HadoopBase-MIP

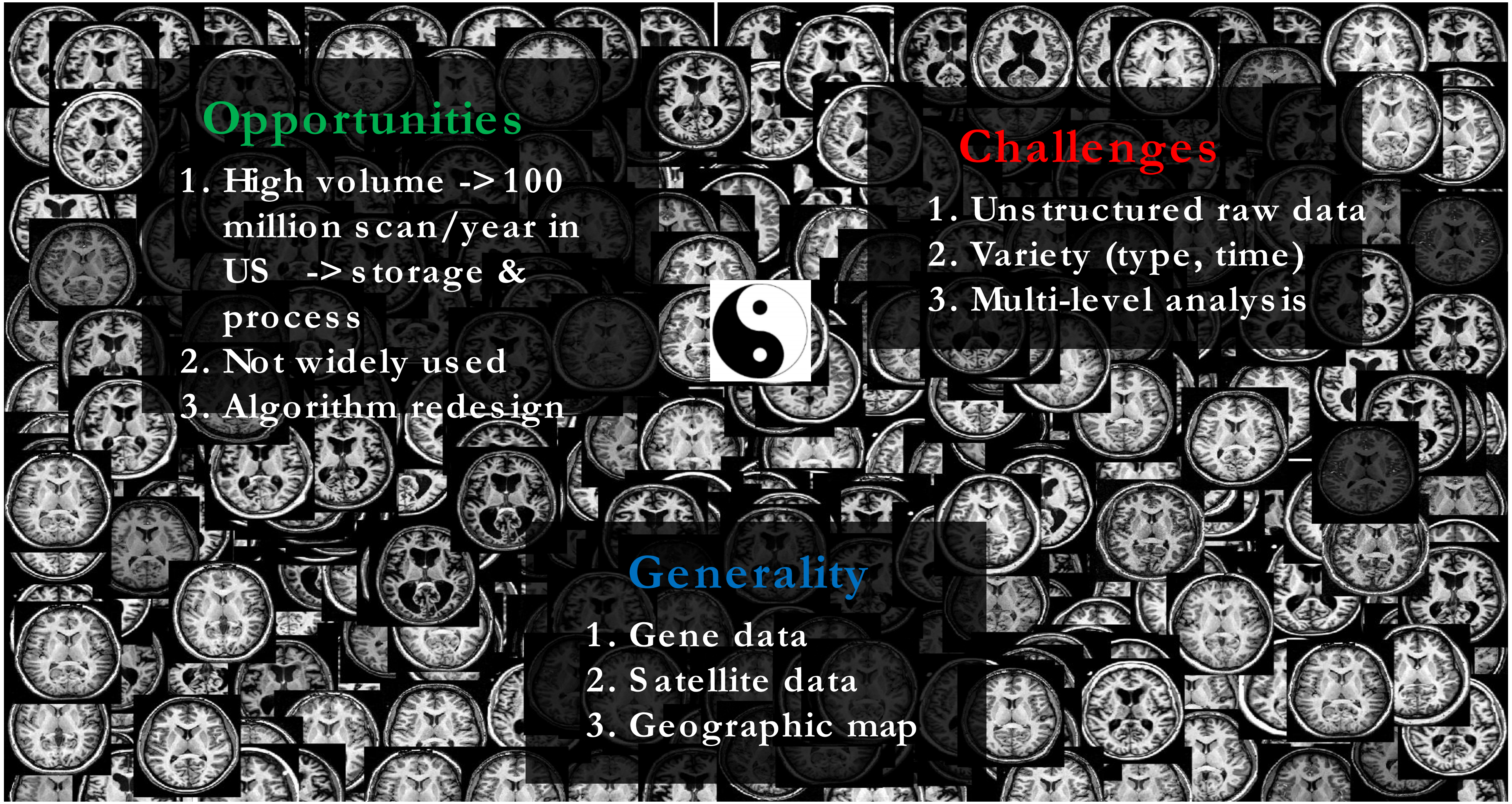
Hadoop & HBase - based toolkit for medical image processing

Shunxing Bao

Nov 2018

Big data framework for medical image processing

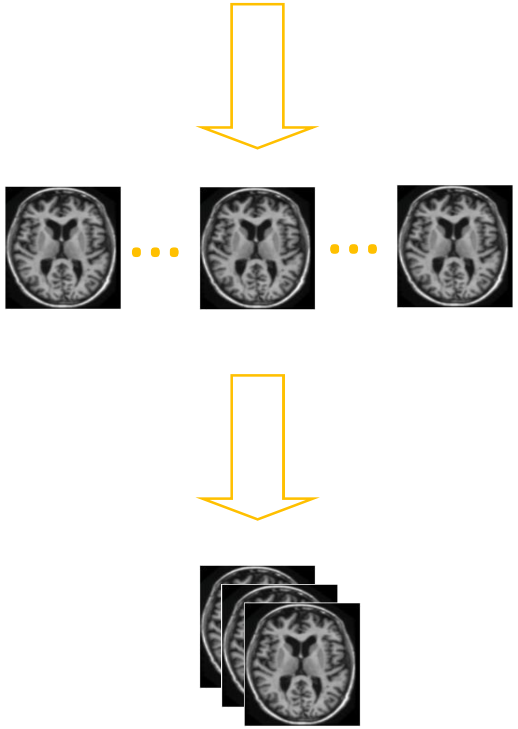
ACCRE
Advanced Computing Center for Research & Education



Need for data colocation

- Convert Digital Imaging and Communications in Medicine (DICOM) to Neuroimaging Informatics Technology Initiative (NiFTI)
  - DICOM
    - Standard image formats for modern medical image equipments
    - 2-D slice
  - NiFTI
    - Research software file format
    - 3-D, 4-D
- 1st step in processing

CT MRI



DICOM

NiFTI

Why need data colocation cont’d

GRID ENGINE

Thousands of jobs

DICOM -> NiFTI

Cluster (300 cores, 1 Gigabit bandwidth)

Network

ge

Minimizing
data movement
network

Group of DICOM

1 gigabit network

$$x \cdot \frac{50\ MB}{15\ s} = 100\ \mathrm{MB/s} \Rightarrow x = 30$$

Conversion time

Max Simultaneous Processes

Background for data colocation

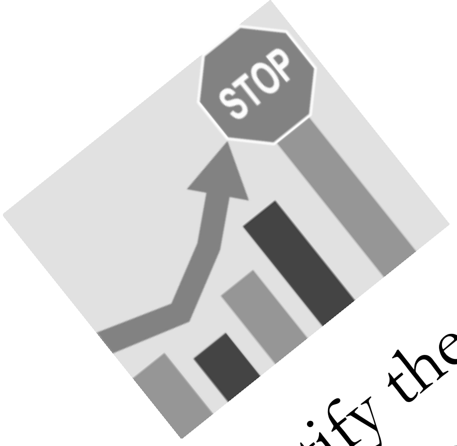
- **Hadoop**
  - Hadoop distributed file system (HDFS)
  - MapReduce – dispatch computations to data
  - Inefficient for large volume small data
- **HBase** NoSQL database
  - <key , value> store built upon HDFS
  - Logically physically sort the row key
  - Flexible translation layer – region split policy
- **Data colocation**
  - Hadoop & HBase colocate
  - Locate relevant data as close as possible
    - Group based analysis: project / subject / session / scan based

Outline

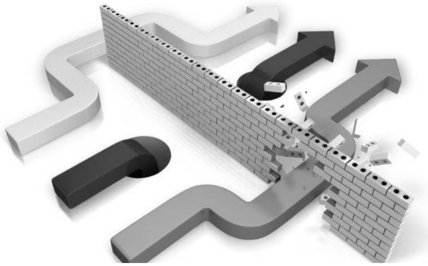
Data colocation grid prototype for Medical image processing-as-a-service

(1) Build up a data colocation framework

HadoopBase-MIP



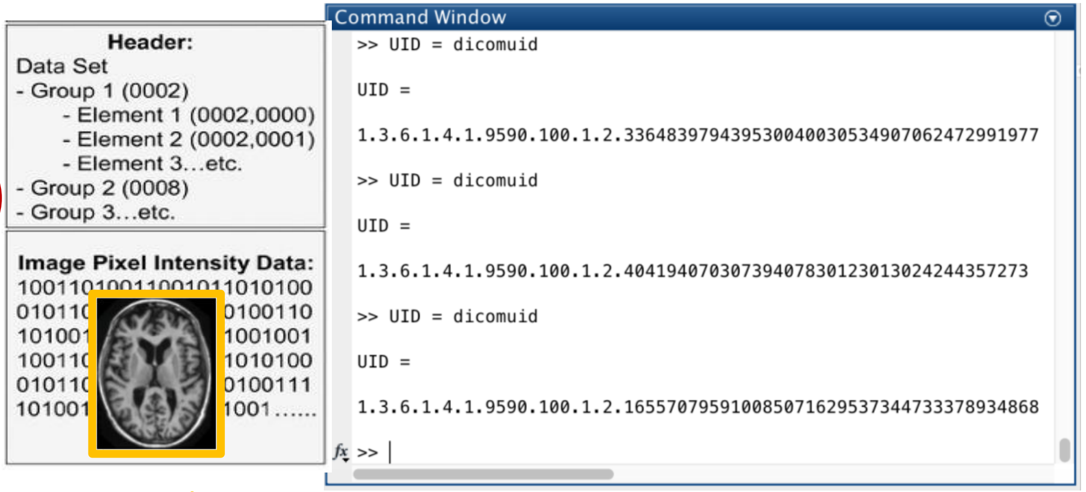
(2) Identify theoretical bound and promote system optimization



(3) Identify and reduce barriers of traditional medical image processing

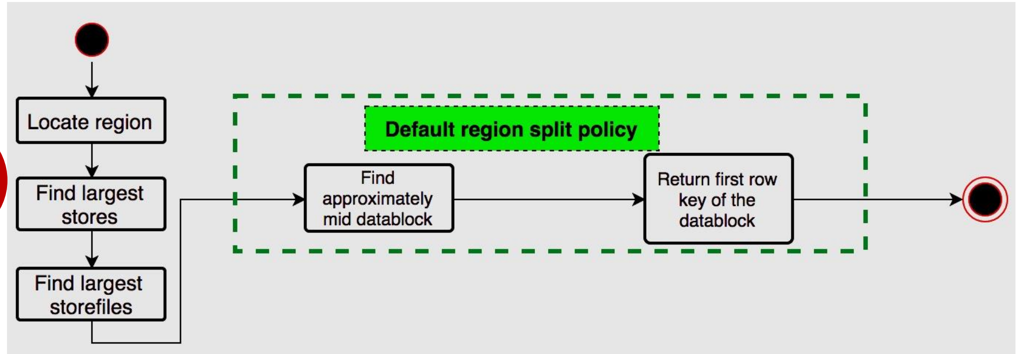
1. Global Unique Identifier (GUID) cannot reveal file internal structure.
2. HBase default region split policy - balance split
3. Default HBase MapReduce does not support group analysis

1

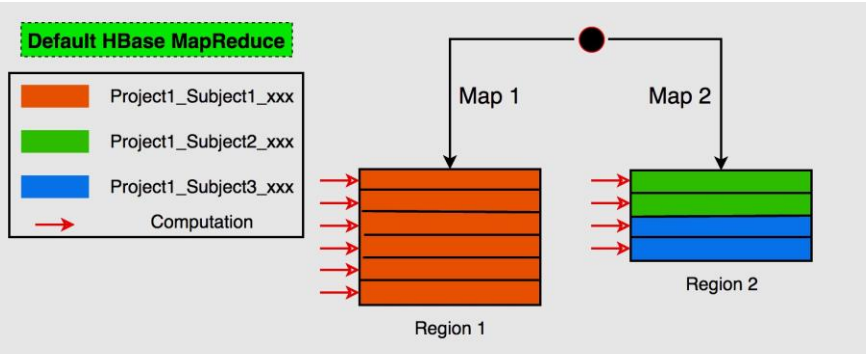


DICOM File

2



3



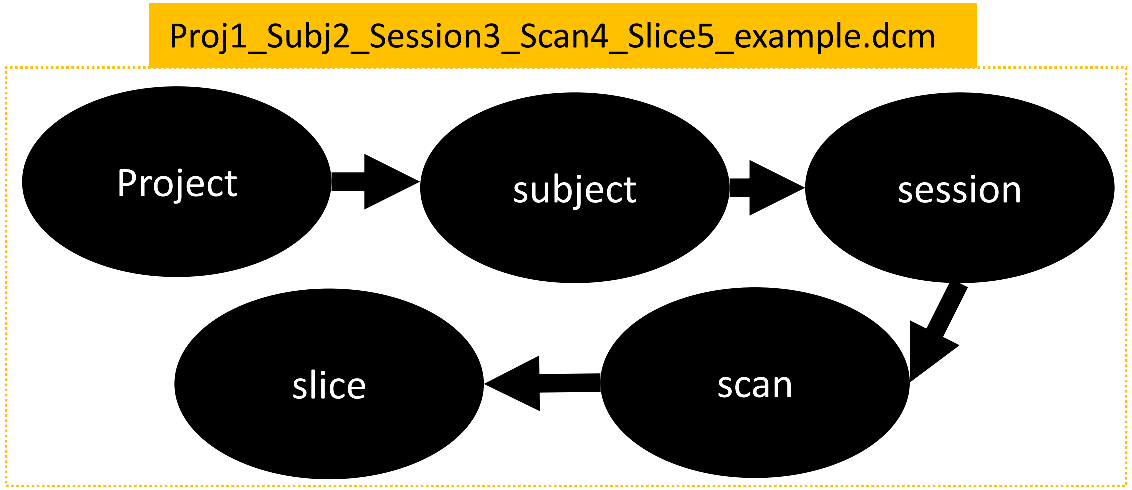
Challenge of making data colocation

- Can Hadoop & HBase be used in big data medical image processing for minimizing the data movement via network on commodity hardware?
  - Data storage
    - Structurize data placement order
    - Enforce relevant proximity
  - Data access & processing – Tuning Map~~Reduce~~
    - Utilize the benefit of data storage
    - Reduce the effort of algorithm/software re-design

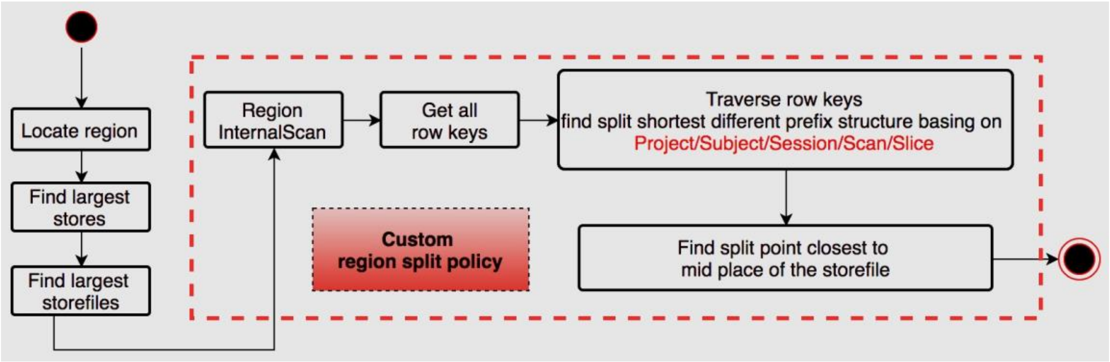
Our solution for data colocation

1. Custom hierarchical ID.
2. Custom region split policy – maximize data colocation
3. Custom HBase MapReduce
   - Group analysis
   - Avoid algorithm re-design

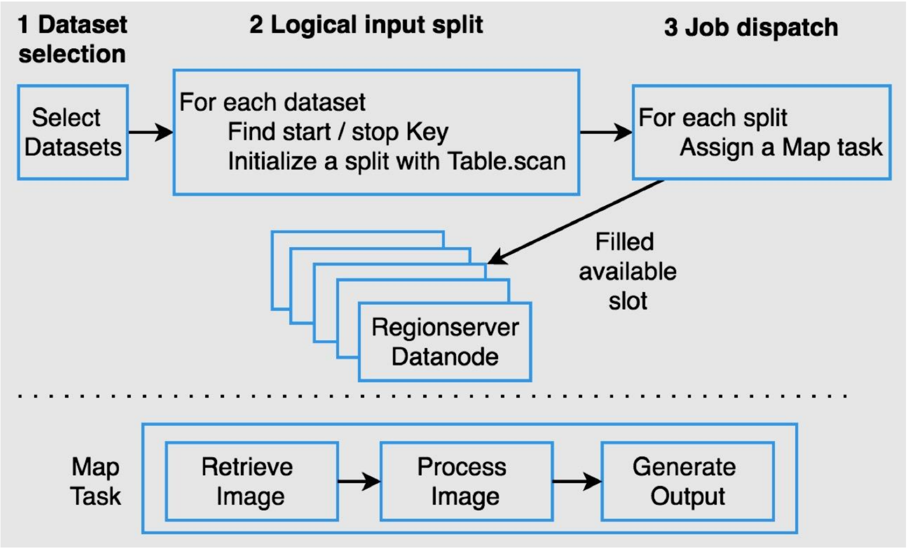
1



2



3



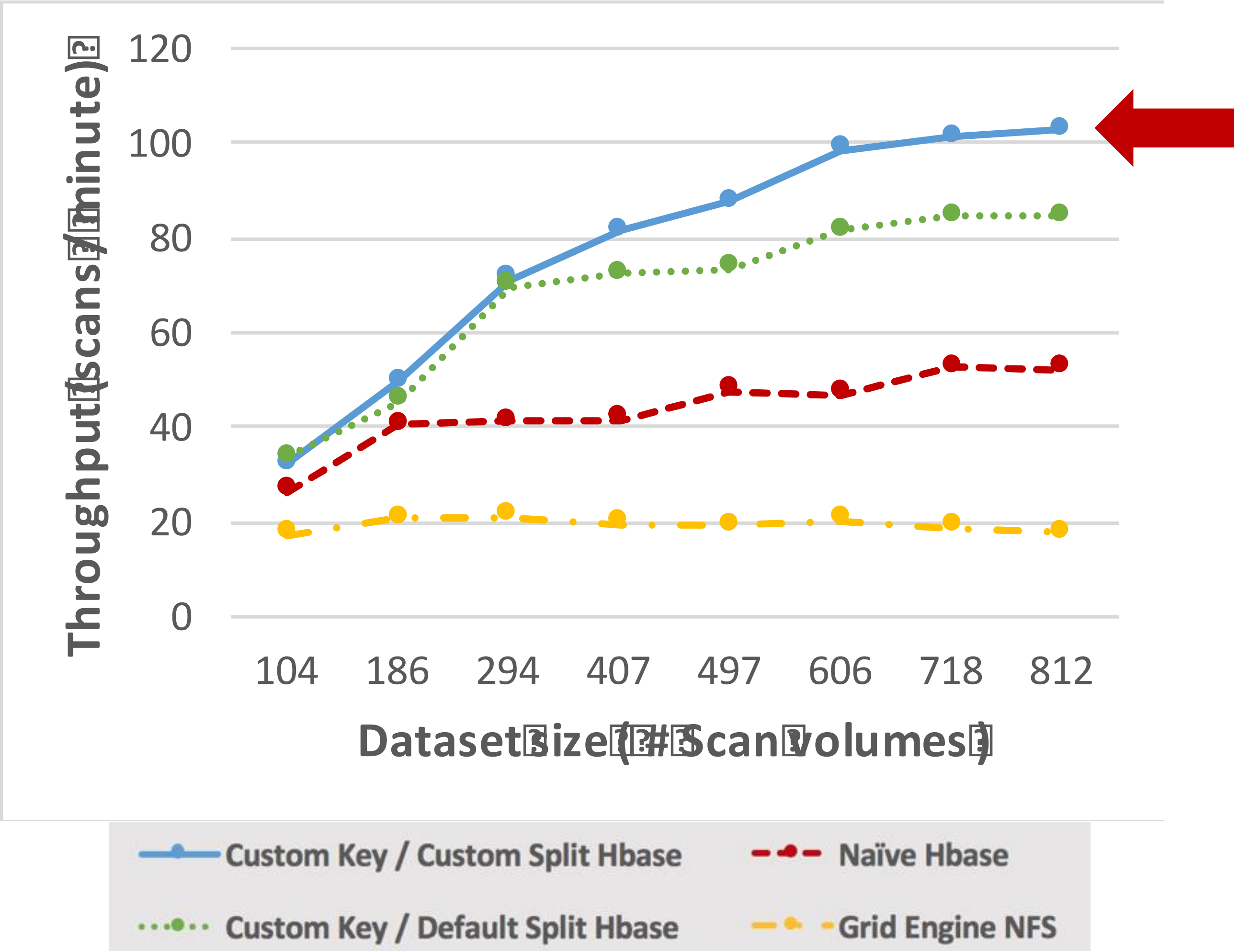
Validation for our data colocation

- Data access latency
  - Average data retrieval time per dataset

	Grid Engine NAS	Naïve HBase	Custom Key / Standard Split HBase	Custom Key / Custom Split HBase
latency (s)	4.76	19.02	3.29	2.56

Validation for our data colocation

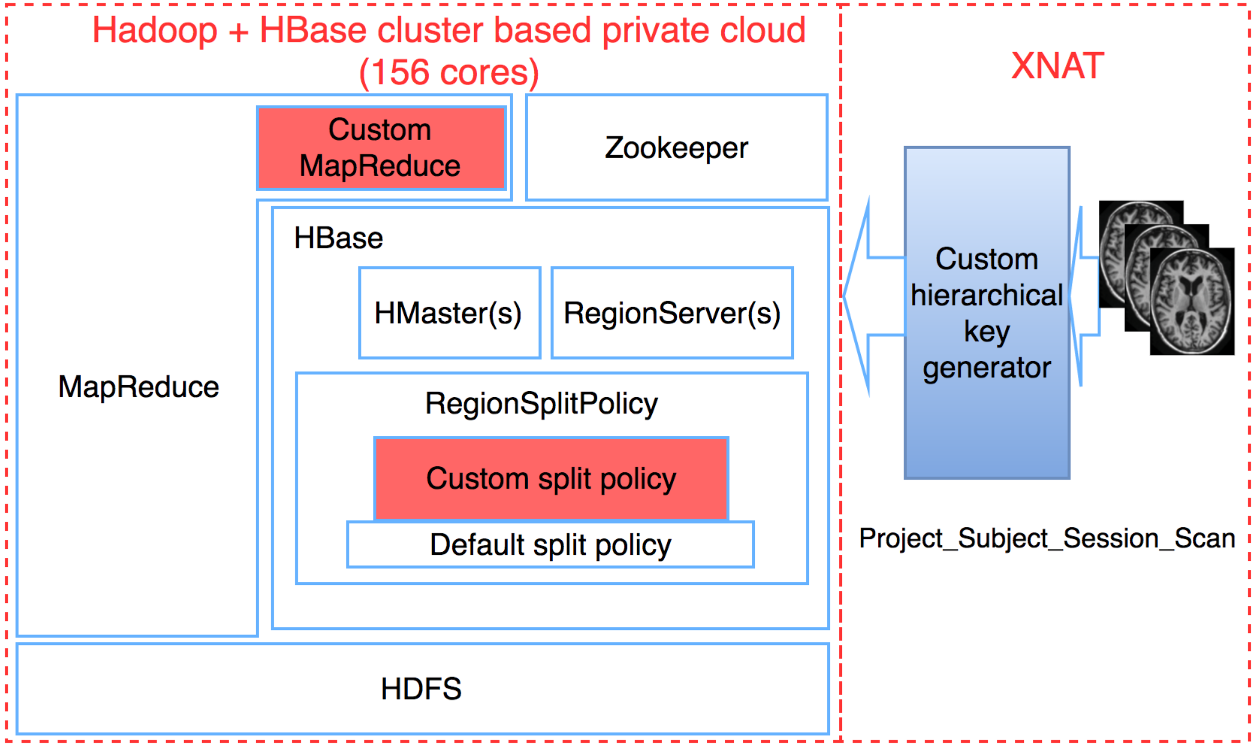
- Data processing throughput (dataset / minute)



Lesson's learned from our data colocation prototype

- A big data data colocation framework
- Improve data processing throughput
- Reduce data processing data access latency

Hadoop & HBase-based toolkit for medical image processing (HadoopBase-MIP)



Bao, et al. "Cloud Engineering Principles and Technology Enablers for Medical Image Processing-as-a-Service." IC2E 2017. (acceptance rate 22%)

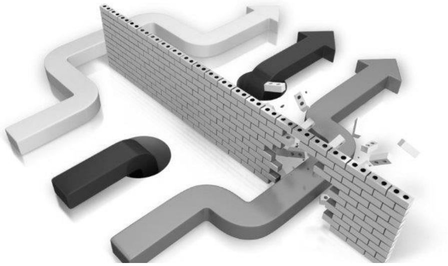
Outline

- Establishing Theoretical Bounds
- HadoopBase-MIP optimization

(1) Build up a data colocation framework

HadoopBase-MIP

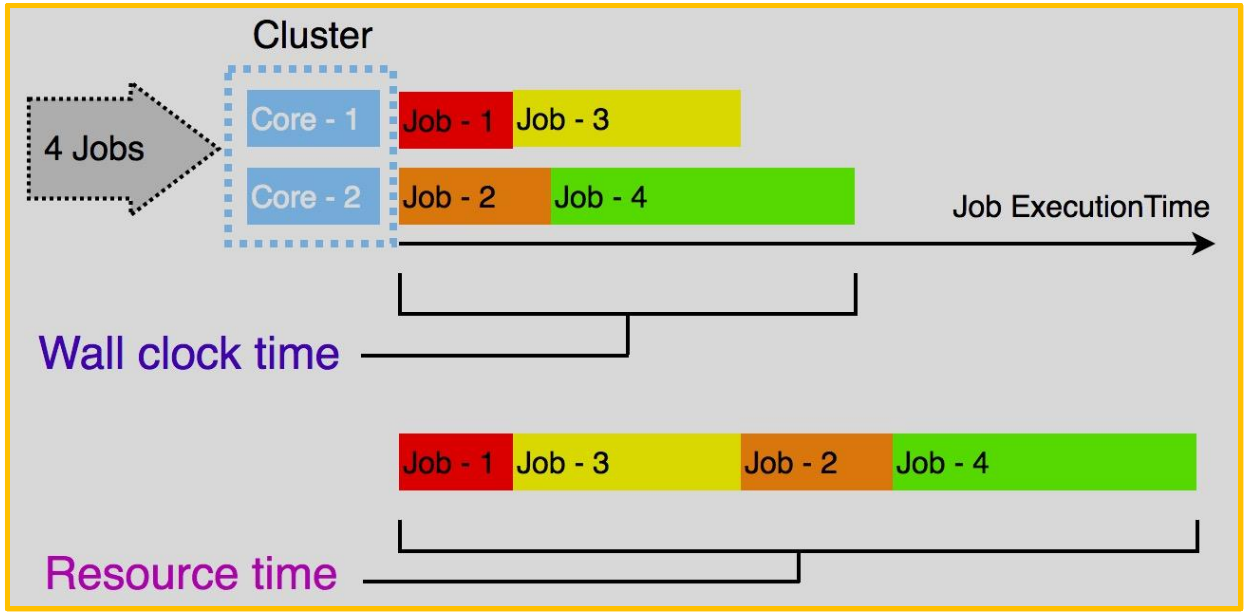
(2) Identify theoretical bound and promote system optimization



(3) Identify and reduce barriers of traditional medical image processing

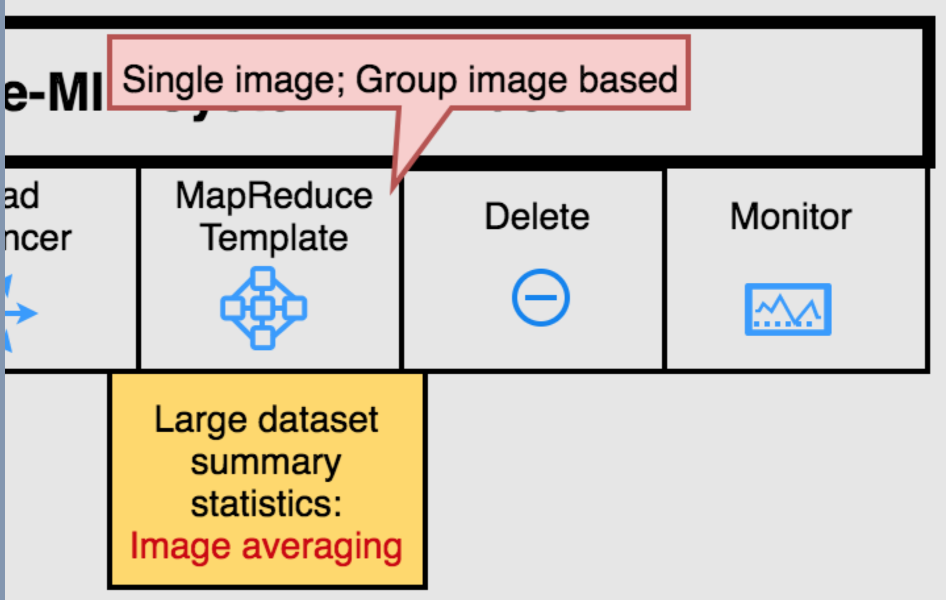
Performance metrics user may care

- **Wall clock time**
  - The total time as experienced by the user.
- **Resource time**
  - Elapsed time on each core when a process starts across all cores in a cluster.



Limitation of HadoopBase-MIP prototype framework

- Different types of application (fast - long)
- Different number of machines
- Large dataset analysis use case - average 5153 images (77.4GB)
  - Average all? – limit by machine memory
  - Split 5153 into several groups? - limit by cluster machine
  - Traditional cluster? – limit by network;
    > 16 CPU hours (1 Gbps)
- HadoopBase-MIP? < 2 CPU hours
  split tasks - Map
  combine tasks - Reduce
  $chunk\ size = \lceil 5153/\#groups \rceil$

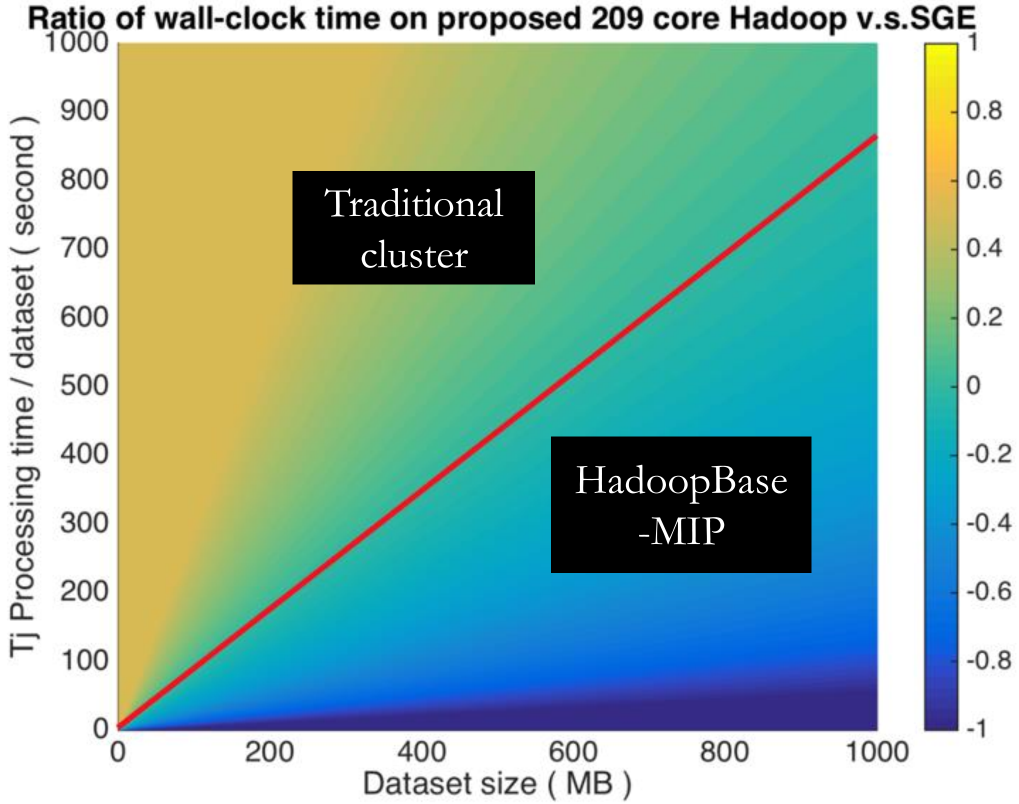


Challenge to understand & optimize HadoopBase-MIP

- What is the theoretical way to know when HadoopBase-MIP helps or hurts compared with traditional cluster?
- What is the theoretical way to know the split chunk size for large dataset image averaging?
- Do theoretical models translate practical things?

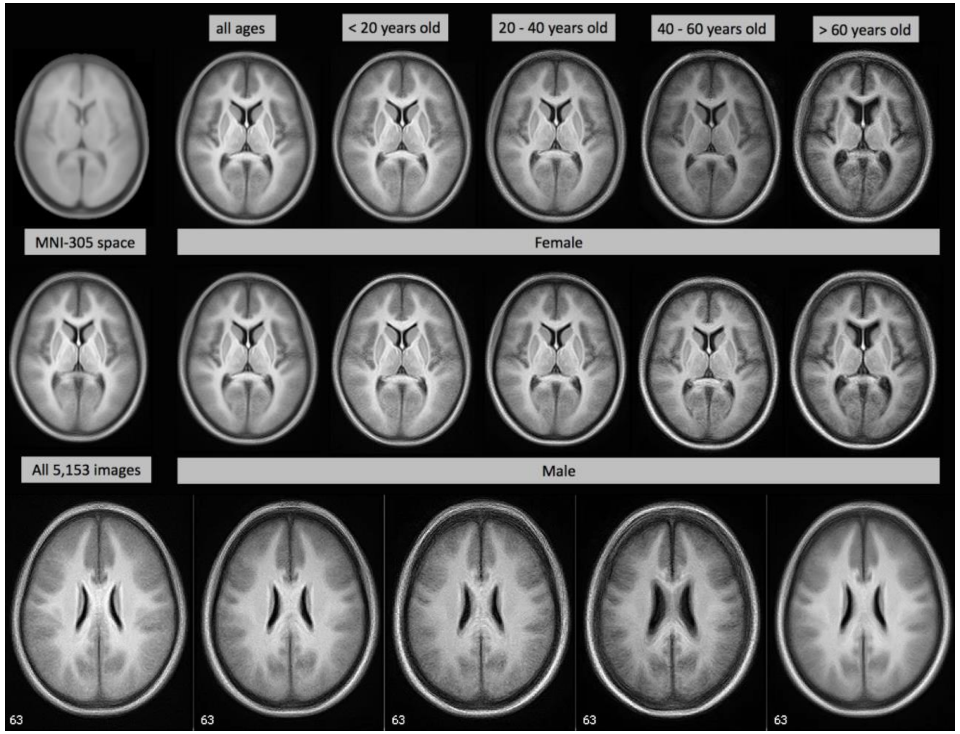
Lessons learned from theoretical bounds & system optimization

Theoretical bounds



Bao, et al. "Theoretical and empirical comparison of big data image processing with apache Hadoop and sun grid engine." SPIE Medical Imaging 2017.

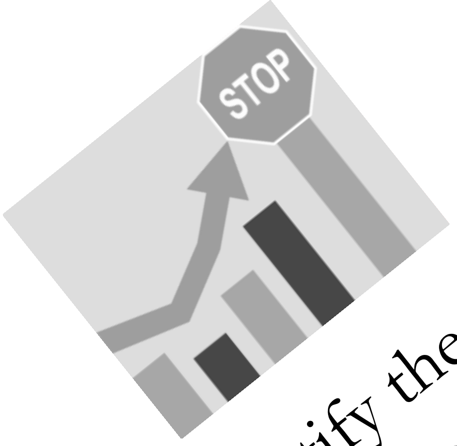
Real time large datasets image averaging



Bao, et al.. "A Data Colocation Grid Framework for Big Data Medical Image Processing-Backend Design." SPIE Medical Imaging 2018.

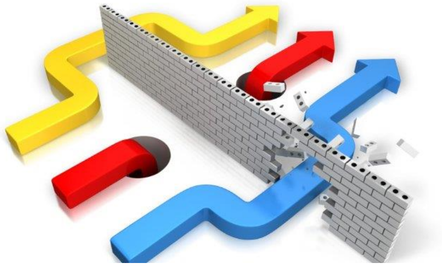
Outline

(1) Build up a data colocation framework



(2) Identify theoretical bound and promote system optimization

HadoopBase-MIP



(3) Identify and reduce barriers of traditional medical image processing

Optimizing traditional multi-level Medical image analysis's quality assurance

Why we care about quality assurance for multi-level analysis

- Multi-level analysis
  - 1st level - Preprocessing (slow)

As a researcher in brain image processing, Professor Bennett Landman goes through a lot of data – up to 96 hours to process one head! When he first arrived at Vanderbilt, he would push ACCRE to its limits and take down ACCRE about every six months. Nowadays, he manages his own experimental cluster with 480 CPUs and 2TB of RAM while using ACCRE for less risky computing tasks. For Professor Landman, ACCRE provides many advantages: it's cheap, reliable, stable, and easy to manage, it's backed up regularly, it makes it easy to collaborate with others, and it raises less legal concerns than a cloud-based server. Professor Landman provides support to ACCRE's operations as the co-chair of its Faculty Advisory Board.

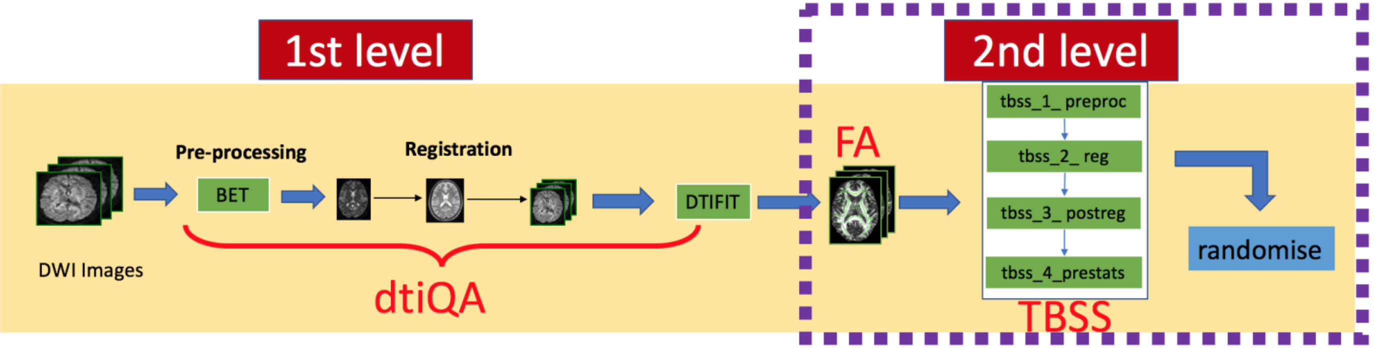
datasets

- Cost and resource conservation in cloud?

Challenges for faster multi-level analysis quality assurance

ACCRE
Advanced Computing Center for Research & Education

- How can we detect outliers in 1^{st} level **as early as possible**?
- How can we draw expected conclusion **as early as possible**?
- How to automate the quality assurance process?



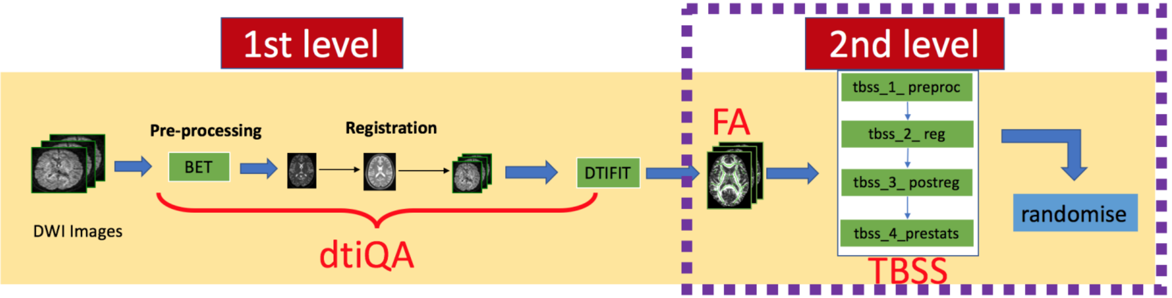
$$\left\lceil \frac{7hours \cdot total\ imges}{cluster\ cores} \right\rceil$$

$$0.02hours \cdot total\ imges$$

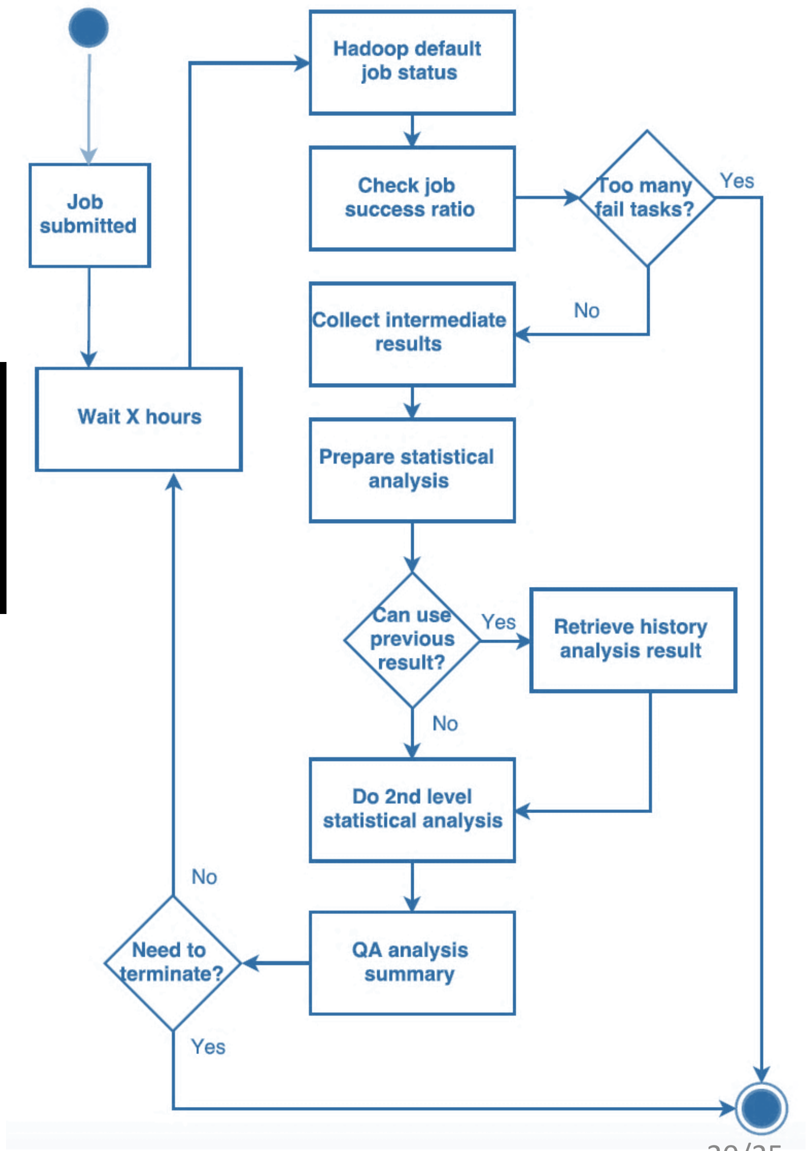
Single core

Our solution for efficient multi-level analysis quality assurance

- Collect enough intermediate result from 1st level.
  - Run 2nd level group analysis incrementally.
- Use 2nd level incrementally analysis result to identify error / weird outcome.
- Draw expected conclusion early.

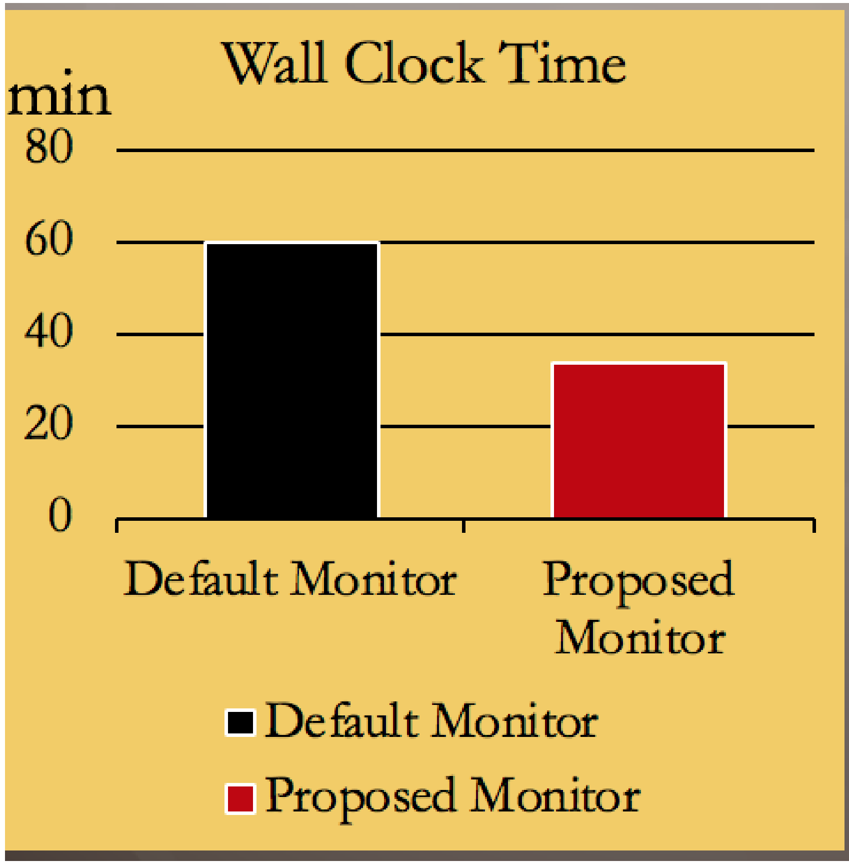


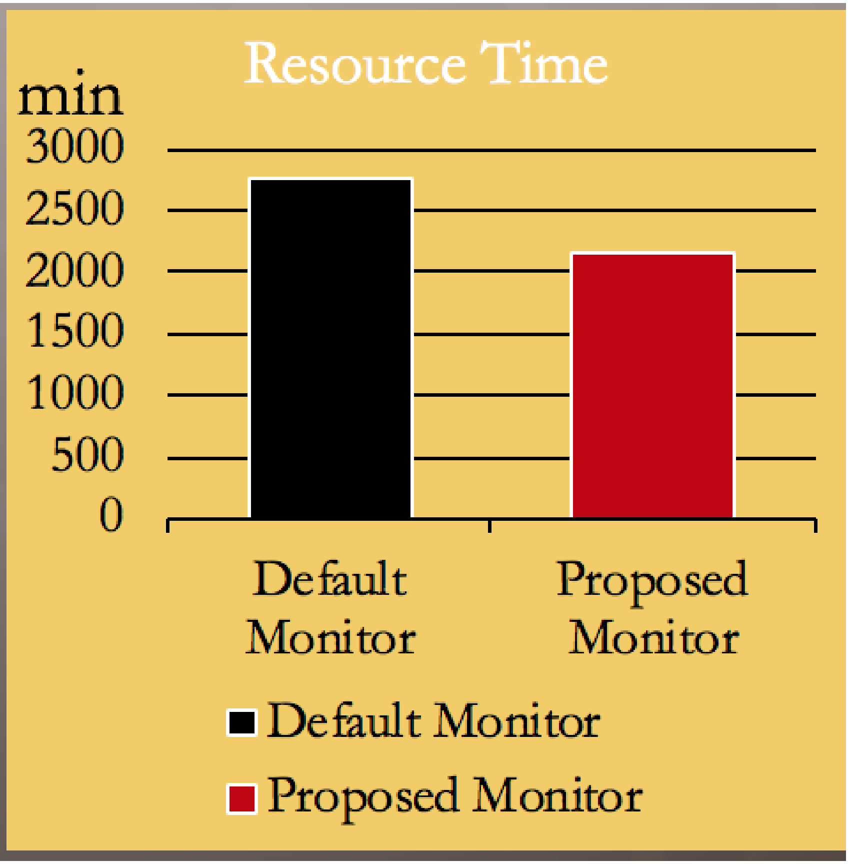
A semi-automatic real-time monitor and checkpoint framework



Validation for our quality assurance monitor

- Cost conservation



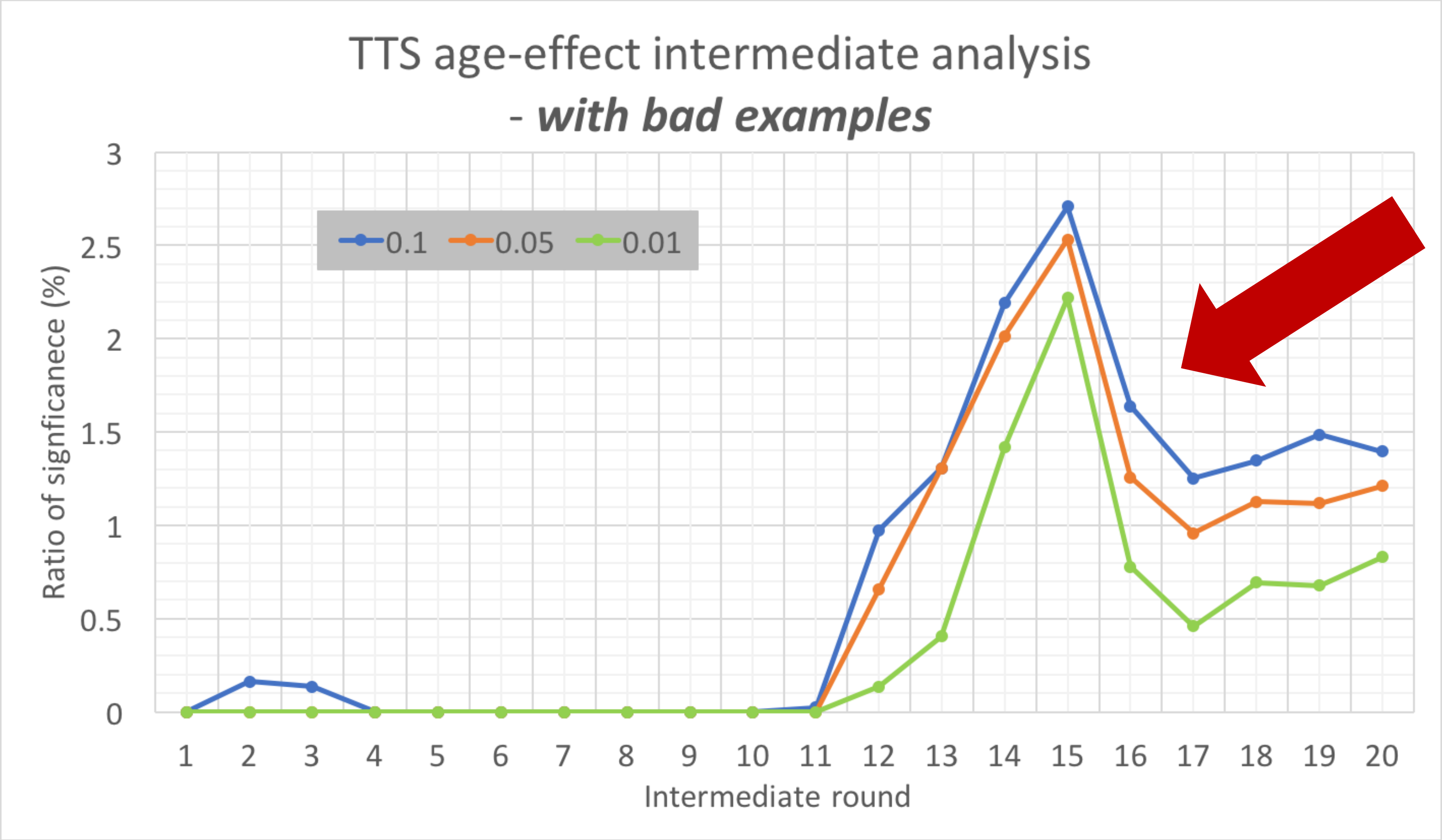


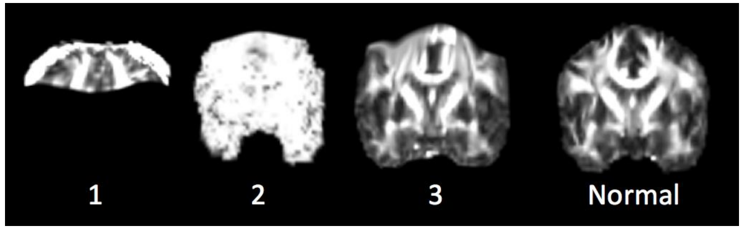
76.7%

29.2%

Validation for our quality assurance monitor

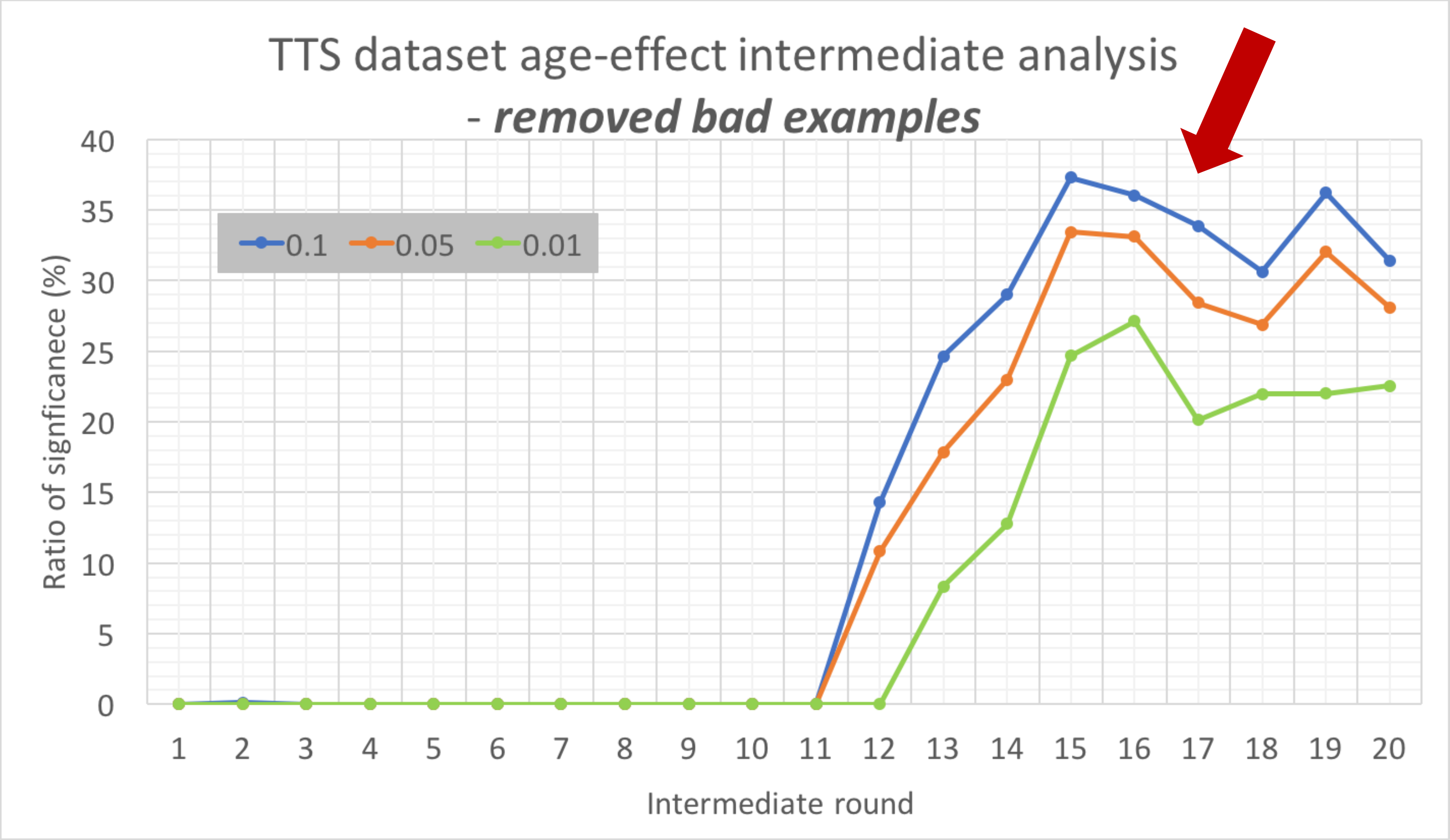
- Early error detection





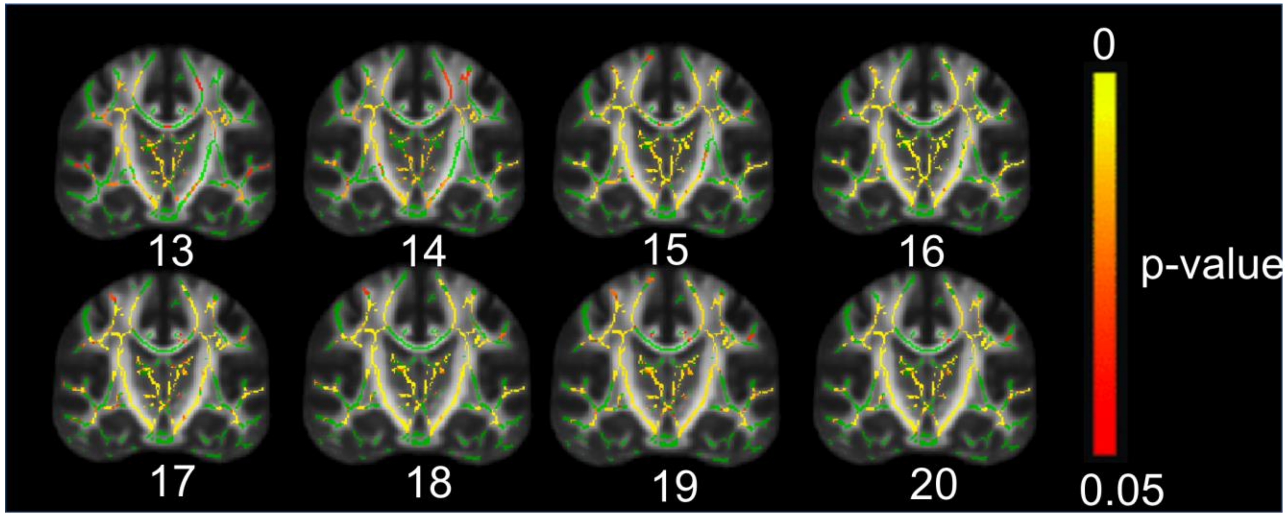
Validation for our quality assurance monitor

- Draw conclusion early



Lessons learned from our quality assurance monitor

- Reproducible result with Kodiweera et al. *NeuroImage* 2016
- Our innovation help us re-think current Multi-level analysis
- Can be easily extended to current pipeline using traditional cluster



Bao et al., " Technology Enablers for Cloud-based Multi-level Analysis Applications in Medical Image Processing " IEEE BigData 2018 (accepted) (acceptance rate 18.9%)

Thank you very much. Questions?