Balanced Task Clustering in Scientific Workflows

Weiwei Chen*, Rafael Ferreira da Silva‡, Ewa Deelman*, Rizos Sakellariou§
*University of Southern California, Information Sciences Institute, USA
‡Université de Lyon, CNRS, INSERM, CREATIS, Villeurbanne, France
§University of Manchester, School of Computer Science, Manchester, U.K.

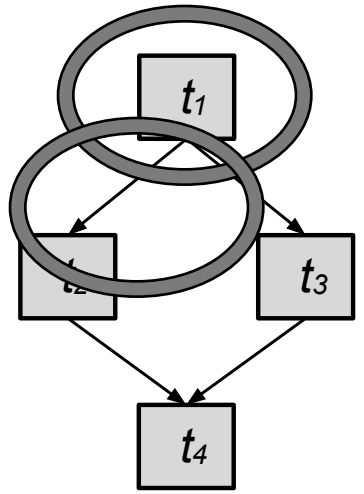
Oct 24, 2013, CNCC, Beijing

Outline

- Introduction
  - Workflow Model
  - Execution Environment
- Task Clustering
- Imbalance
  - Runtime Imbalance
  - Dependency Imbalance
- Balancing Methods
  - Horizontal Runtime Balancing
  - Horizontal Impact Factor Balancing
  - Horizontal Distance Balancing
- Experiments
- Conclusion

What is a workflow?

- Scientific workflows are used to orchestrate complex, multi-stage computations
- Scientific workflows are usually expressed as DAGs (Directed Acyclic Graphs)
- A DAG has
  - Task: a program specified by user
  - Data dependency: data transfer between tasks



DAG

Examples of Scientific Workflows

- LIGO: detect and measure gravitational waves predicted by general relativity, Caltech etc.
- Epigenomics: DNA sequencing and mapping, USC
- Montage: generate science-grade mosaics of the sky, NASA IPAC

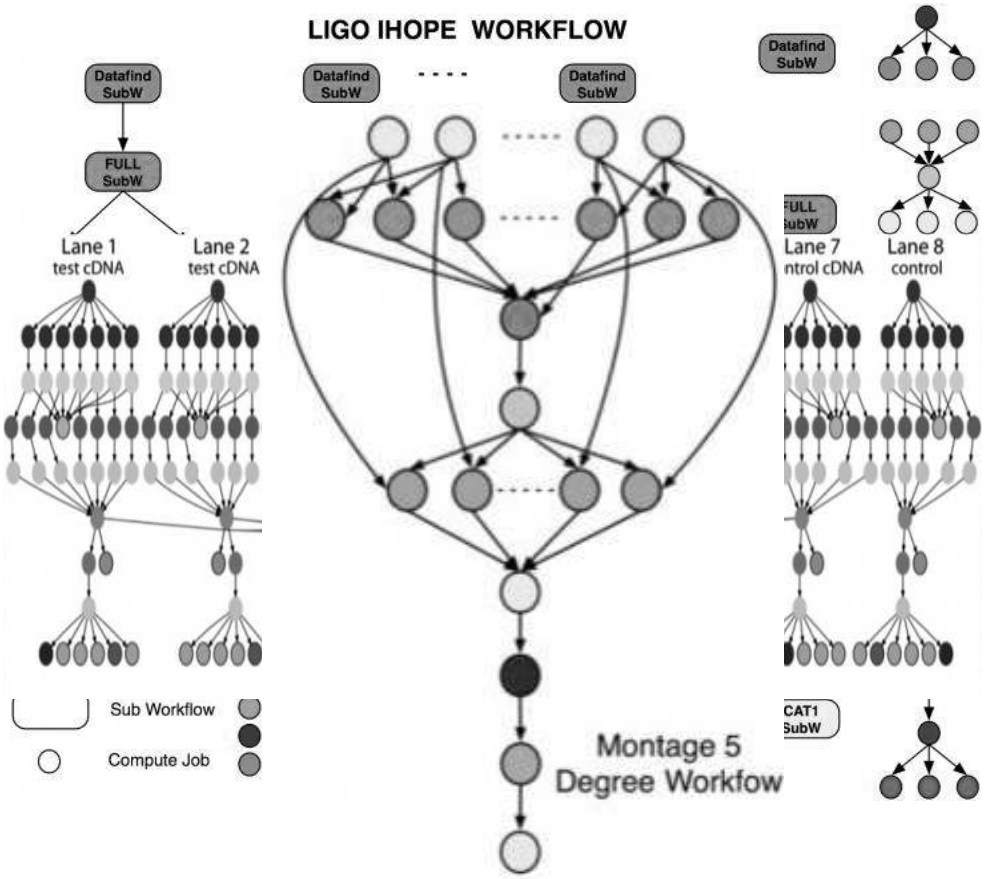
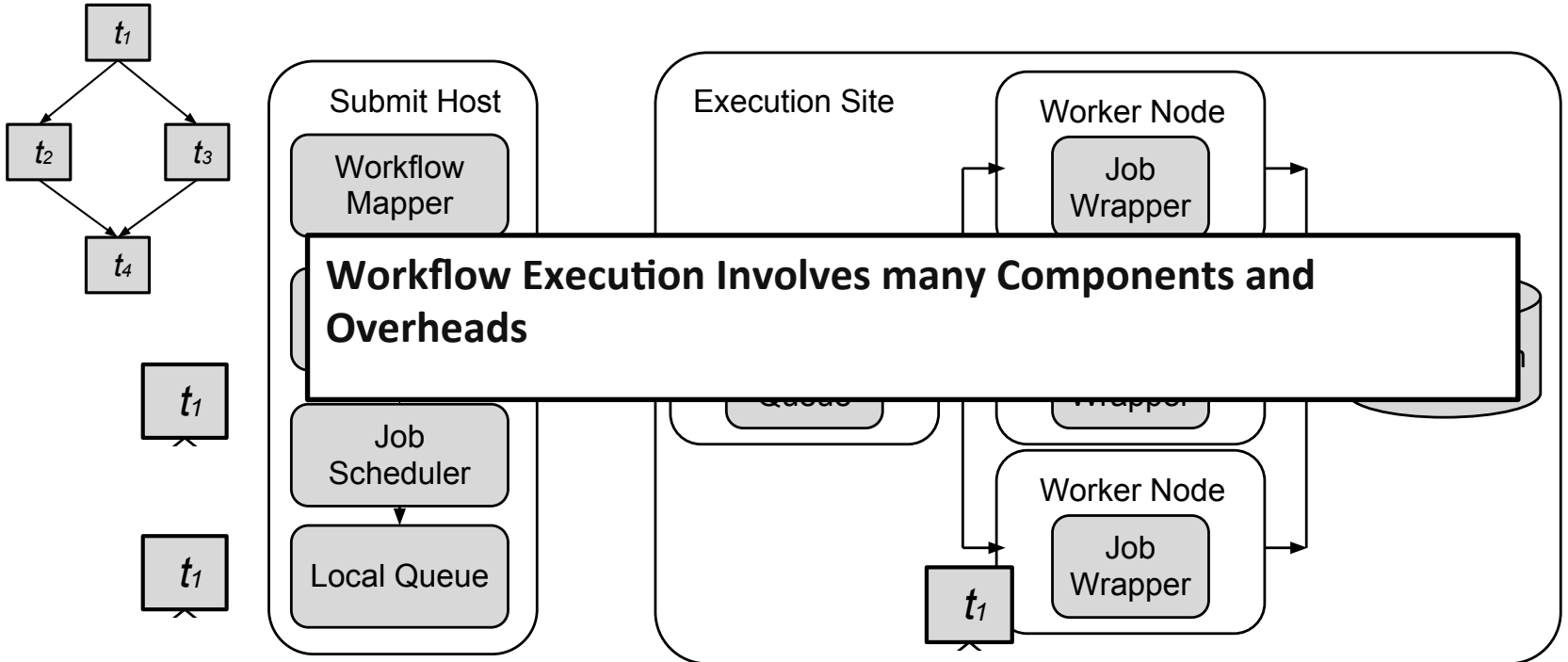
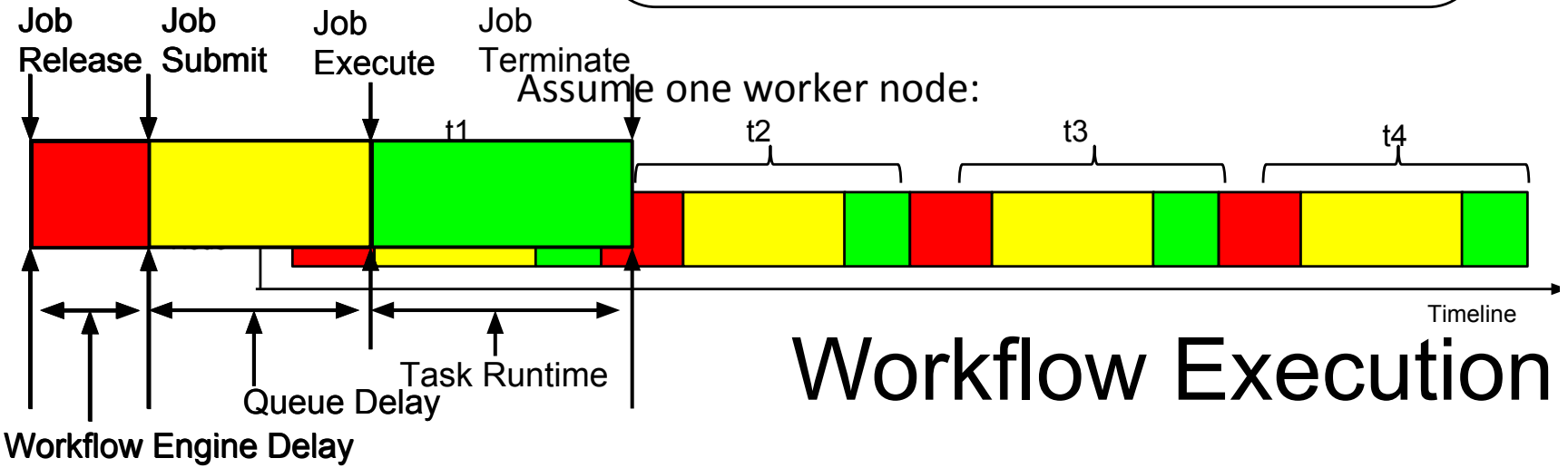


Image Source: http://pegasus.isi.edu/applications/



Workflow Execution Involves many Components and Overheads



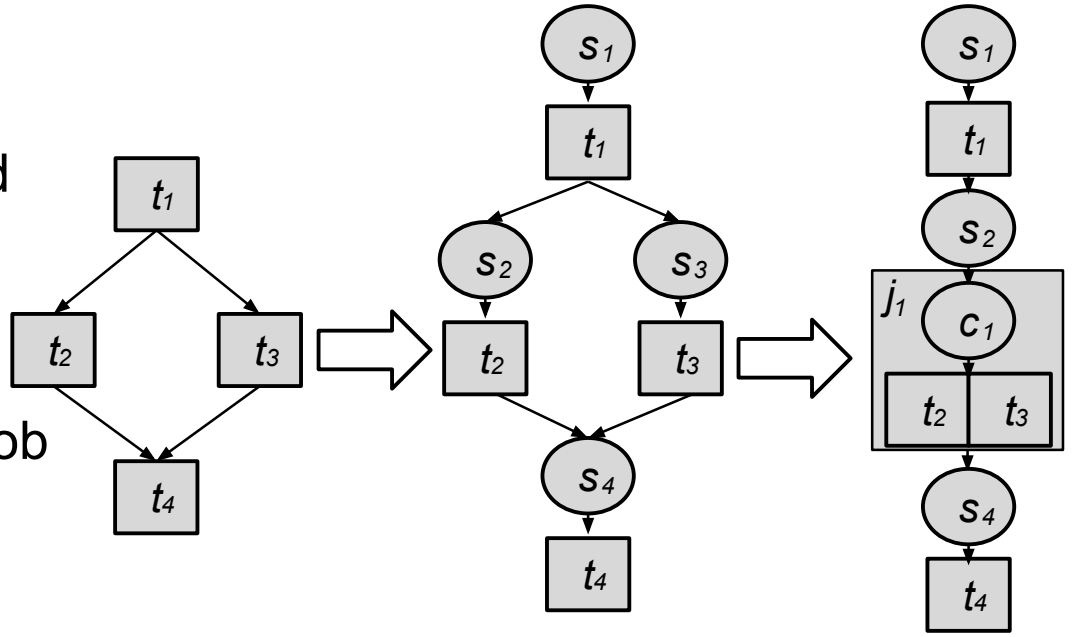
Workflow Execution

- System Overhead> Task Runtime
- Number of Tasks>Number of Resources

Workflow	Venue	Number of Nodes	Number of Tasks	Avg. Workflow Engine Delay (second/ percentage over task runtime)	Avg. Queue Delay (second/ percentage over task runtime)	Avg. Task Runtime(second)
SIPHT	UW Madison	8	33	17(85%)	69(345%)	20
Broadband	Amazon EC2	8	770	17(6%)	945(307%)	308
Epigenomics	Amazon EC2	8	83	6(4%)	311(197%)	158
CyberShake	Skynet	5	24142	12(243%)	188(3768%)	5
Periodogram	OSG	39	235300	464(2109%)	2230(10136%)	22
Montage	USC/ISI	20	10427	182(35%)	26136(4997%)	523

Task Clustering

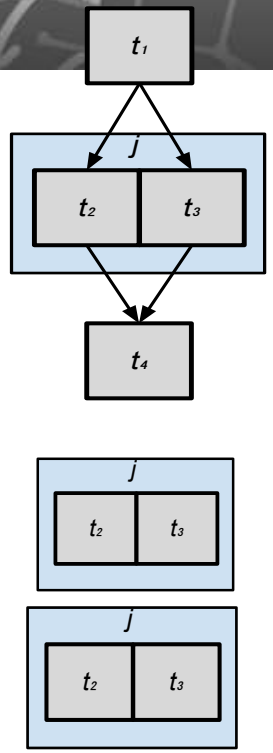
- ODAG model (overhead aware DAG): an overhead added as a node
- Merge small tasks into jobs and submit jobs to the execution system.
- In the execution site, a job wrapper extracts tasks from a job and execute them in a worker node
- Horizontal Clustering merges tasks at the same workflow level

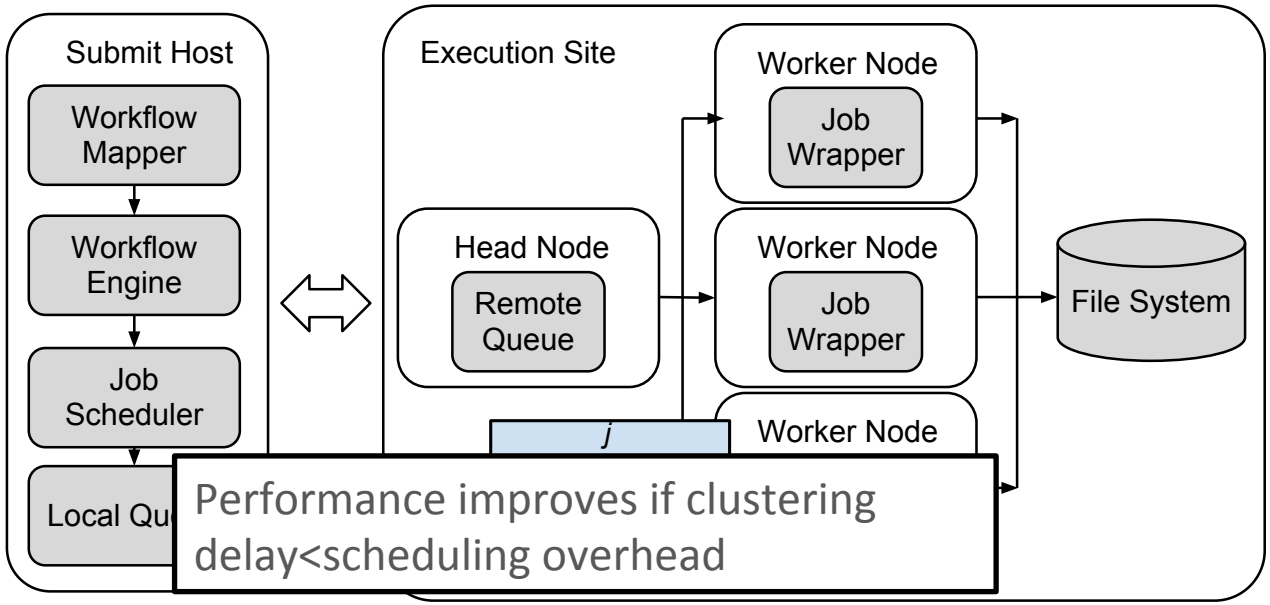


Pegasus

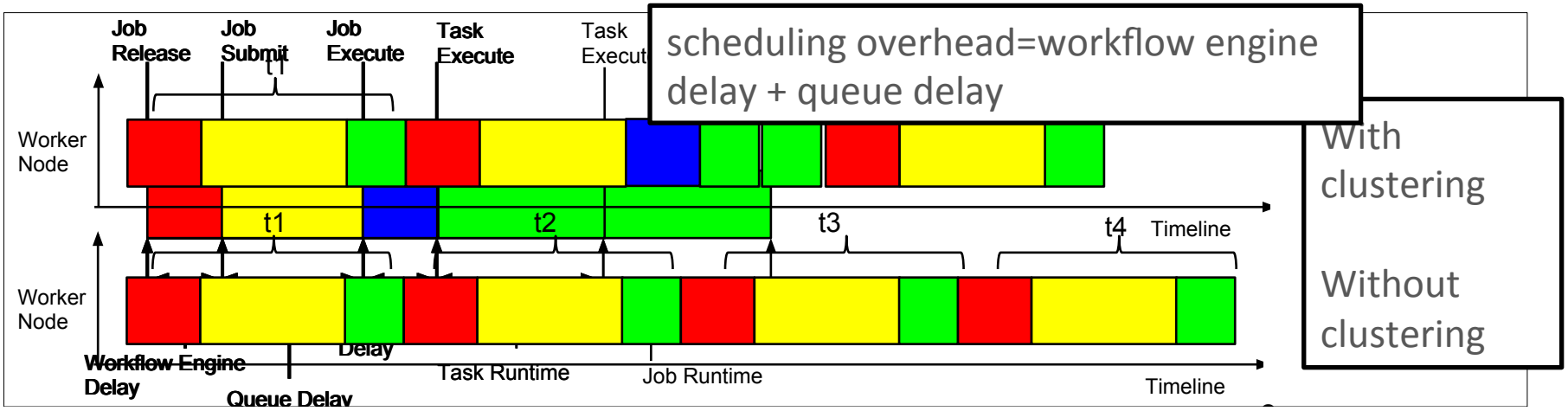
Workflow Management System

http://pegasus.isi.edu





Performance improves if clustering delay<scheduling overhead



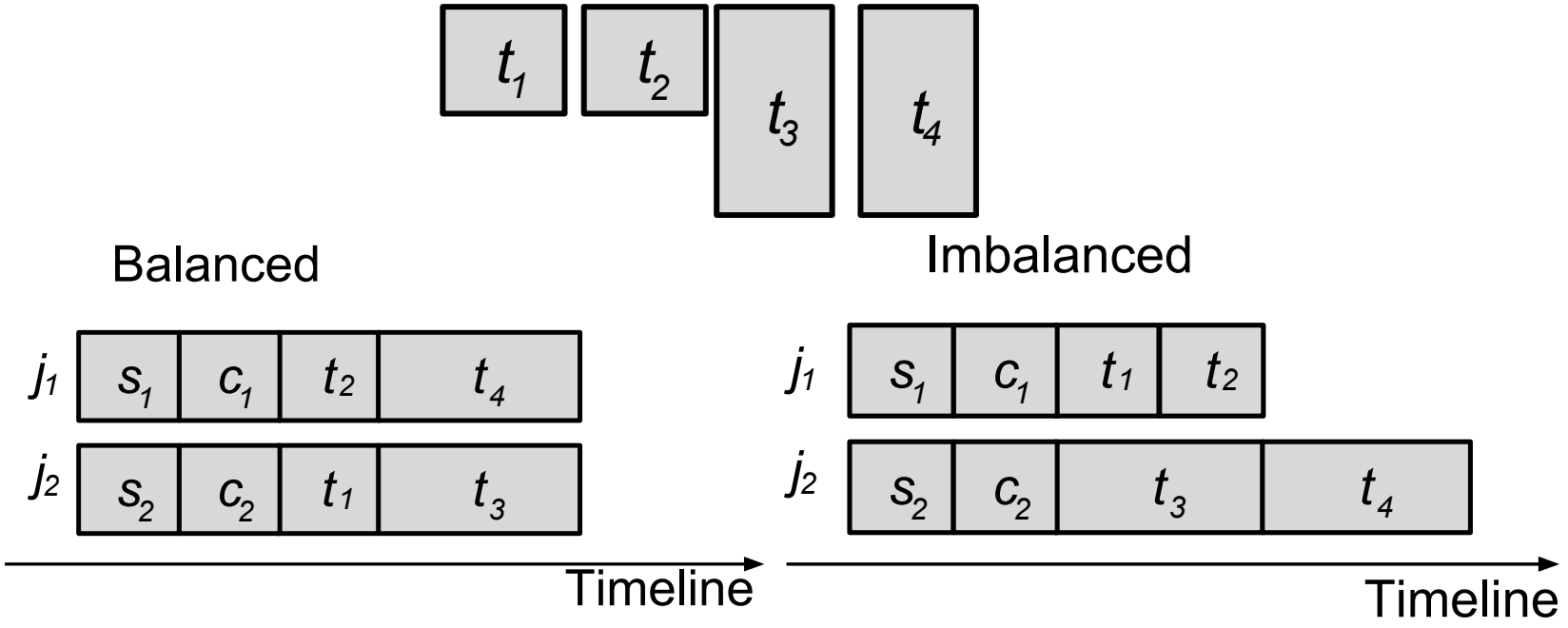
scheduling overhead=workflow engine delay + queue delay

With clustering

Without clustering

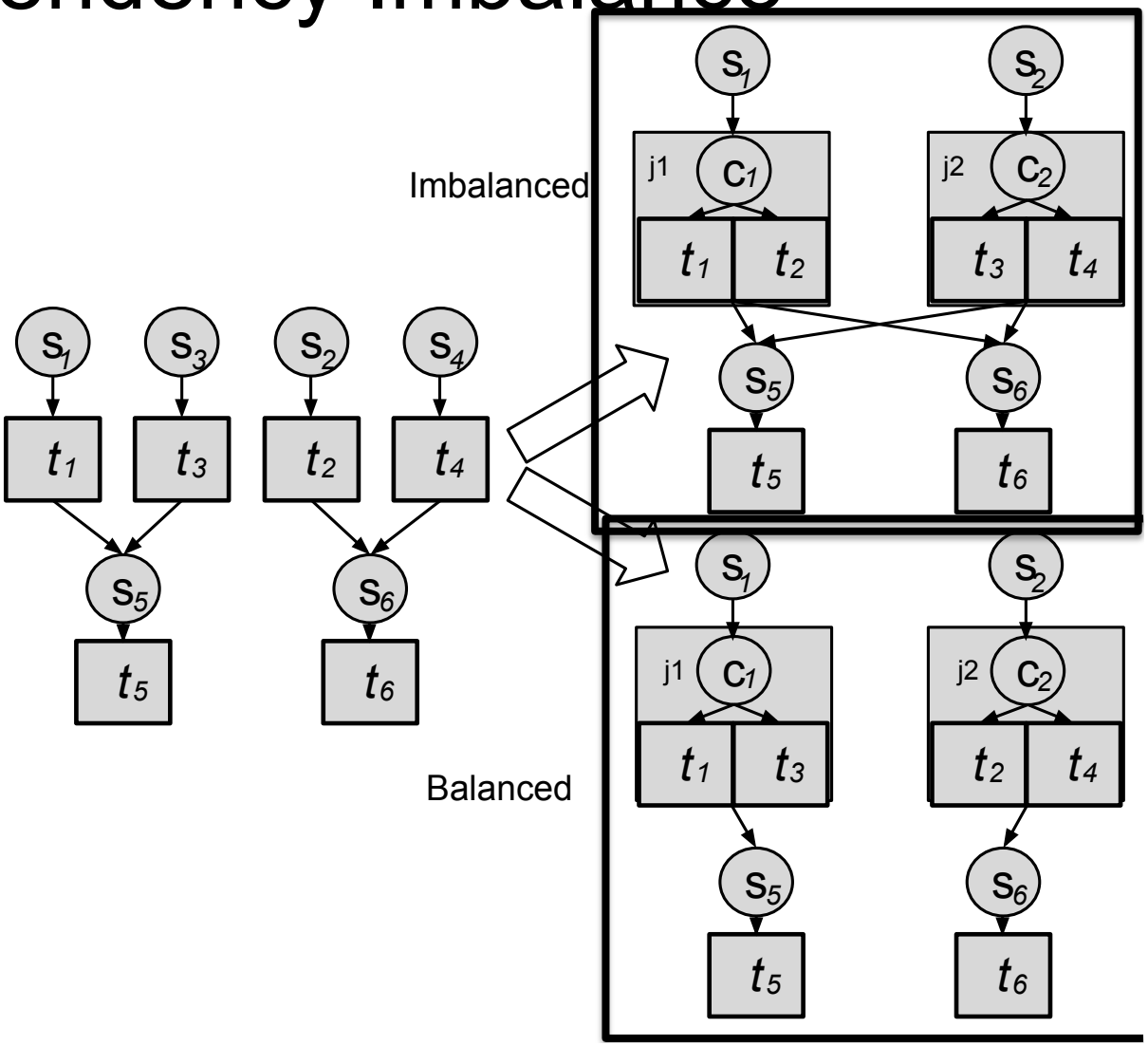
Challenge: Runtime Imbalance

- Runtime Imbalance describes the difference of the task/ job runtime of a group of tasks/jobs.



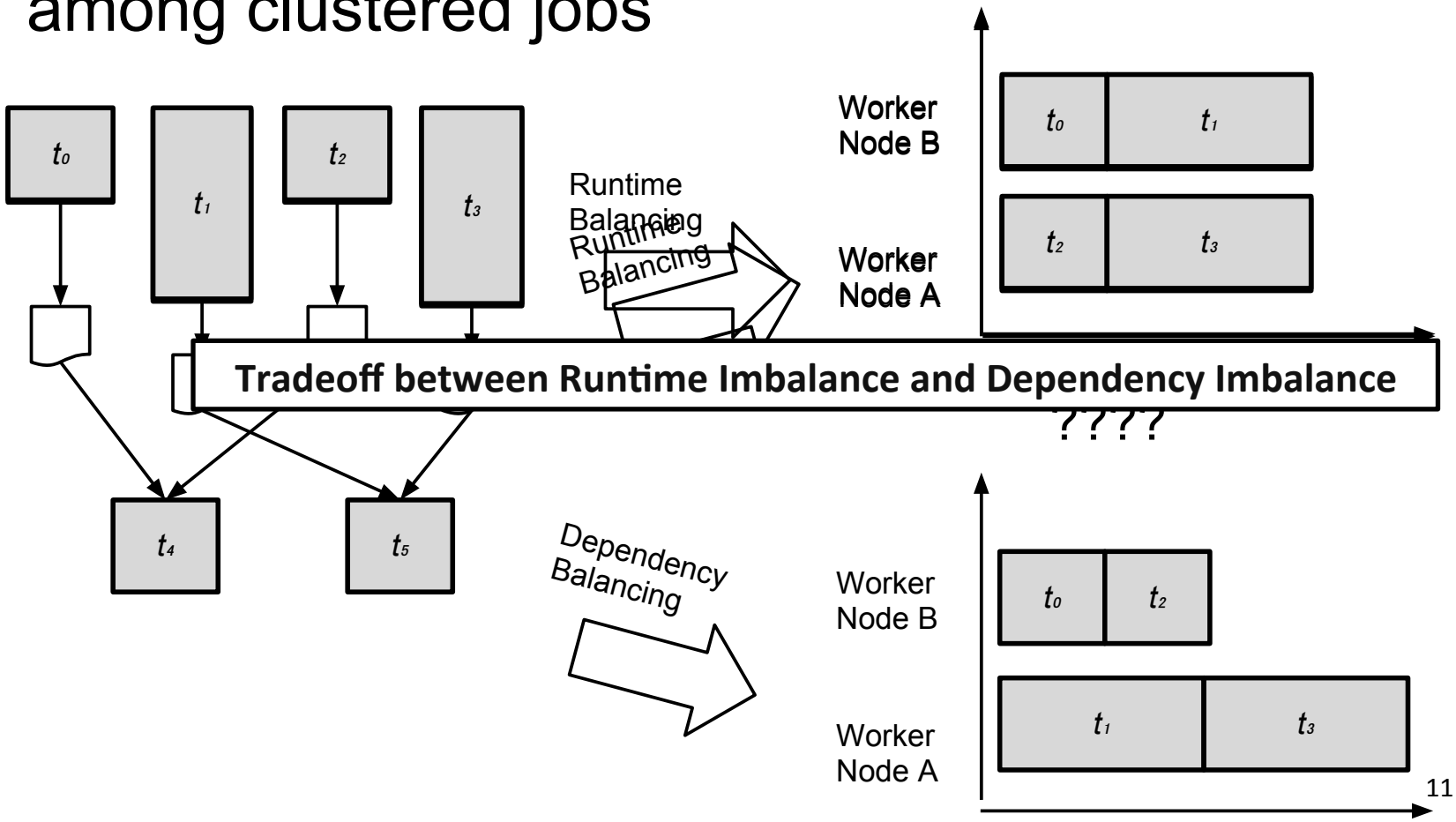
Challenge: Dependency Imbalance

Task clustering at one horizontal level forces the tasks at the next levels to have severe data locality problem and loss of parallelism



Balancing Computation and Data Dependencies

- Task Clustering varies the load distribution among clustered jobs



Runtime Variance

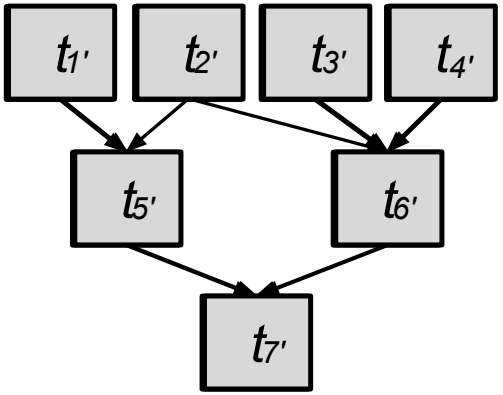
- Horizontal Runtime Variance (HRV) as the standard deviation in task runtime per workflow level.
- Intuition: at the same horizontal level, the job with the longest runtime often controls the release of the next level jobs.
- Other RV: Pipeline Runtime Variance (PRV) as the standard deviation in the sum of task runtime in the same pipeline
- Intuition: the pipeline with the longest runtime controls the finish time of the workflow

How to measure Dependency Imbalance?

- Impact Factor:
  - Already used in web search engine
  - Defined iteratively as the sum of the proportional IFs of its children

$$IF(t_u) = \sum_{t_v \in Child(t_u)} \frac{IF(t_v)}{\|Parent(t_v)\|}$$

- Characteristics
  - The larger, the more important to the workflow (more control of other jobs, bottleneck, etc.)



$$IF(t_7) = 1.0$$

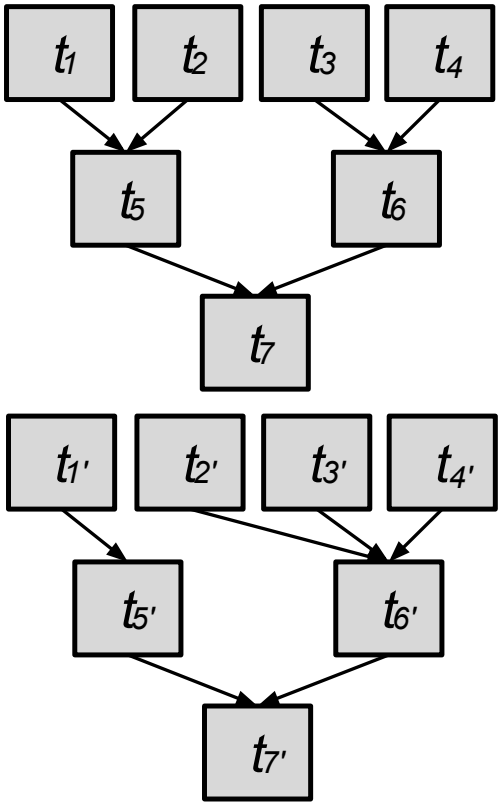
$$IF(t_{5'}) = IF(t_{6'}) = IF(t_{7'}) / 2 = 0.5$$

$$IF(t_{1'}) = IF(t_{2'}) = IF(t_3) = IF(t_4) = IF(t_6) / 2 = 0.25$$

$$IF(t_{2'}) = IF(t_{3'}) = IF(t_{4'}) = IF(t_{6'}) / 3 = 0.17$$

How to measure Dependency Imbalance?

- Horizontal Impact Factor Variance (HIFV) is the standard deviation of IFs of tasks at the same horizontal level



$$IF(t_7) = 1.0$$

$$IF(t_6) = IF(t_5) = IF(t_7)/2 = 0.5$$

$$IF(t_1) = IF(t_2) = IF(t_3) = IF(t_4) = IF(t_6)/2 = 0.25$$

$$HIFV(t_1, t_2, t_3, t_4) = 0.0$$

The larger HIFV is, the more irregular, imbalanced the structure is

$$IF(t_{7'}) = 1.0, IF(t_{6'}) = IF(t_{5'}) = IF(t_{7'})/2 = 0.5$$

$$IF(t_{1'}) = IF(t_{5'}) = 0.5$$

$$IF(t_{2'}) = IF(t_{3'}) = IF(t_{4'}) = IF(t_{6'})/3 = 0.17$$

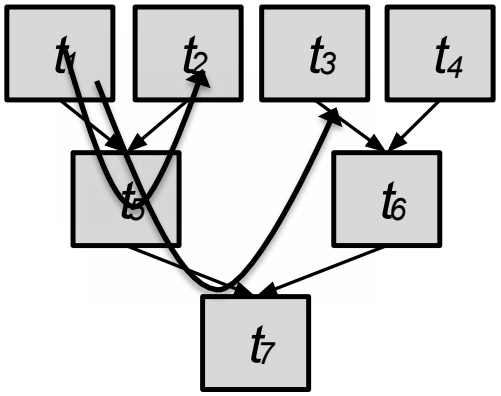
$$HIFV(t_{1'}, t_{2'}, t_{3'}, t_{4'}) = 0.17$$

Distance Metric

- Distance: the shortest path of two tasks
- Distance Metric

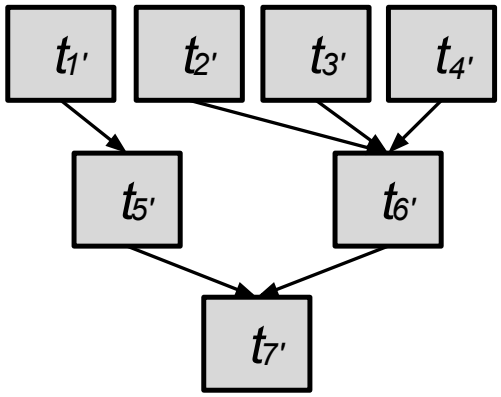
$$D = \{d_{ij}\}$$

- Horizontal Distance Variance (HDV): the standard deviation of the distance metric (symmetric, diagonal elements are 0). (i>j)
- The larger HDV is, the more irregular and imbalanced the structure is.



d_{ij}	t_1	t_2	t_3	t_4
t_1	0	2	4	4
t_2		0	4	4
t_3			0	2
t_4				0

$$HDV(t_1, t_2, t_3, t_4)$$
$$= \sigma(d_{12}, d_{13}, d_{14}, d_{23}, d_{24}, d_{34})$$
$$= 1.0$$



d_{ij}	t_1	t_2	t_3	t_4
t_1	0	4	4	4
t_2		0	2	2
t_3			0	2
t_4				0

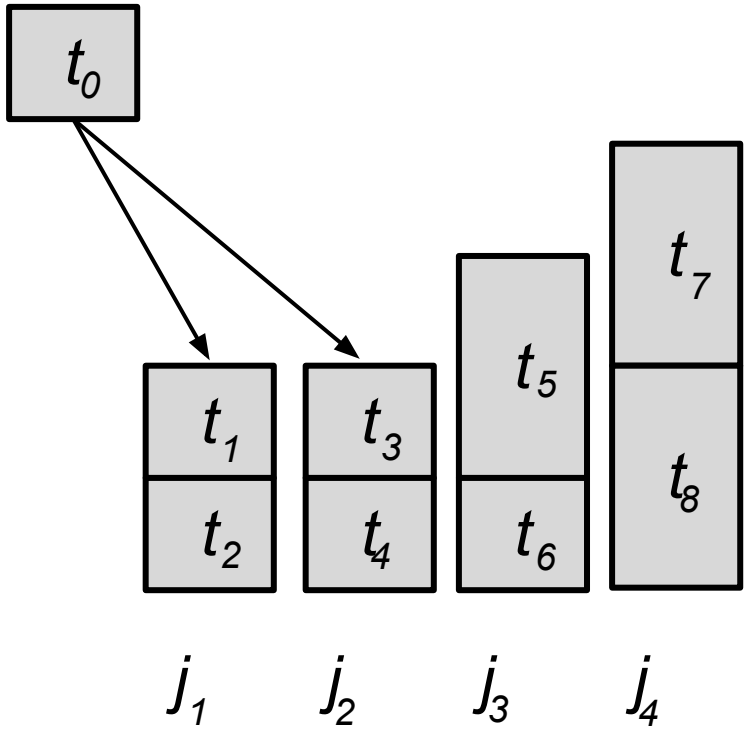
$$HDV(t_{1'}, t_{2'}, t_{3'}, t_{4'}) = 1.1$$

Metrics

- Horizontal Runtime Variance (HRV): O(n)
- Horizontal Impact Factor Variance (HIFV): O(n)
- Horizontal Distance Variance (HDV): O(n^2)
- Magic: Quantitatively Measure Runtime Imbalance and Dependency Imbalance
- Map structural information into numerical values

Balancing Methods:

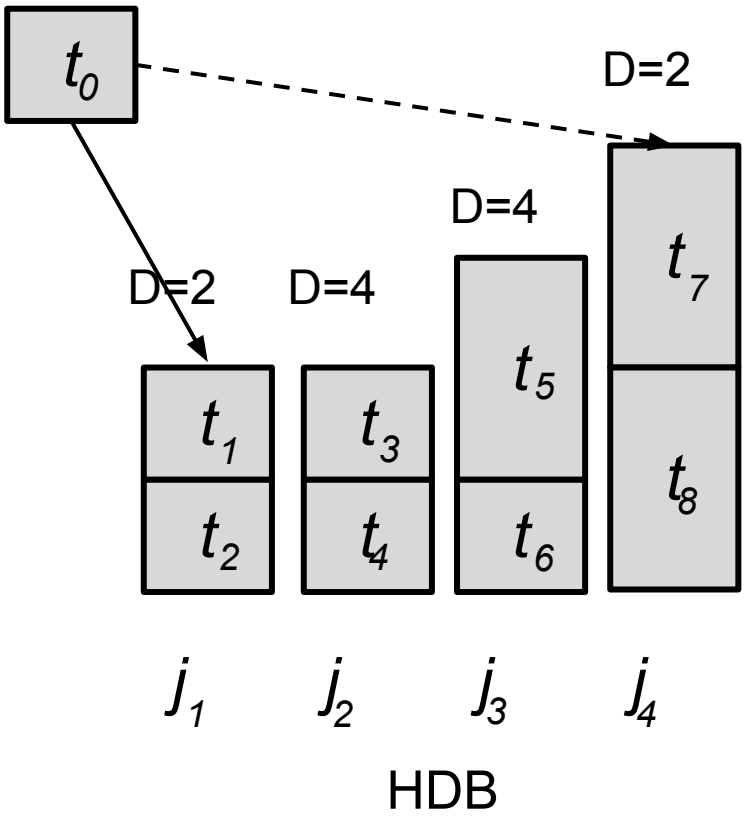
- Horizontal Runtime Balancing (HRB): add the longest task to the shortest job



HRB

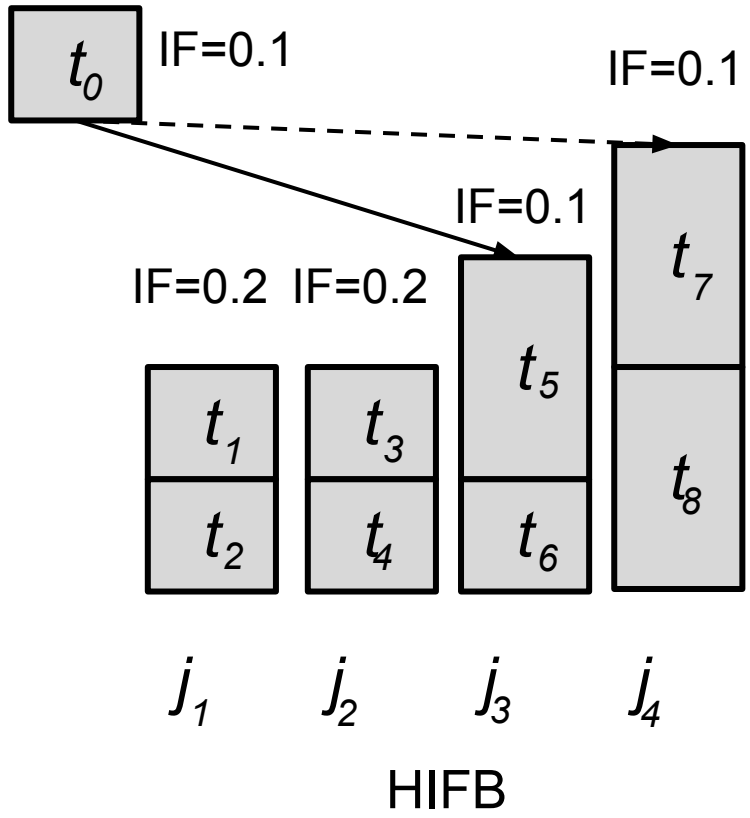
Horizontal Distance Balancing (HDB)

- Sort tasks based on distance, then HRB



Horizontal Impact Factor Balancing (HIFB)

- Group tasks based on IFs then HRB



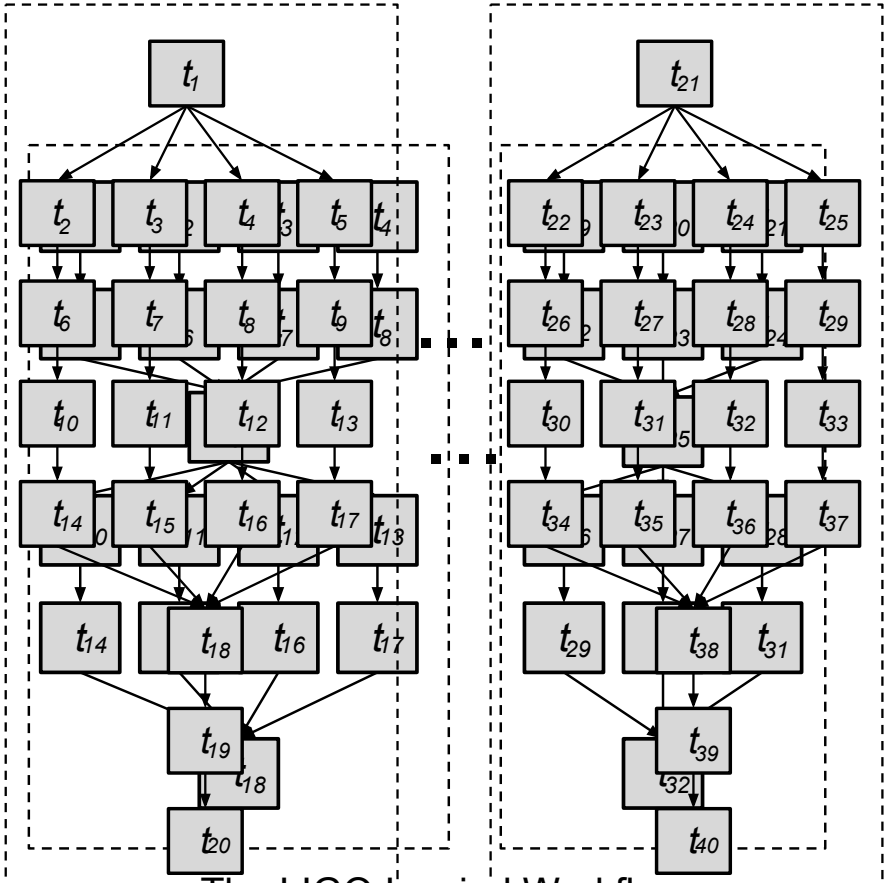
Experiments and Setup

- We extended the WorkflowSim[1]: simulator with balanced clustering methods and imbalance metrics to simulate a distributed environment.
- 20 virtual machines (worker nodes): a typical environment such as Amazon EC2 and FutureGrid. Each virtual machine has 512MB of memory.
- Evaluate the performance of our methods when varying the average data size and task runtime.

[1]WorkflowSim is available in Github: http://www.github.com/WorkflowSim

Workflows Used

- LIGO Inspiral analysis (400 tasks), and Epigenomics (500 tasks).
- Both workflows are generated and varied using the WorkflowGenerator[1].
- Runtime (average and task runtime distribution) and overhead (workflow engine delay, queue delay, and network bandwidth) information were collected from real traces production environments , then used as input parameters for the simulations.



The LIGO Inspiral Workflow
The Epigenomics Workflow

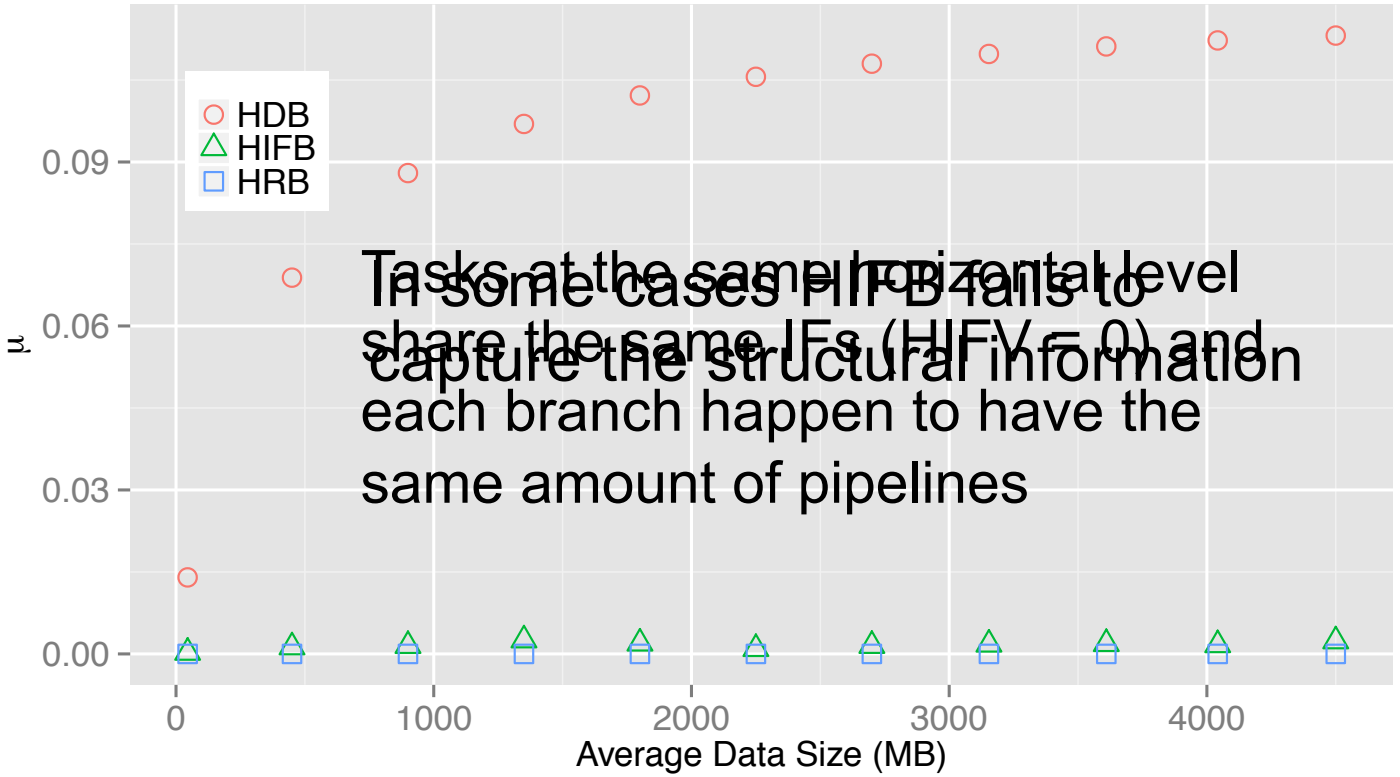
[1]WorkflowGenerator: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

Experiment I (LIGO)

- Performance gain (μ) over Horizontal Clustering
- The original average data size (both input and output data) of the LIGO workflow is about 5MB

Experiment I (Epigenomics)

- Performance gain (μ) over HC
- The original average data size (both input and output data) of the Epigenomics workflows is about 45MB.

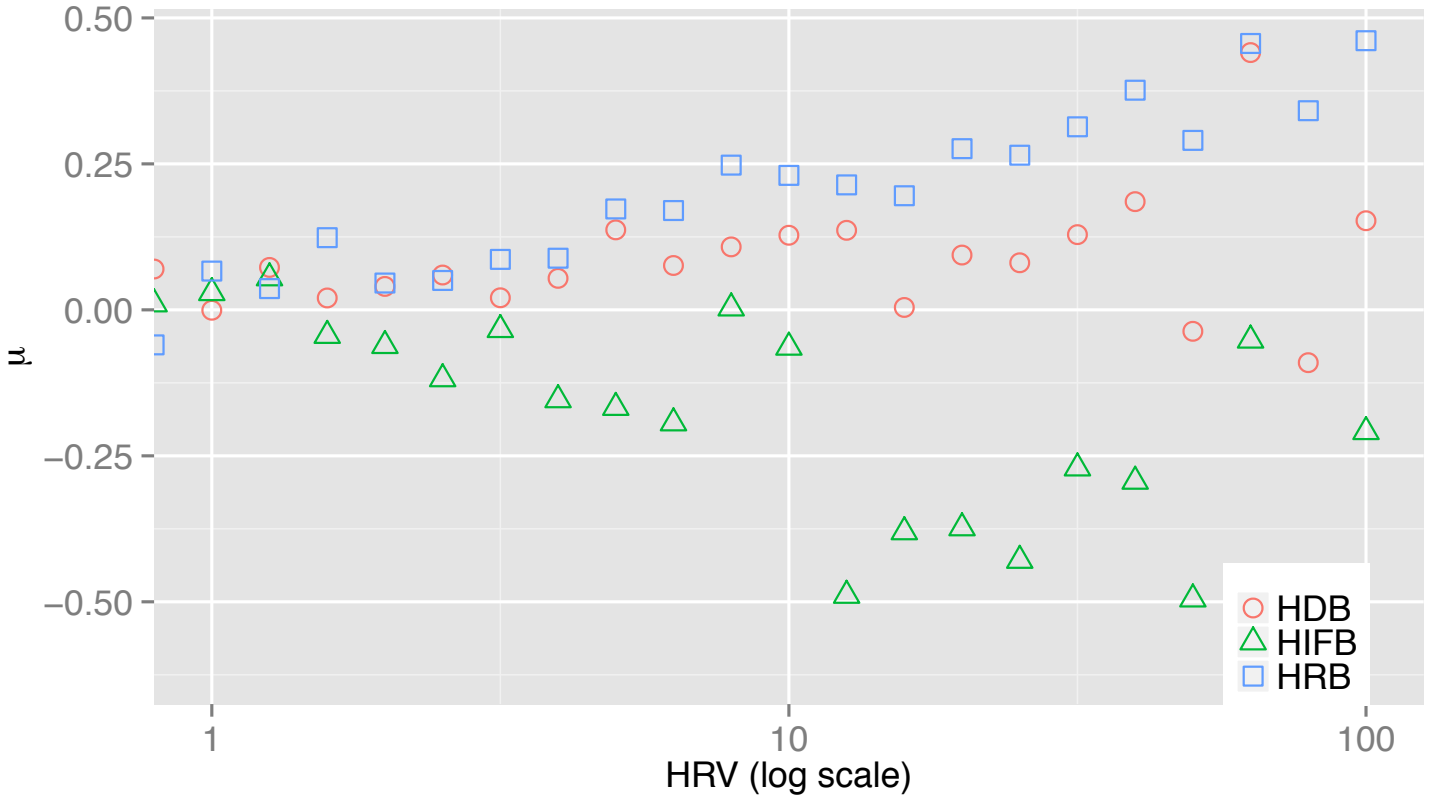


In some cases HIFB fails to capture the structural information

Tasks at the same horizontal level share the same IFs (HIFV = 0) and each branch happen to have the same amount of pipelines

Experiment II

- Varying HRV (LIGO)



For big HRV, we use HRB. Otherwise, HDB or HIFB.

Conclusion

- Initial attempt to identify multiple causes of imbalance in task clustering synthetically and quantitatively
- Proposed three metrics to measure both dependency imbalance and runtime imbalance and then three balancing methods based on these metrics respectively.
- Evaluated the three methods with real workflow traces in simulation: Distance metric > Impact Factor metric > Runtime Metric with the increase of data
- In the future,
  - Reveal the relationship between imbalance metrics and balancing methods
  - Vertical task clustering
  - Workflow characterization and analysis
  - Other optimization techniques

Q&A

Thank you!

- Main page: http://pegasus.isi.edu
- WorkflowSim is an open source project: http://www.github.com/WorkflowSim
- WorkflowGenerator is available at: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator
- Email: wchen@isi.edu
- Acknowledgement: This work is supported by NFS. We thank FutureGrid and Amazon Education Grant for providing cloud resources