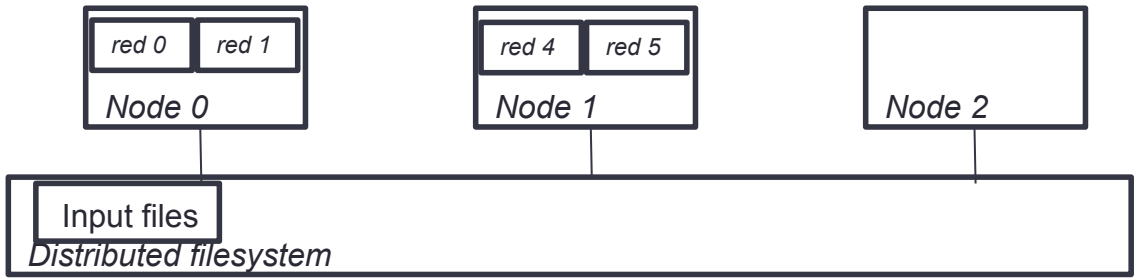
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

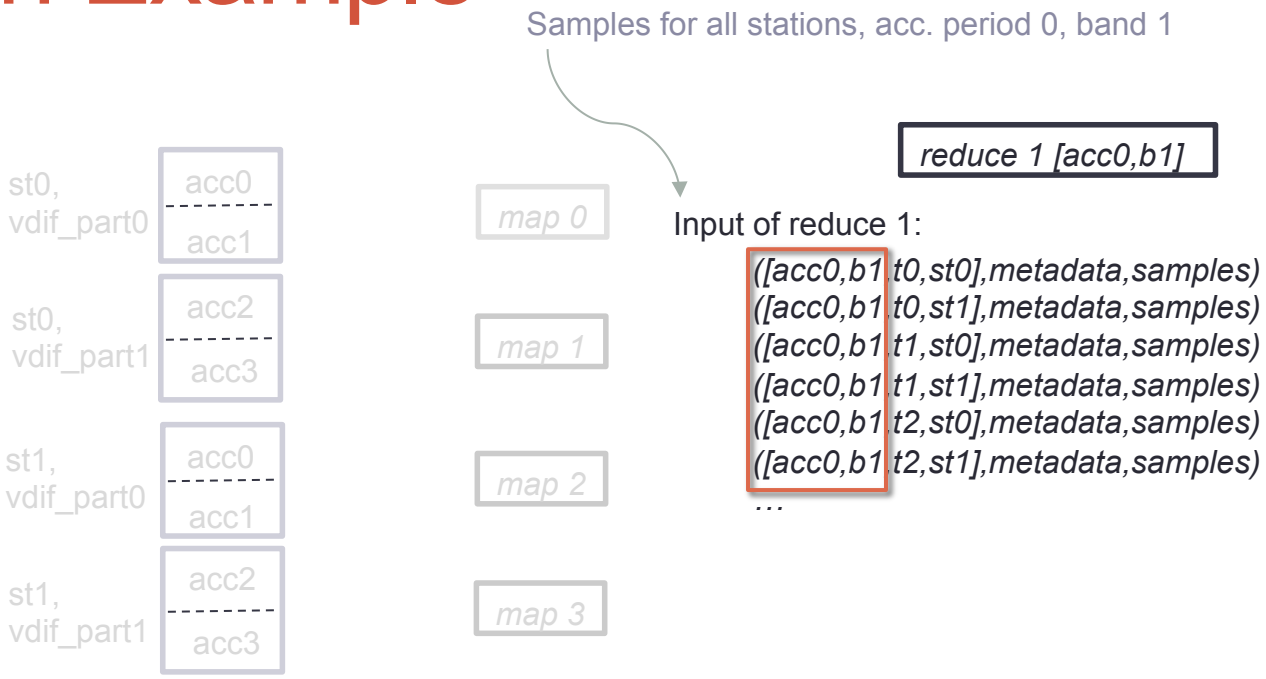
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.



Mapreduce description

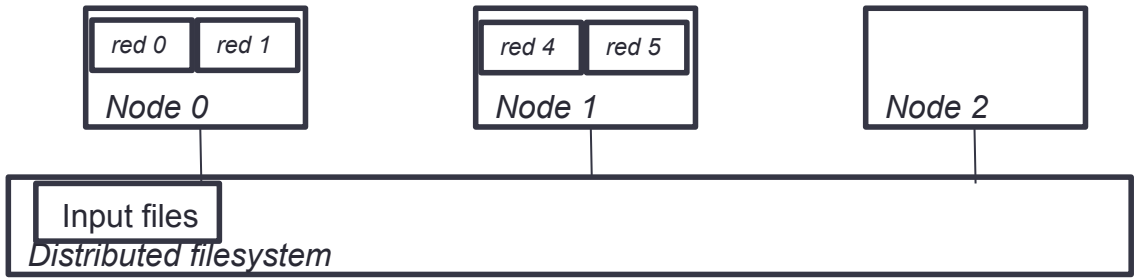
Allocation of tasks and data distribution

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

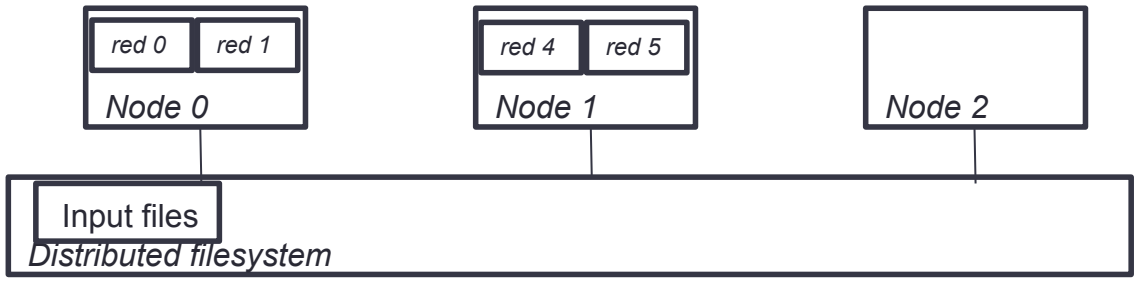
1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

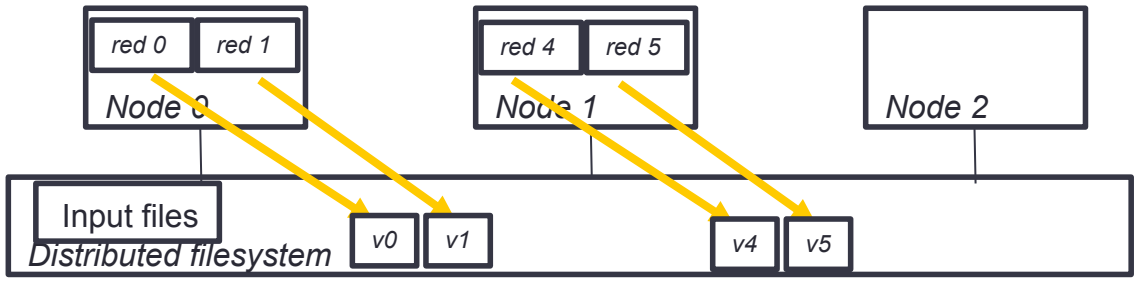
1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.
6. Generation of partial results.

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

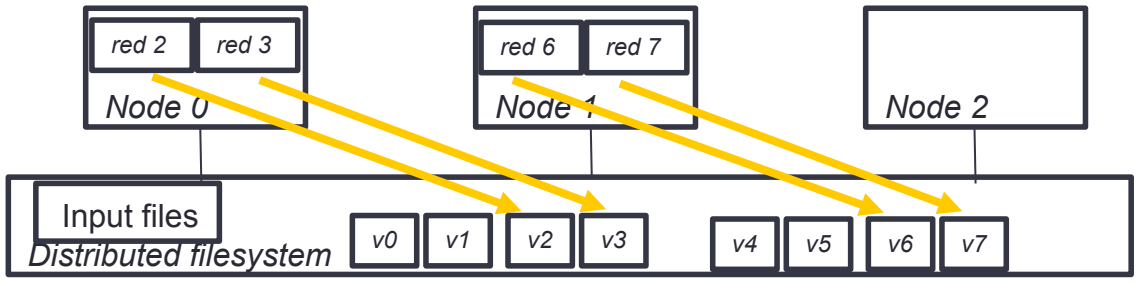
1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.
6. Generation of partial results.

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

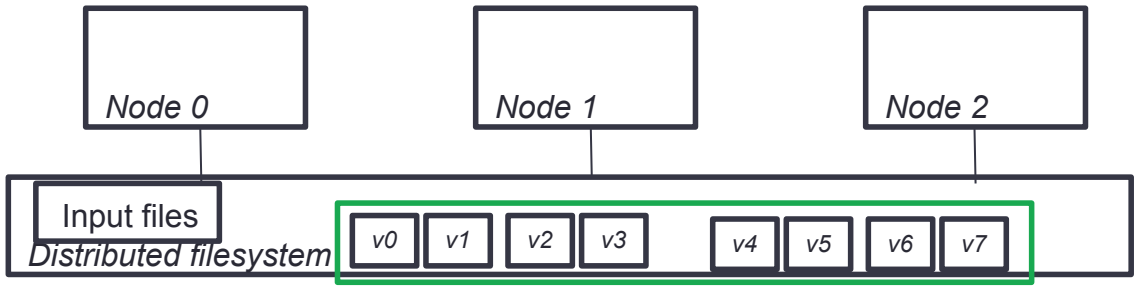
1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.
6. Generation of partial results.
7. Merge results.

MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

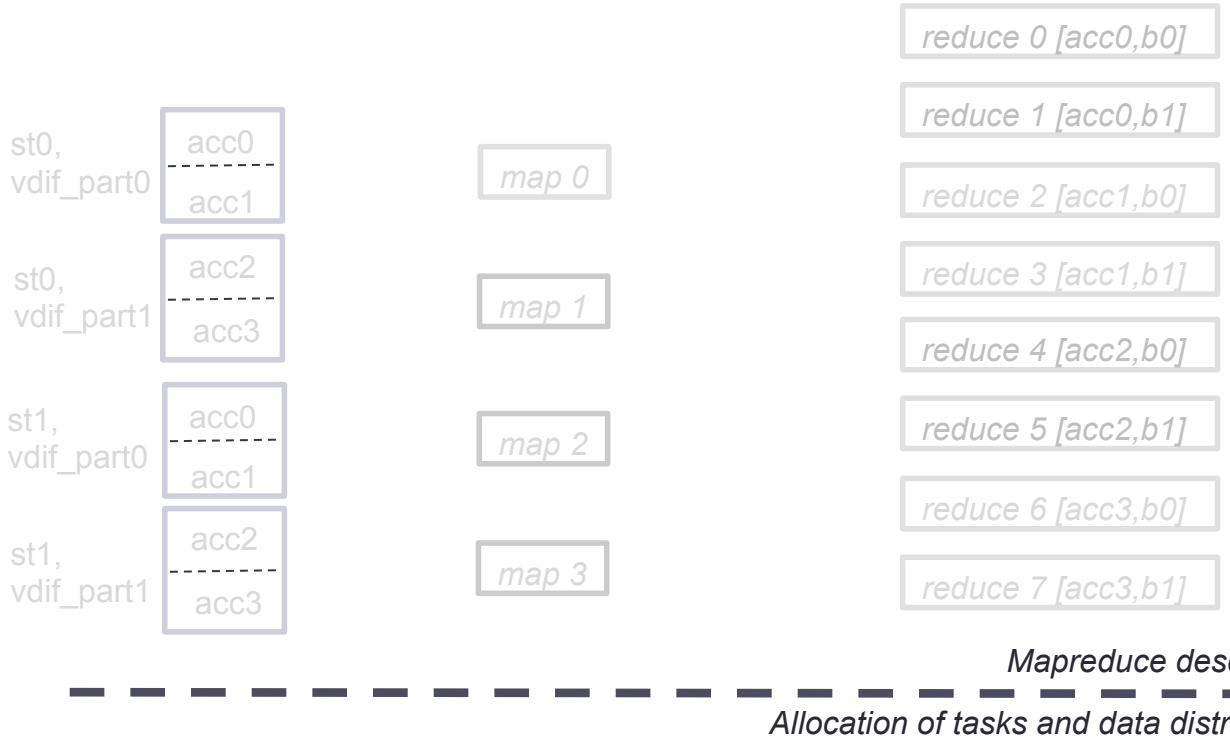
Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Correlation Example

1. Move input data into distributed filesystem.
2. The framework launches one map per input file.
3. Each mapper generates records with keys associated with reducers.
4. The framework distributes and sorts the records.
5. The framework will launch one reducer per partition.
6. Generation of partial results.
7. Merge results.

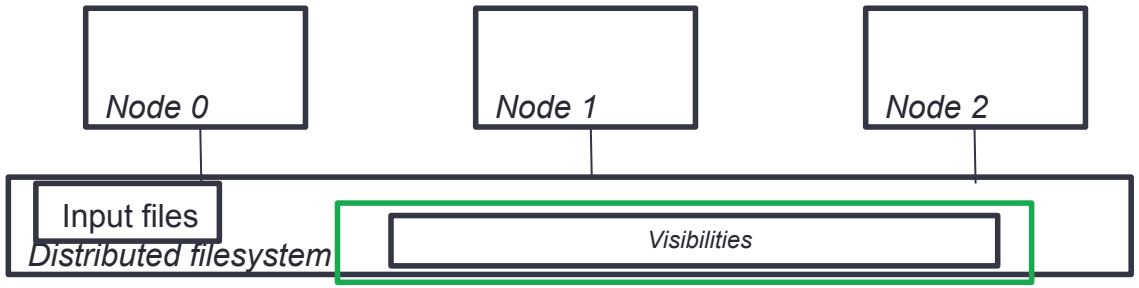


MapReduce correlation example:

Number of nodes: 3
Number of cores per node: 2

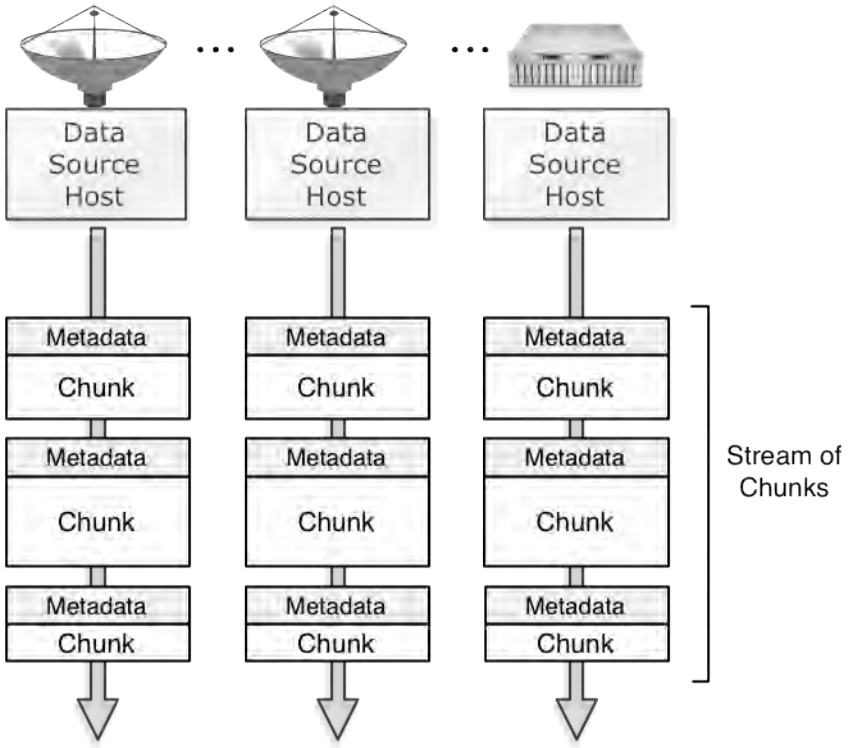
Stations: 2
Input splits: 2 per file.
Acc. Periods: 4.
Bands: 2.

Number of mappers: [sts]*[splits]=4
Number of reducers: [accs]*[bands]=8



MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

CorrelX: Chunk Streaming Model

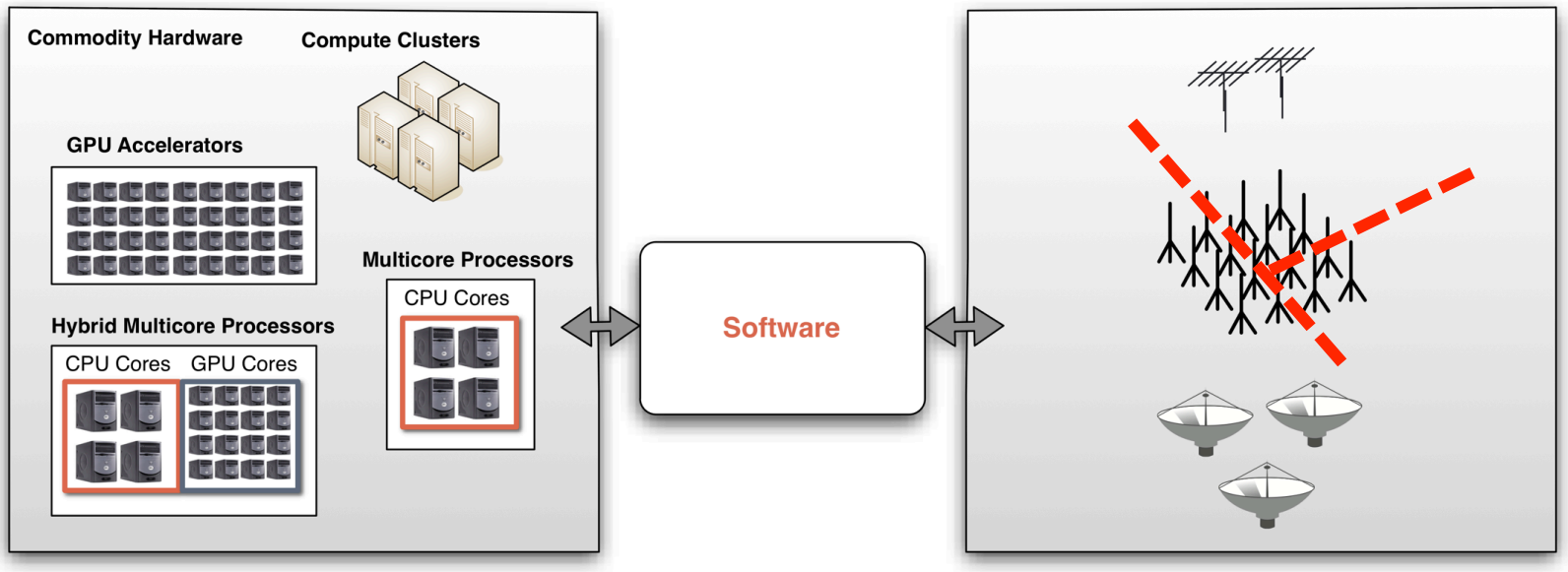


- Chunks introduce locality → out-of-order parallel processing
- Enables tunable chunks (sizes, sampling, numerical precision, etc.)
- Insert control chunks with known results for validity checks on heterogeneous or unknown hardware

MIT HAYSTACK OBSERVATORY Massachusetts Institute of Technology

Perspectives

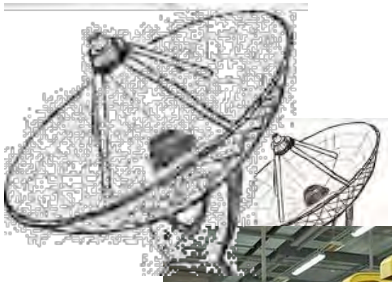
Chunk model enables partitioning of large instrument arrays to work in different modes at the same time.



Chunk model enables execution on heterogeneous hardware. Different hardware → different reducers for chunks. Will use machine learning for performance tuning of chunk execution on each platform.

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology

Trends: Software-Based Instruments & Cloud Computing



Cloud Computing

- Signal Processing
- Algorithms
- Imaging
- Search & Classification
- Simulations
- Data analysis, mining, retrieval

MIT HAYSTACK OBSERVATORY
Massachusetts Institute of Technology