Acceleration of Data-intensive Workflow Applications by Using File Access History

Miki Horiuchi and Kenjiro Taura, the University of Tokyo

WORKS 2012, Salt Lake City

Agenda

- Background & motivation
  - Data-intensive computing and data-aware job scheduler
  - Current data-aware job scheduler
  - Related work
- Design and implementation of proposal
  - GXP Make and Mogami
  - Prediction of input/output files by using file access history
- Evaluation
  - Case Frame Construction
  - Montage
- Conclusion & future work

Background

- More and more applications have been becoming data-intensive
  - Workflow applications
  - Scientific data analysis, text processing and machine learning
- Important to manage data transfer efficiently in distributed computing
  - Distributed file systems
  - Data-aware job scheduler

Current Data-aware Job Scheduling

- Require users to explicitly describe input/output files
  - Burden for workflow developers
  - E.g.) Montage (mConcatFit)

Input files:

- fit.792.793.txt
- fit.792.875.txt
- fit.792.795.txt
- fit.792.794.txt
- fit.792.796.txt
- fit.792.870.txt
- fit.350.354.txt

:

:

more than **2,700** files in one job

Not appear in the command line

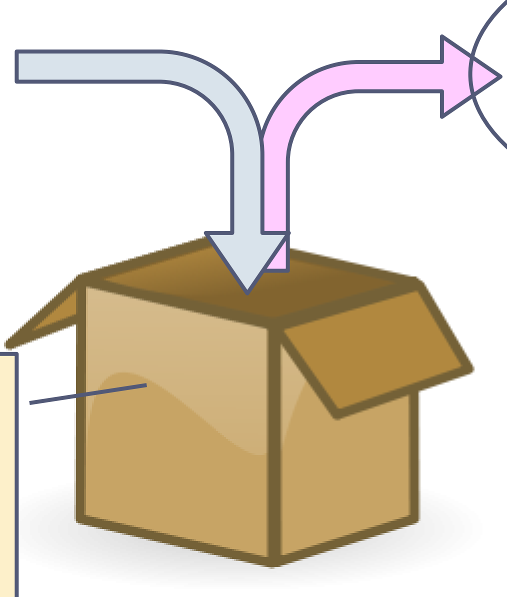
Workflow developers must know and write the precise sets of access files for such jobs

- Cannot utilize the flexibility of distributed file systems

Our proposal

- Method to deduce input/output files without any user-supplied annotations
  - Automatic data-aware job scheduling

Target command (just before dispatched):
'mProject X Y Z'



Access file prediction:

X.txt	4096kB
Y	1024kB

Applicable for data-aware job scheduling

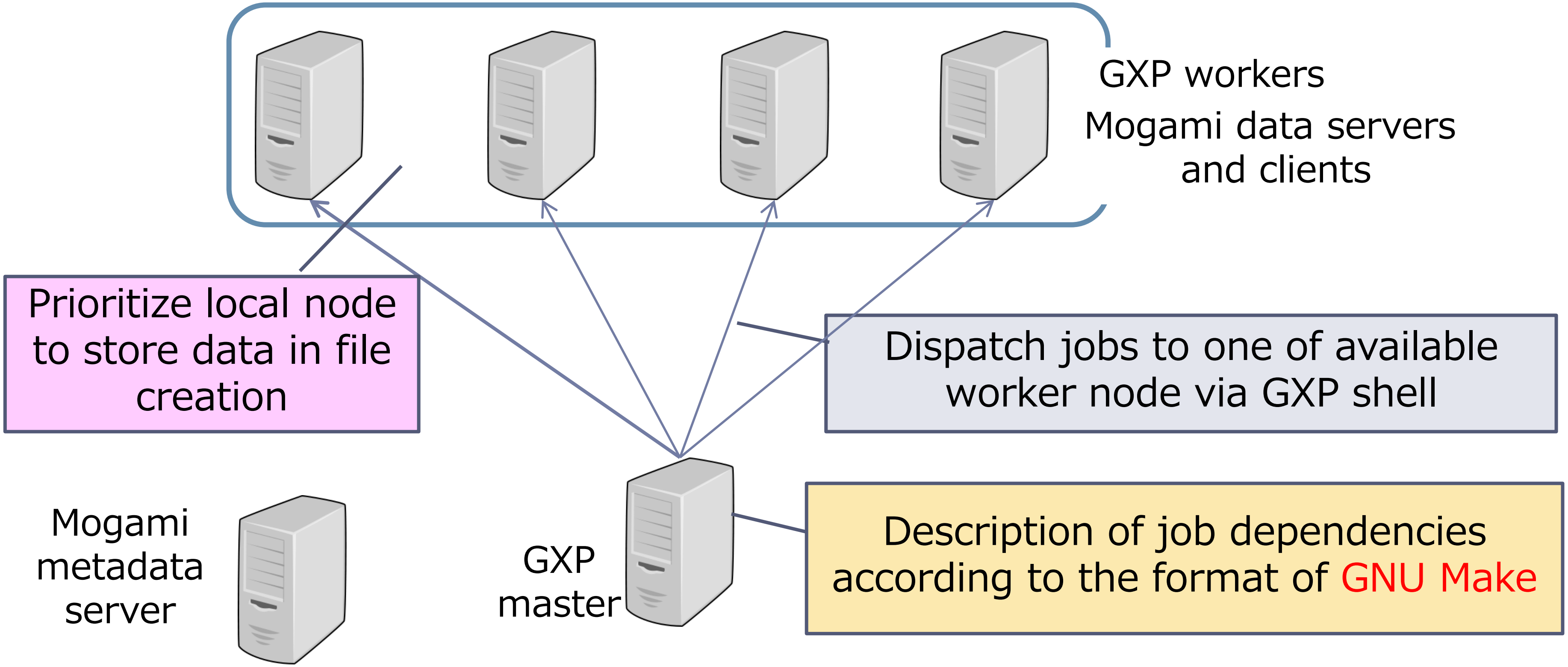
file access history gathered in a similar execution (i.e. with different data, parameters etc.)

Related Work

- Improvement of data transfer in workflow applications
  - Stork data placement scheduler [Kosar et al. 2009]
  - Minimizing data transfer between nodes by applying MCGP [Tanaka et al. 2012]
- Workflow-aware storage system
  - Optimization of MosaStore for common data access patterns [Vairavanathan et al. 2012]
- Frameworks which encourages data locality
  - Hadoop
  - Sector (a distributed file system) and Sphere (an associated programming framework) [Gu et al. 2011]

System Overview for Proposal

- GXP Make – a workflow engine [Taura et al. 2010]
- Mogami – a distributed file system [Horiuchi et al. 2011]



GXP workers
Mogami data servers and clients

Prioritize local node to store data in file creation

Dispatch jobs to one of available worker node via GXP shell

Mogami metadata server

GXP master

Description of job dependencies according to the format of GNU Make

Input/output Files of Each Job in Makefile

- In using a distributed file system, each job may
  - Read files other than those listed in its prerequisites
  - Write to files other than the target

E.g.)

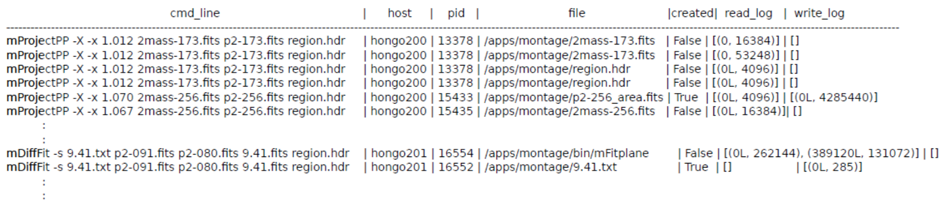
```
X:  A
        cmd A –f B
```

In this example, the job of ‘cmd A –f B’ may read files other than ‘A’ and write to files other than ‘X’

So predicting access files is not easy.

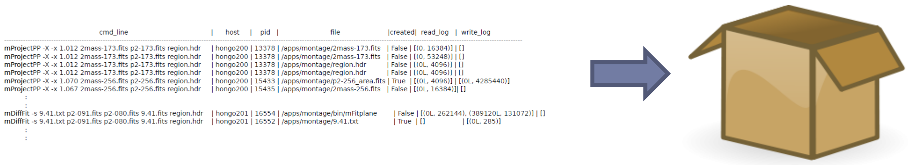
Steps of Our Proposal

Step1: Gather the file access history in a profiling run

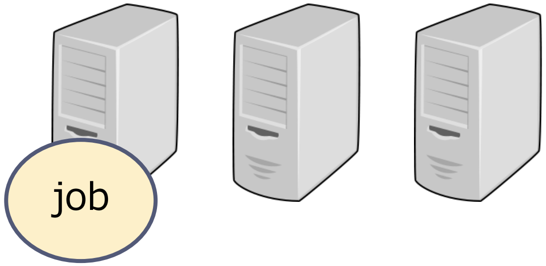


```
                        cmd_line                          |     host    |   pid  |                    file                |created|  read_log    |  write_log
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/2mass-173.fits    | False | [(0, 16384)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/2mass-173.fits    | False | [(0, 53248)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/region.hdr        | False | [(0L, 4096)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/region.hdr        | False | [(0L, 4096)] | []
mProjectPP -X -x 1.070 2mass-256.fits p2-256.fits region.hdr     | hongo200 | 15433 | /apps/montage/p2-256_area.fits | True  | [(0L, 4096)] | [(0L, 4285440)]
mProjectPP -X -x 1.067 2mass-256.fits p2-256.fits region.hdr     | hongo200 | 15435 | /apps/montage/2mass-256.fits    | False | [(0L, 16384)]| []
          :
          :
mDiffFit -s 9.41.txt p2-091.fits p2-080.fits 9.41.fits region.hdr   | hongo201 | 16554 | /apps/montage/bin/mFitplane       | False | [(0L, 262144), (389120L, 131072)] | []
mDiffFit -s 9.41.txt p2-091.fits p2-080.fits 9.41.fits region.hdr   | hongo201 | 16552 | /apps/montage/9.41.txt            | True  | []                 | [(0L, 285)]
          :
          :
```

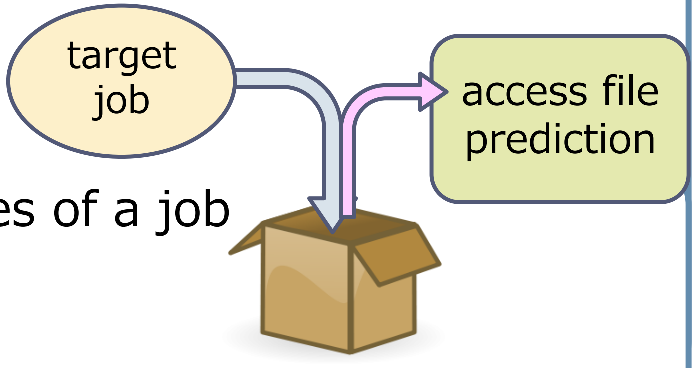
Step2: Analyze the file access history



```
                        cmd_line                          |     host    |   pid  |                    file                |created|  read_log    |  write_log
-------------------------------------------------------------------------------------------------------------------------------------------------------------------
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/2mass-173.fits    | False | [(0, 16384)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/2mass-173.fits    | False | [(0, 53248)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/region.hdr        | False | [(0L, 4096)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr     | hongo200 | 13378 | /apps/montage/region.hdr        | False | [(0L, 4096)] | []
mProjectPP -X -x 1.070 2mass-256.fits p2-256.fits region.hdr     | hongo200 | 15433 | /apps/montage/p2-256_area.fits | True  | [(0L, 4096)] | [(0L, 4285440)]
mProjectPP -X -x 1.067 2mass-256.fits p2-256.fits region.hdr     | hongo200 | 15435 | /apps/montage/2mass-256.fits    | False | [(0L, 16384)]| []
          :
          :
mDiffFit -s 9.41.txt p2-091.fits p2-080.fits 9.41.fits region.hdr   | hongo201 | 16554 | /apps/montage/bin/mFitplane       | False | [(0L, 262144), (389120L, 131072)] | []
mDiffFit -s 9.41.txt p2-091.fits p2-080.fits 9.41.fits region.hdr   | hongo201 | 16552 | /apps/montage/9.41.txt            | True  | []                 | [(0L, 285)]
          :
          :
```



Step4: Schedule jobs utilizing the access file prediction

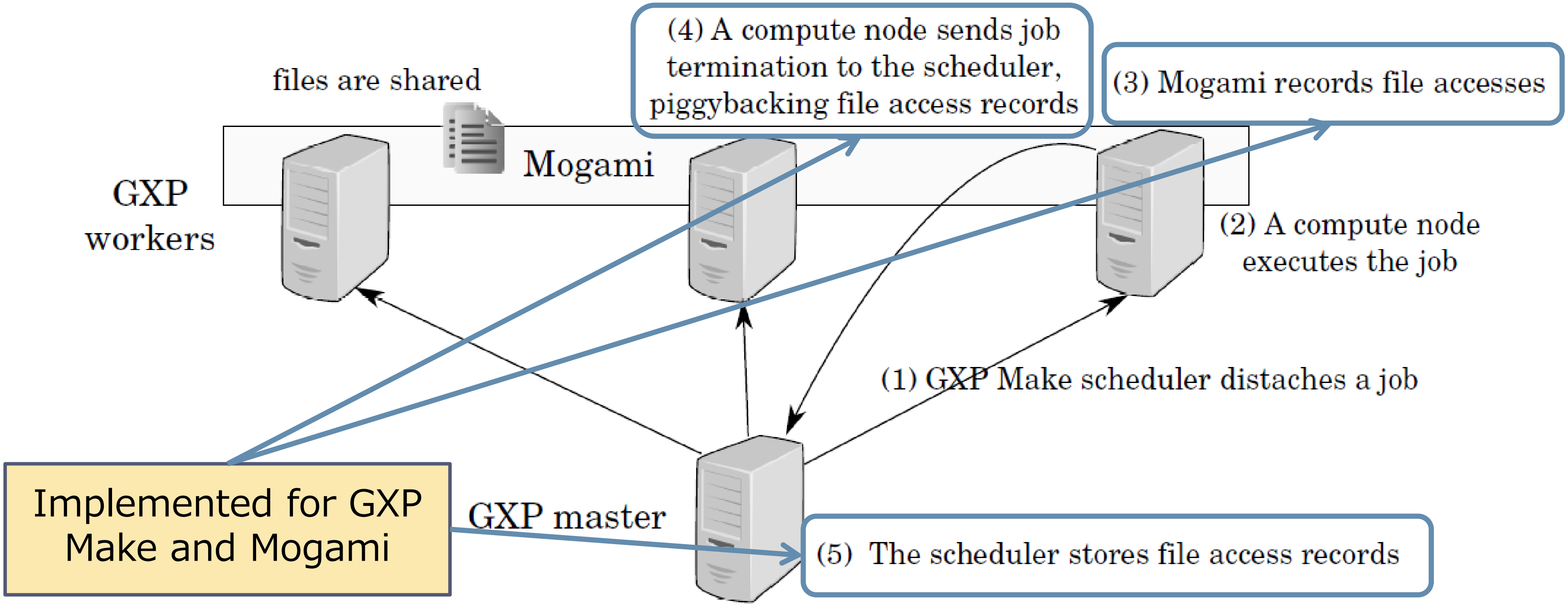


Step3: Predict input/output files of a job

1) Filtering phase
2) Weighing phase
3) Prediction phase

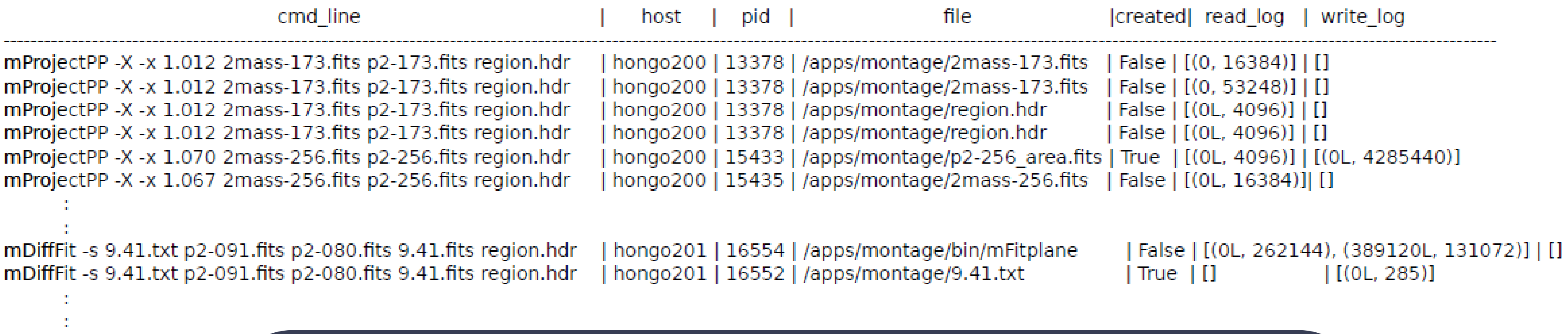
Profiling Run

- Run workflow once as a profiling run
  - work with different data or different parameters as long as same modules are used
- Gather file access history in the profiling run


(4) A compute node sends job termination to the scheduler, piggybacking file access records
files are shared
(3) Mogami records file accesses
Mogami
GXP workers
(2) A compute node executes the job
(1) GXP Make scheduler distaches a job
Implemented for GXP Make and Mogami
GXP master
(5) The scheduler stores file access records

Format of File Access History

Actual file access history of Montage

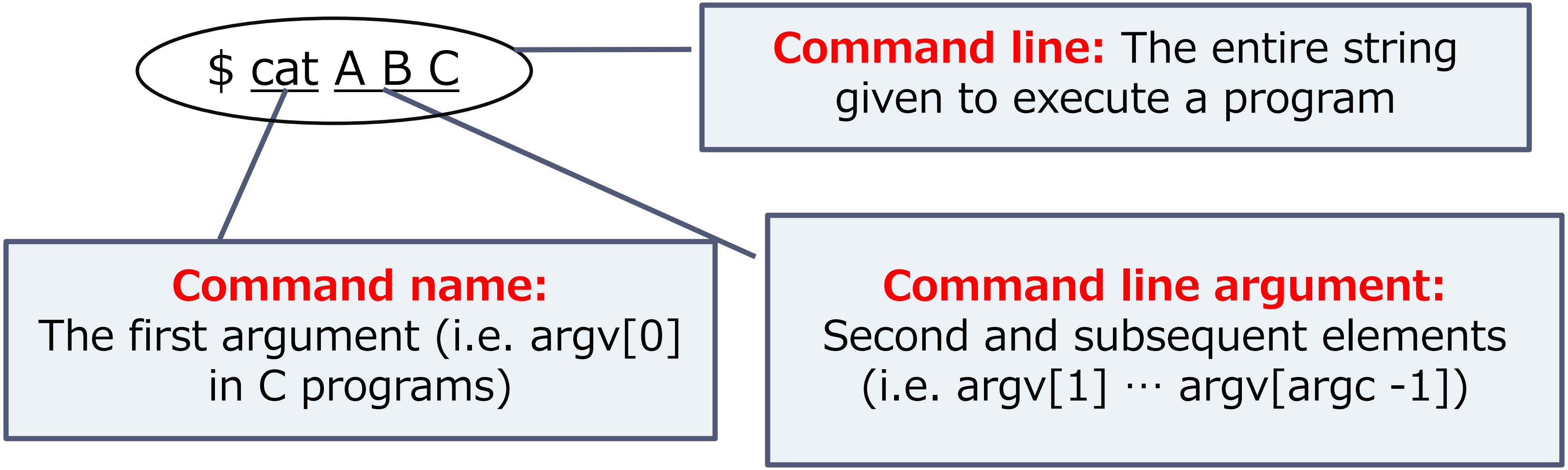


```
                  cmd_line                          |   host    |  pid  |                file                |created|  read_log   |  write_log
------------------------------------------------------------------------------------------------------------------------------------------------------
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr    | hongo200 | 13378 | /apps/montage/2mass-173.fits   | False | [(0, 16384)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr    | hongo200 | 13378 | /apps/montage/2mass-173.fits   | False | [(0, 53248)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr    | hongo200 | 13378 | /apps/montage/region.hdr       | False | [(0L, 4096)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr    | hongo200 | 13378 | /apps/montage/region.hdr       | False | [(0L, 4096)] | []
mProjectPP -X -x 1.070 2mass-256.fits p2-256.fits region.hdr    | hongo200 | 15433 | /apps/montage/p2-256_area.fits | True  | [(0L, 4096)] | [(0L, 4285440)]
mProjectPP -X -x 1.067 2mass-256.fits p2-256.fits region.hdr    | hongo200 | 15435 | /apps/montage/2mass-256.fits   | False | [(0L, 16384)]| []
      :

      :
mDiffFit -s 9.41.txt p2-091.fits p2-080.fits 9.41.fits region.hdr   | hongo201 | 16554 | /apps/montage/bin/mFitplane      | False | [(0L, 262144), (389120L, 131072)] | []
mDiffFit -s 9.41.txt p2-091.fits p2-080.fits 9.41.fits region.hdr   | hongo201 | 16552 | /apps/montage/9.41.txt           | True  | []              | [(0L, 285)]
      :
      :
```

cmd_line: command line to execute the job
host: hostname where the command was executed
pid: pid of process that opened file
file: file path
created: if the file was created or not
read_log: (offset, size) of each read
write_log: (offset, size) of each write

Note: Definitions of Words

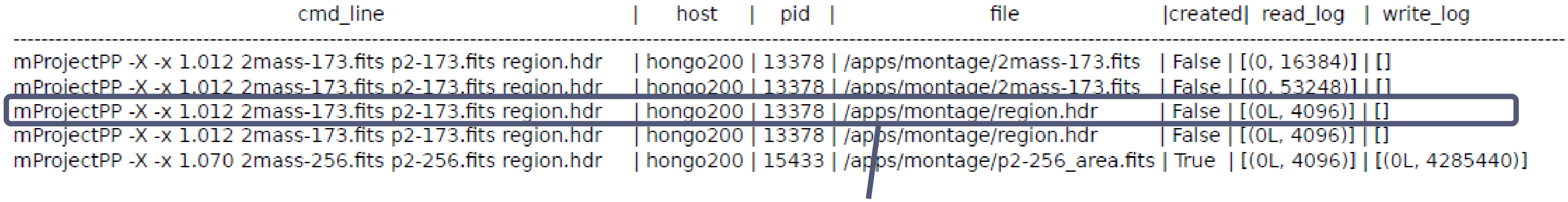


$ cat A B C

Command line: The entire string given to execute a program

Command name: The first argument (i.e. argv[0] in C programs)

Command line argument: Second and subsequent elements (i.e. argv[1] ··· argv[argc -1])



```
                       cmd_line                        |    host    |  pid  |                  file                |created|  read_log   |  write_log
------------------------------------------------------------------------------------------------------------------------------------------------------------
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr   | hongo200 | 13378 | /apps/montage/2mass-173.fits   | False | [(0, 16384)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr   | hongo200 | 13378 | /apps/montage/2mass-173.fits   | False | [(0, 53248)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr   | hongo200 | 13378 | /apps/montage/region.hdr       | False | [(0L, 4096)] | []
mProjectPP -X -x 1.012 2mass-173.fits p2-173.fits region.hdr   | hongo200 | 13378 | /apps/montage/region.hdr       | False | [(0L, 4096)] | []
mProjectPP -X -x 1.070 2mass-256.fits p2-256.fits region.hdr   | hongo200 | 15433 | /apps/montage/p2-256_area.fits | True  | [(0L, 4096)] | [(0L, 4285440)]
```

File access record:
A record of file access history (one job might have more than one file access records)

Analyzing File Access History

- Make one or two rules from each file access record
  - Focus on the relationship of file name and command line arguments
- Types of rule
  - ReplacePos(*n*, *a*, *b*): the job accessed its *n*-th argument, with its *a* replaced by *b*
  - InsertPos(*n*, *i*, *b*): the job accessed its *n*-th argument, with *b* inserted after its *i*-th character
  - ReplaceOpt(*f*, *a*, *b*): the job accessed the argument that follows *f*, with its *a* replaced by *b*
  - InsertOpt(*f*, *i*, *b*): the job accessed the argument that follows *f*, with *b* inserted after its *i*-th character

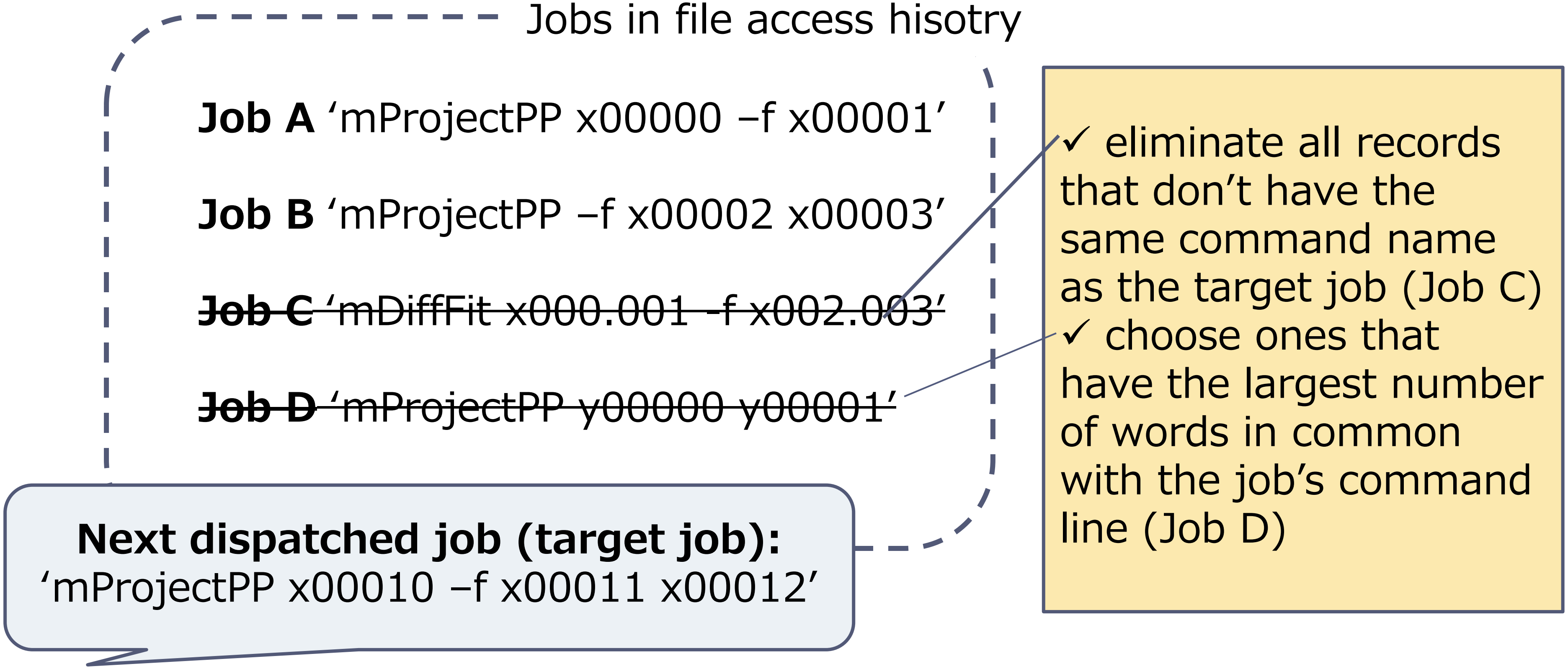
E.g.) ‘hoge A.txt -f B.txt’ accessed ‘B.dat’

⇩

ReplacePos(3, ‘.txt’, ‘.dat’) and ReplaceOpt(‘-f’, ‘.txt’, ‘.dat’)

How to Predict Input Files (1/3)

- 1) Filtering phase
  - selects records in the file access history that most closely match the command line of the job



Jobs in file access hisotry

Job A ‘mProjectPP x00000 –f x00001’

Job B ‘mProjectPP –f x00002 x00003’

~~**Job C** ‘mDiffFit x000.001 -f x002.003’~~

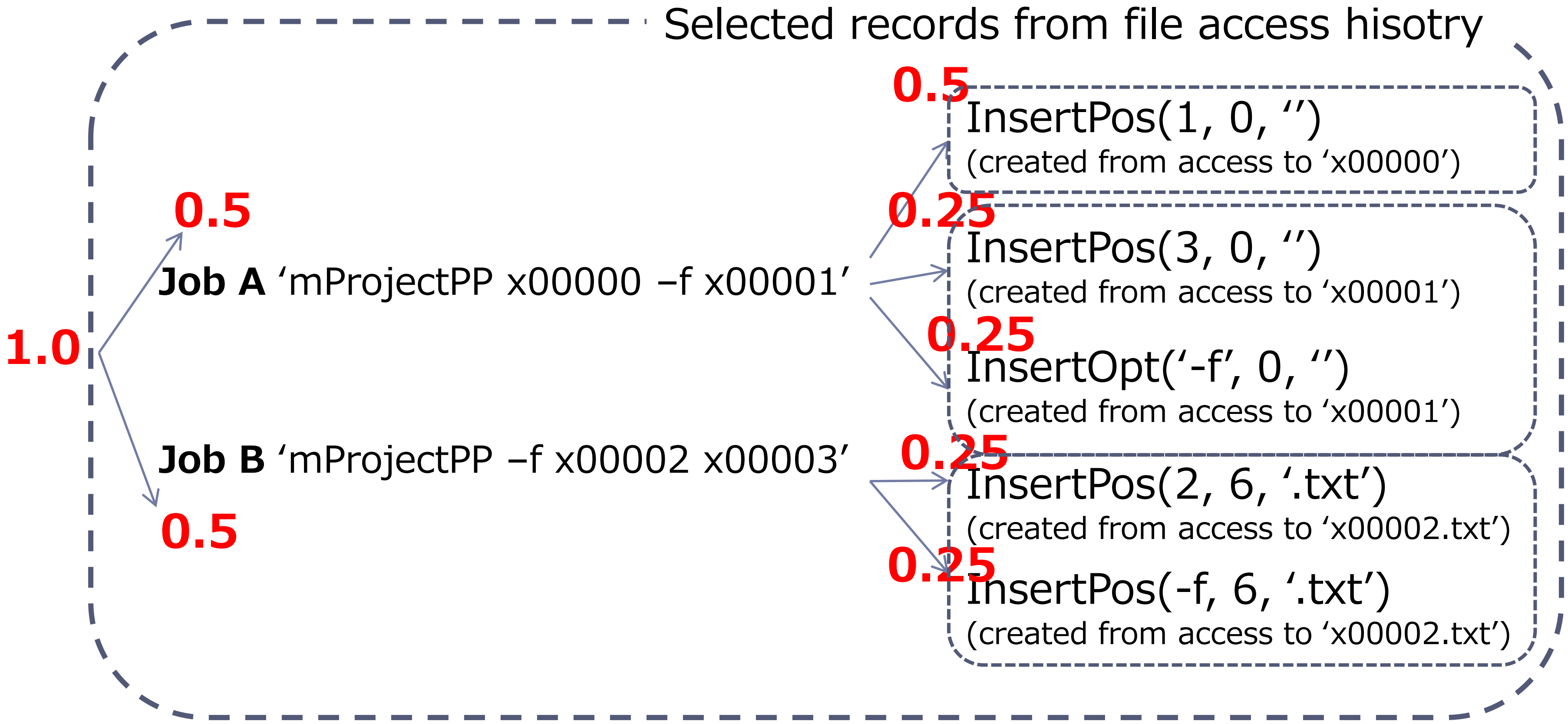
~~**Job D** ‘mProjectPP y00000 y00001’~~

Next dispatched job (target job):
‘mProjectPP x00010 –f x00011 x00012’

✓ eliminate all records that don’t have the same command name as the target job (Job C)
✓ choose ones that have the largest number of words in common with the job’s command line (Job D)

How to Predict Input Files (2/3)

- 2) Weighing phase
  - weighs rules associated with the selected records



Selected records from file access hisotry

1.0

0.5

Job A ‘mProjectPP x00000 –f x00001’

0.5

Job B ‘mProjectPP –f x00002 x00003’

0.5

InsertPos(1, 0, ‘’)
(created from access to ‘x00000’)

0.25

InsertPos(3, 0, ‘’)
(created from access to ‘x00001’)

0.25

InsertOpt(‘-f’, 0, ‘’)
(created from access to ‘x00001’)

0.25

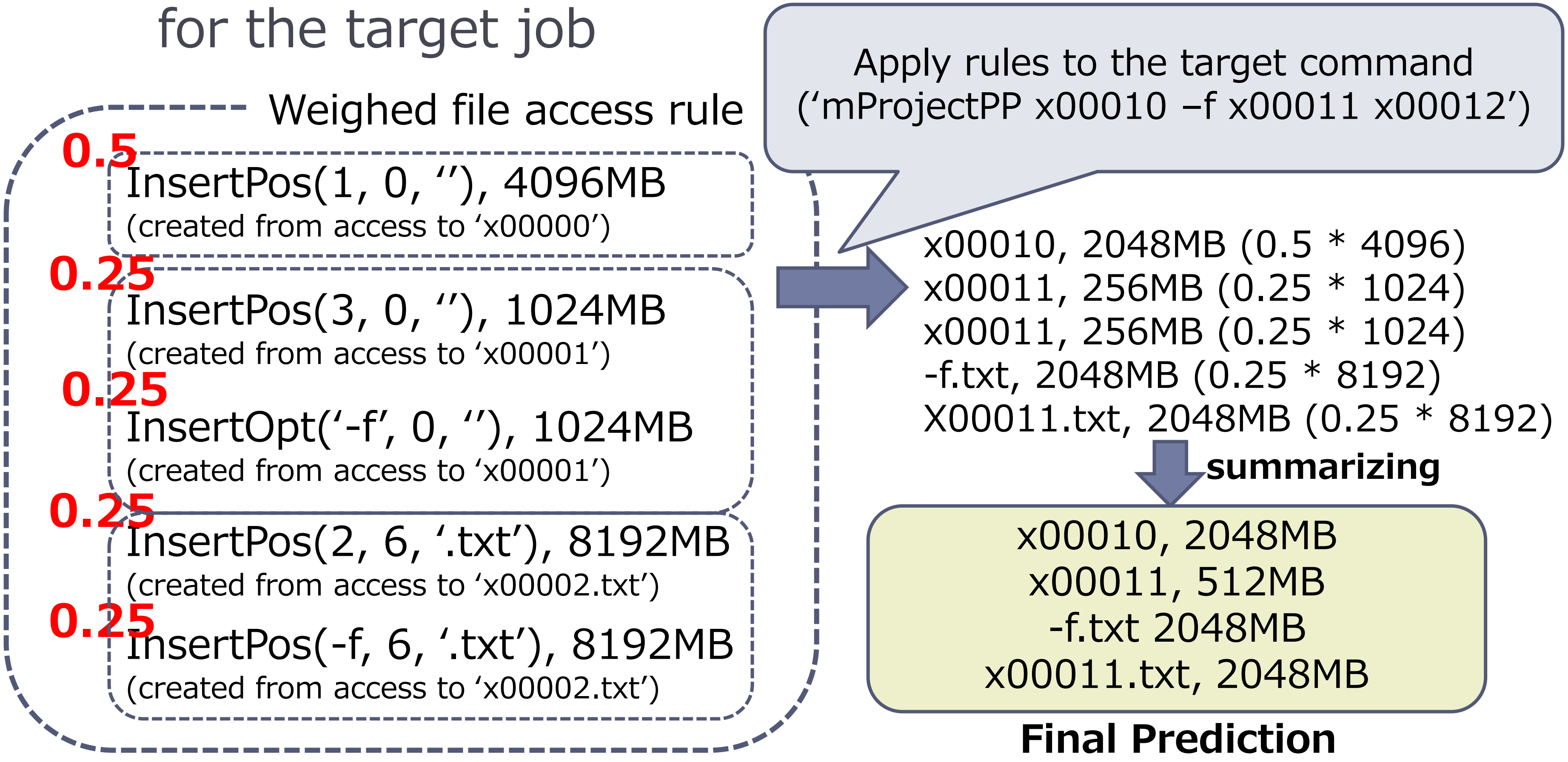
InsertPos(2, 6, ‘.txt’)
(created from access to ‘x00002.txt’)

0.25

InsertPos(-f, 6, ‘.txt’)
(created from access to ‘x00002.txt’)

How to Predict Input Files (3/3)

- 3) Prediction phase
  - Finally predicts input files and expected read sizes for the target job



Weighed file access rule

0.5
InsertPos(1, 0, ‘’), 4096MB
(created from access to ‘x00000’)

0.25
InsertPos(3, 0, ‘’), 1024MB
(created from access to ‘x00001’)

0.25
InsertOpt(‘-f’, 0, ‘’), 1024MB
(created from access to ‘x00001’)

0.25
InsertPos(2, 6, ‘.txt’), 8192MB
(created from access to ‘x00002.txt’)

0.25
InsertPos(-f, 6, ‘.txt’), 8192MB
(created from access to ‘x00002.txt’)

Apply rules to the target command
(‘mProjectPP x00010 –f x00011 x00012’)

x00010, 2048MB (0.5 * 4096)
x00011, 256MB (0.25 * 1024)
x00011, 256MB (0.25 * 1024)
-f.txt, 2048MB (0.25 * 8192)
X00011.txt, 2048MB (0.25 * 8192)

summarizing

x00010, 2048MB
x00011, 512MB
-f.txt 2048MB
x00011.txt, 2048MB

Final Prediction

Job Scheduling Mechanism

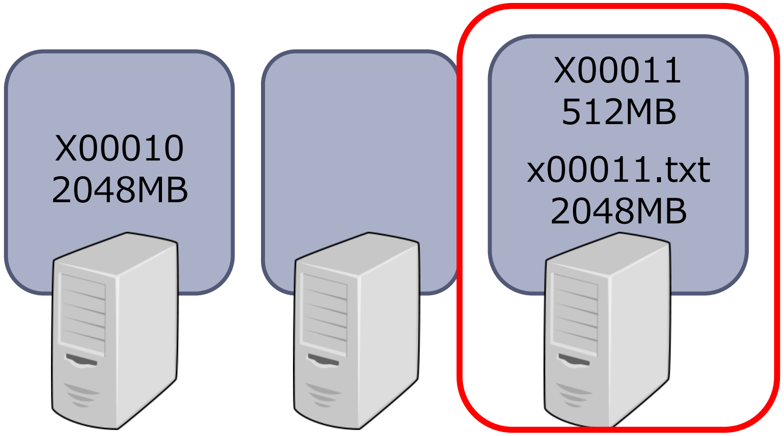
- Existing job scheduling mechanism:
  - considers only computation resources without taking computation-data affinity into account
- New job scheduling mechanism:
  - asks metadata server the locations of the predicted files
  - prioritizes nodes with more data that the job is predicted to read

Note: if the node is fully occupied by jobs, the job is dispatched to another node according to the priority

Final Prediction

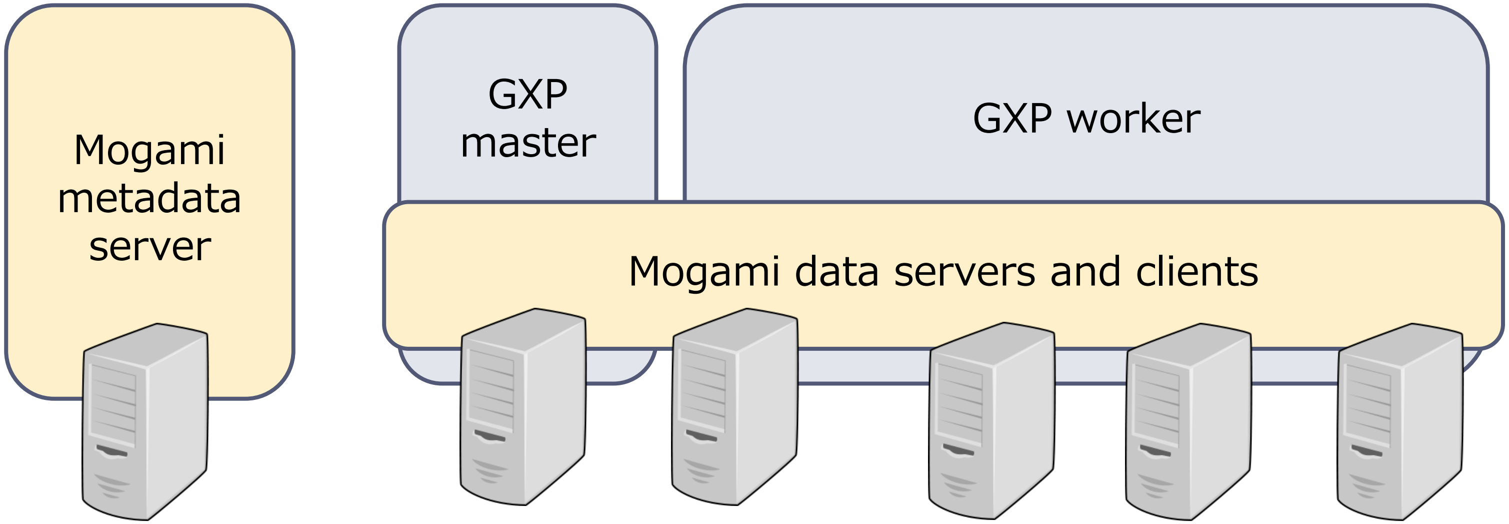
x00010, 2048MB
x00011, 512MB
-f.txt 2048MB
x00011.txt, 2048MB

Target command:
'mProjectPP x00010 –f x00011 x00012'



Experimental Environment

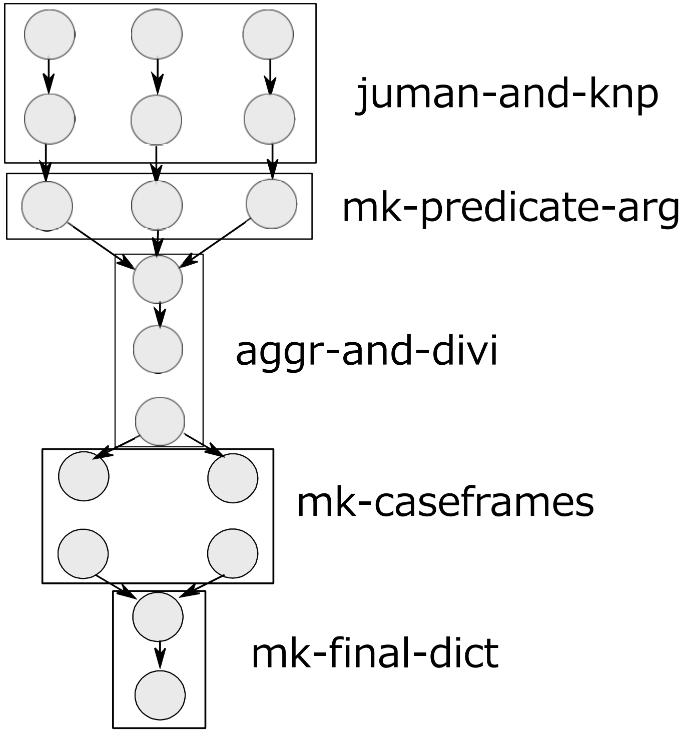
- Linux cluster with 6 nodes
- Each node has
  - Intel Xeon CPU E5410 (2.33GHz, 4 core w/o HT)
  - 32GB memory
  - 900GB of its own local storage
- Every node is connected with each other by 1Gbps Ethernet



Can execute 20 jobs in parallel at the same time

Case Frame Construction: Application Overview & Datasets

- A text processing application for constructing data structures called case frames [D. Kawahara et al. 2000] from Japanese web corpus
  - Applicable for applications such as searching, summarizing and translating



juman-and-knp

mk-predicate-arg

aggr-and-divi

mk-caseframes

mk-final-dict

datasets

	Small	Large
Input file size	400kB on average	1.8MB on average
# of input files	4	24
Sum of data size	1.7MB	39MB
# of jobs	45	1600

used small dataset for a profiling run and applied our proposal to a run with large dataset

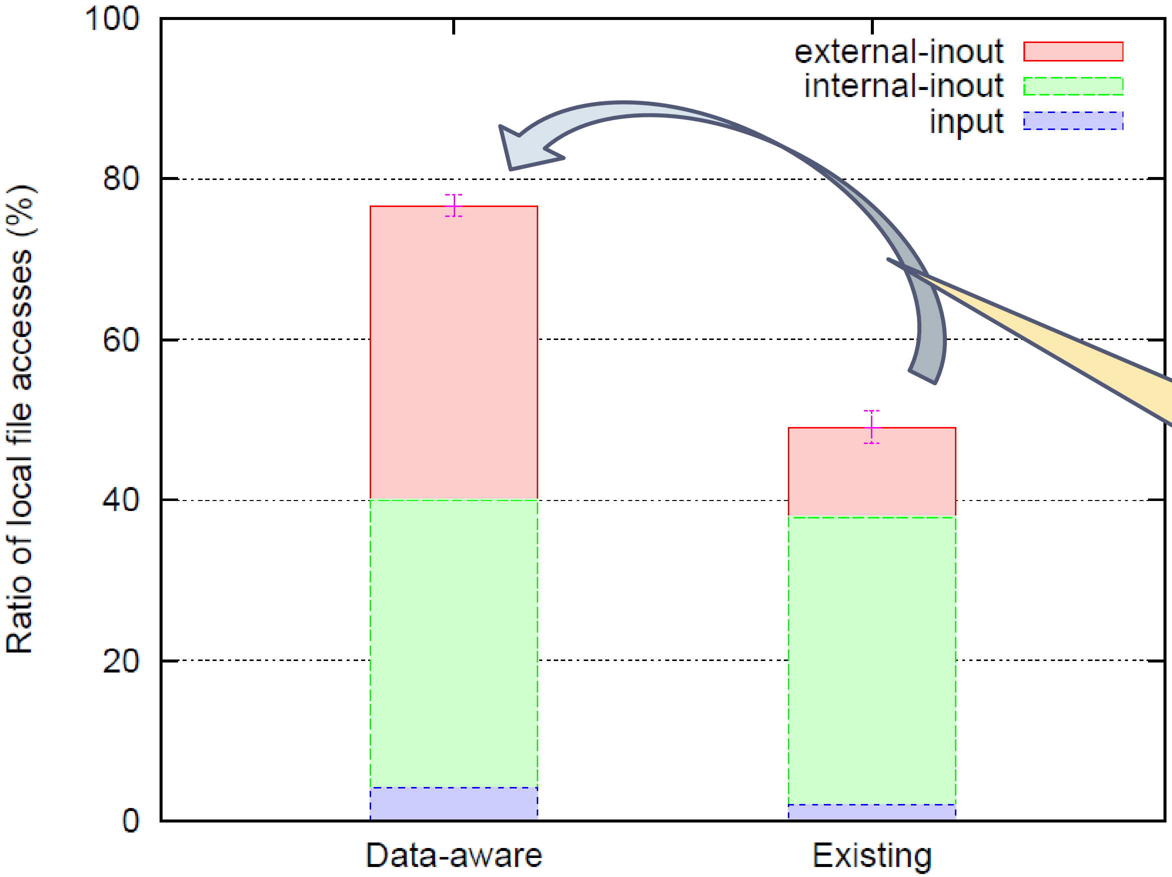
Case Frame Construction: Accuracy Rate

Application	# of jobs	Precision (%)	Recall (%)
juman-and-knp	48	100 (144/144)	84.21 (144/171)
mk-predicate-arg	24	97.959 (144/147)	100 (144/144)
aggr-and-divi	5	69.230 (9/13)	27.272 (9/33)
mk-caseframes	1520	100 (144/144)	100 (144/144)
mk-final-dict	2	90.697 (39/43)	37.50 (39/104)

Almost all input files can be predicted with a high accuracy rate by our proposal

Case Frame Construction: Results (1/2)

- Ratio of local file accesses



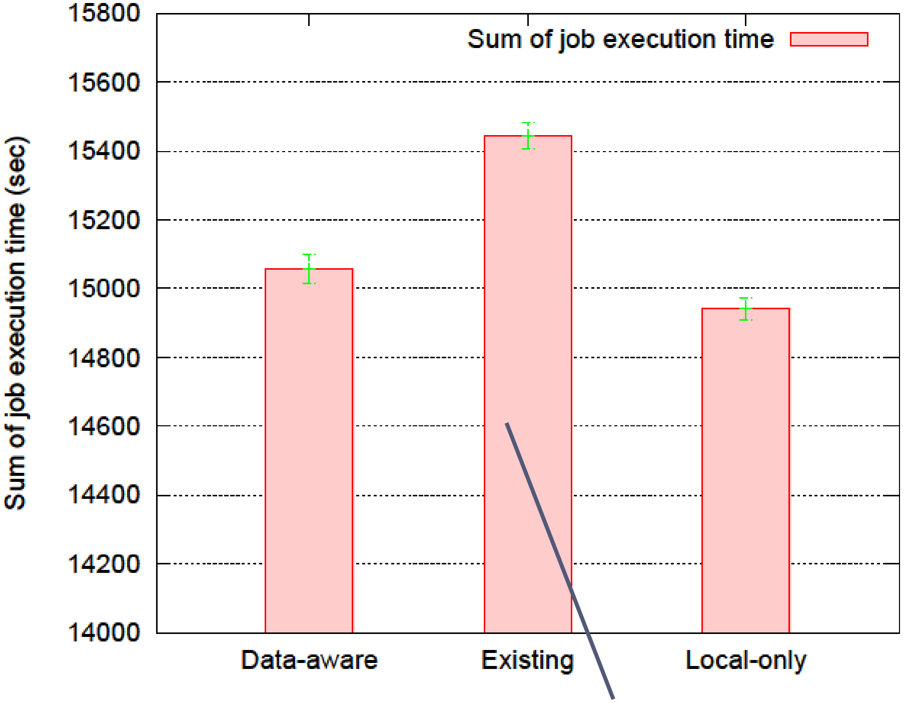
external-inout: read a file created by a different job
internal-inout: read a file created by the job itself
input: read a file that exists before executing the workflow

Increased significantly from 50%
to 75% by proposal

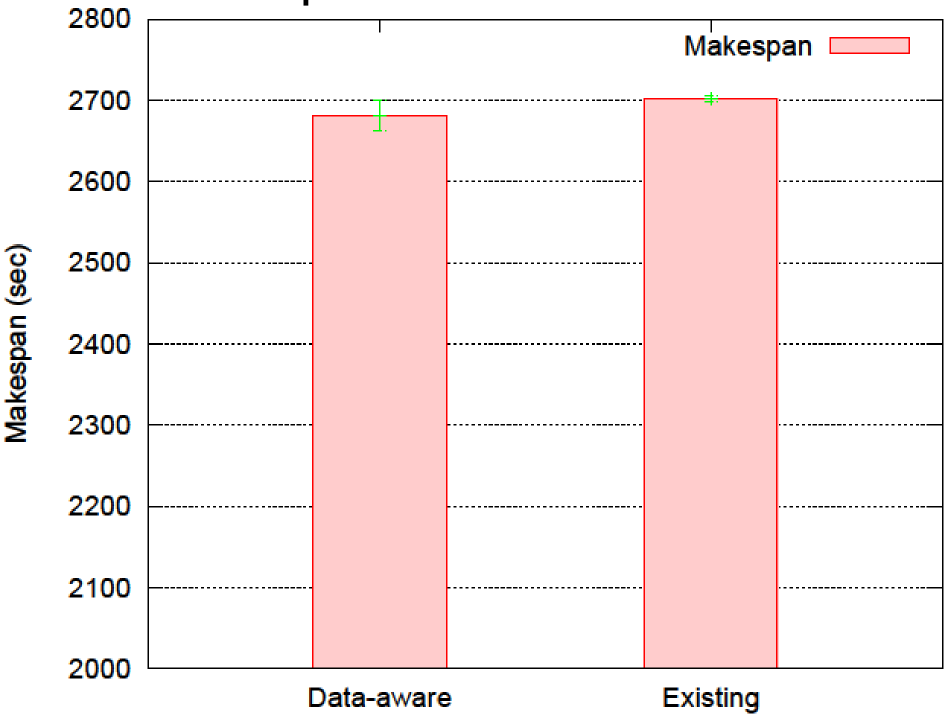
Data-aware: execution w/ proposed method
Existing: execution w/o proposed method

Case Frame Construction: Results (2/2)

- Sum of job execution time



- Makespan

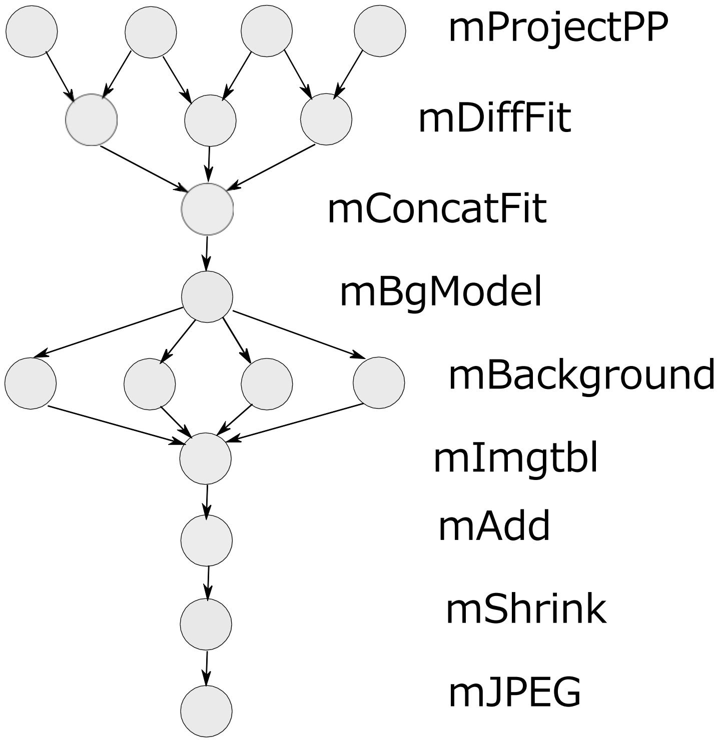


Decreased by about 400 sec to 15,050 sec, which is pretty close to that of Local-only (ideal)

Local-only: execution using only single node (all file accesses are within local storage)

Montage: Application Overview & Datasets

- An Astronomic application for constructing custom astronomical image mosaics of the sky
- Modules such as ‘mProjectPP’ written in C



mProjectPP

mDiffFit

mConcatFit

mBgModel

mBackground

mImgtbl

mAdd

mShrink

mJPEG

datasets

	Small	Large
Input file size	2.1MB	1.7MB or 2.1MB
# of input files	6	609
Sum of data size	12.6MB	1270MB
# of jobs	19	1542

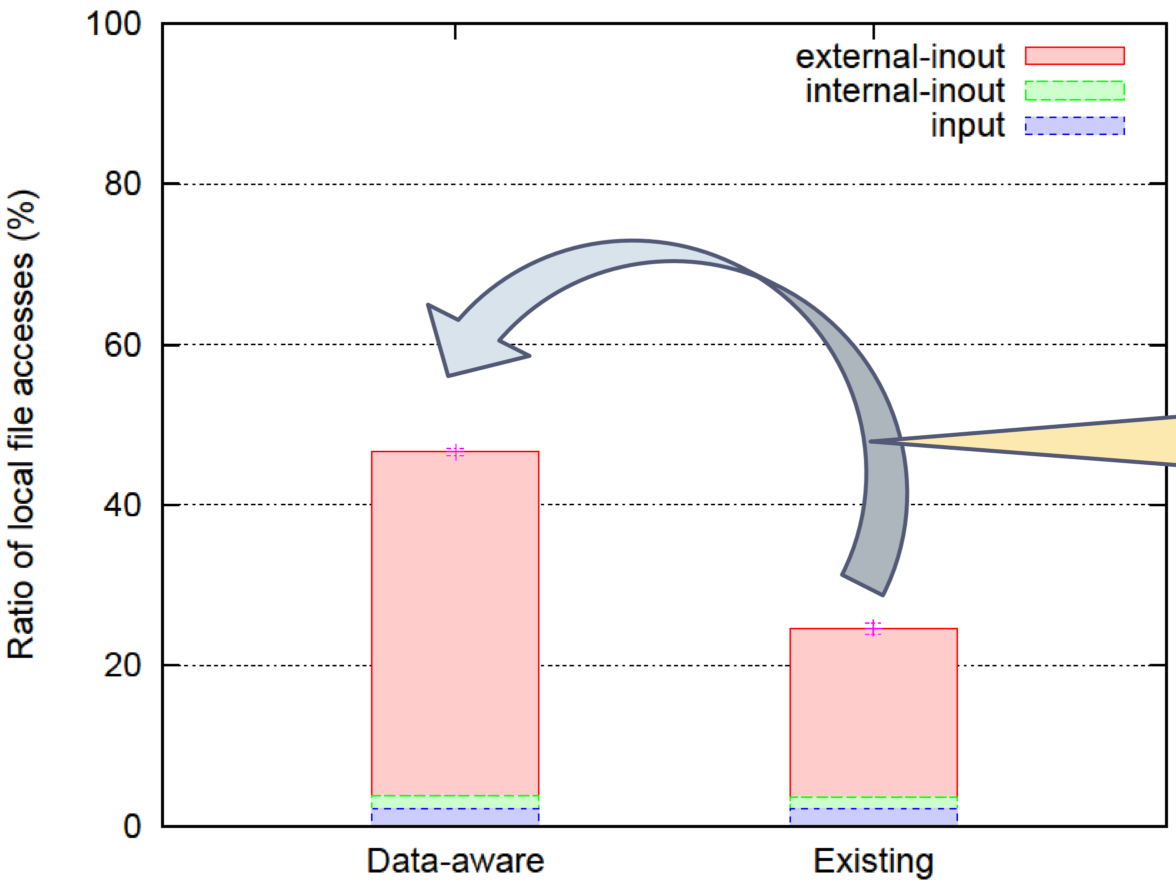
used small dataset for a profiling run and applied our proposal to a run with large dataset

Montage: Accuracy Rate

Application	# of jobs	Precision (%)	Recall (%)
mProjectPP	308	99.84 (1230/1232)	100 (1230/1230)
mDiffFit	913	55.88 (6437/11520)	88.01 (6437/7314)
mConcatFit	1	33.33 (2/6)	0.22 (2/914)
mBgModel	1	100 (2/2)	100 (2/2)
mBackground	308	99.89 (1846/1848)	100 (1846/1846)
mImgtbl	1	33.33 (2/6)	33.33 (2/6)
mAdd	5	33.90 (20/59)	2.78 (20/720)
mShrink	4	50 (4/8)	100 (4/4)
mJPEG	1	100 (1/1)	100 (1/1)

Montage: Results (1/2)

- Ratio of local file accesses



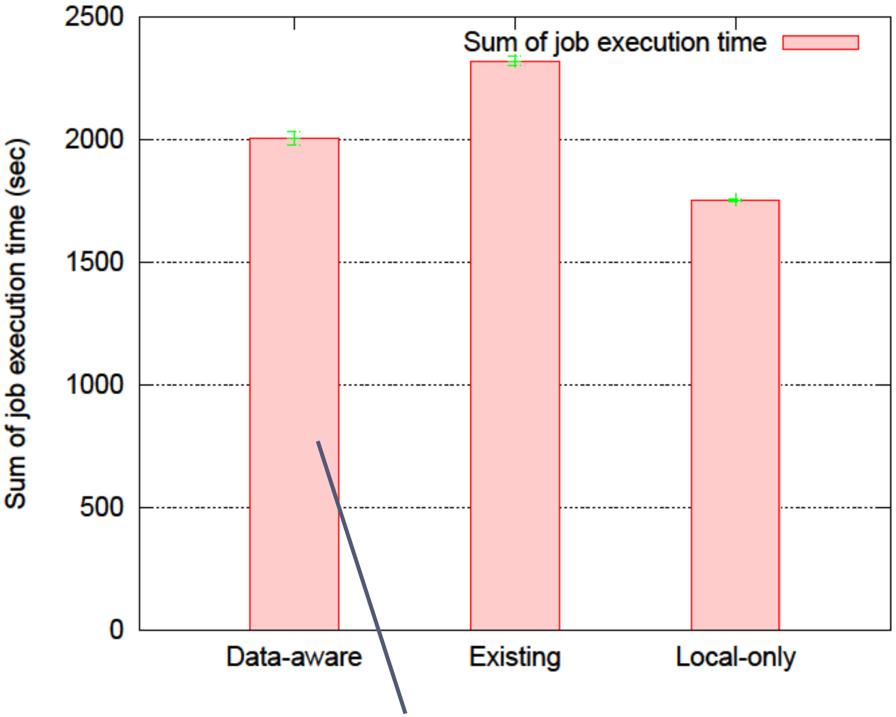
external-inout: read a file created by a different job
internal-inout: read a file created by the job itself
input: read a file that exists before executing the workflow

Increased from 23% to 45% by proposal

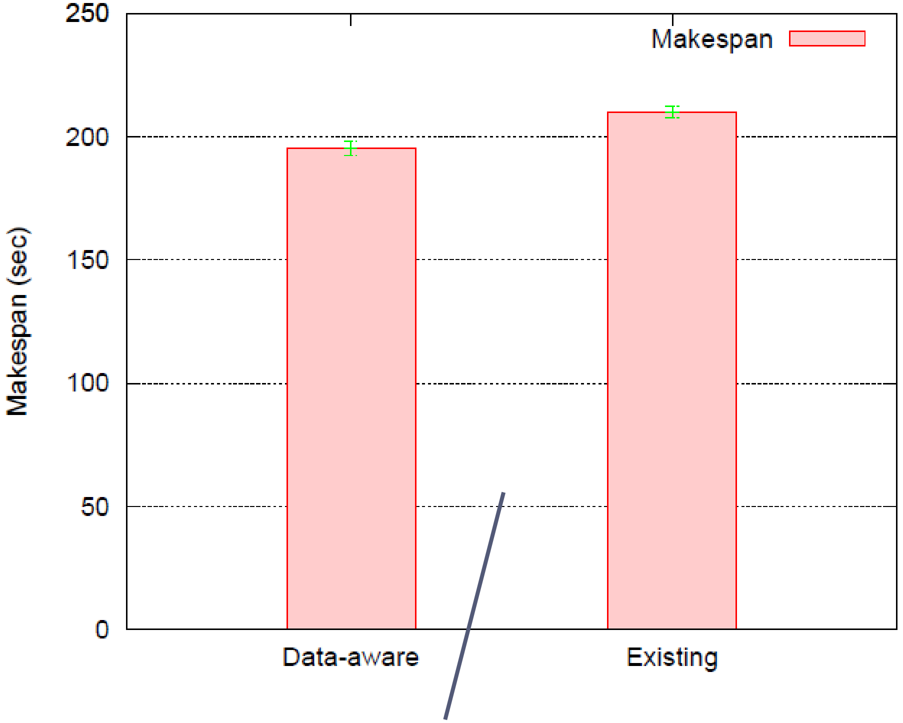
Data-aware: execution w/ proposed method
Existing: execution w/o proposed method

Montage: Results (2/2)

- Sum of job execution time



- Makespan



Decreased by 13% from 2320 sec to 2000 sec

Decreased by 15 sec, which is 7.5% of that of Existing

Local-only: execution using only single node (all file accesses are within local storage)

Conclusion & Future Work

- Summary
  - Proposed a method to deduce input/output files by using file access history gathered in a profiling run
  - Implemented the method for GXP Make and Mogami
  - Evaluated our proposal using 2 real workflow applications
- Future work
  - Enhance the job scheduling algorithm with consideration of jobs dispatched in the near future
  - Utilize other information gathered in profiling run, such as job execution time and file access time

Thank you!

- Questions?

- Contact
  - mikity@eidos.ic.i.u-tokyo.ac.jp
  - http://www.eidos.ic.i.u-tokyo.ac.jp (Taura Lab.)