Clustering Parallel Applications to Enhance Message Logging Protocols

Esteban Meneses

Jaguar is the top 2 supercomputer in the world with 224,162 cores...
During 537 days (Aug-22-2008 to Feb-10-2010)
2.33 failures per day

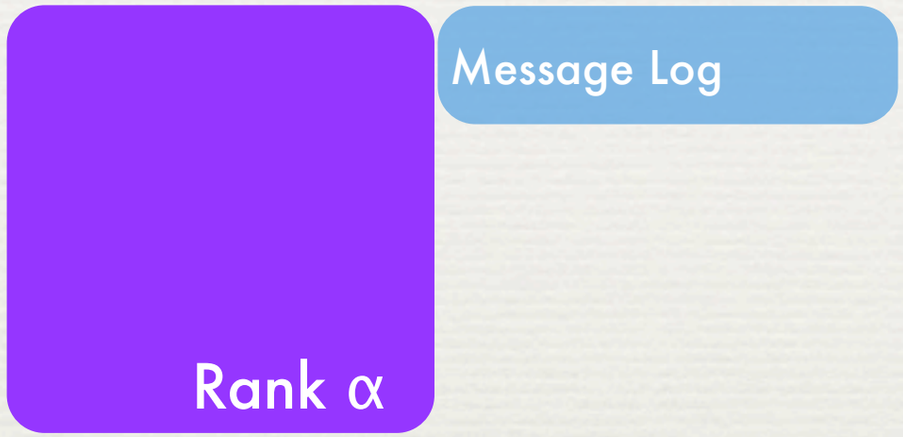
Sequoia will have 1.6 million cores and an exascale machine around 100 million cores...

We will see failures all the time

Agenda

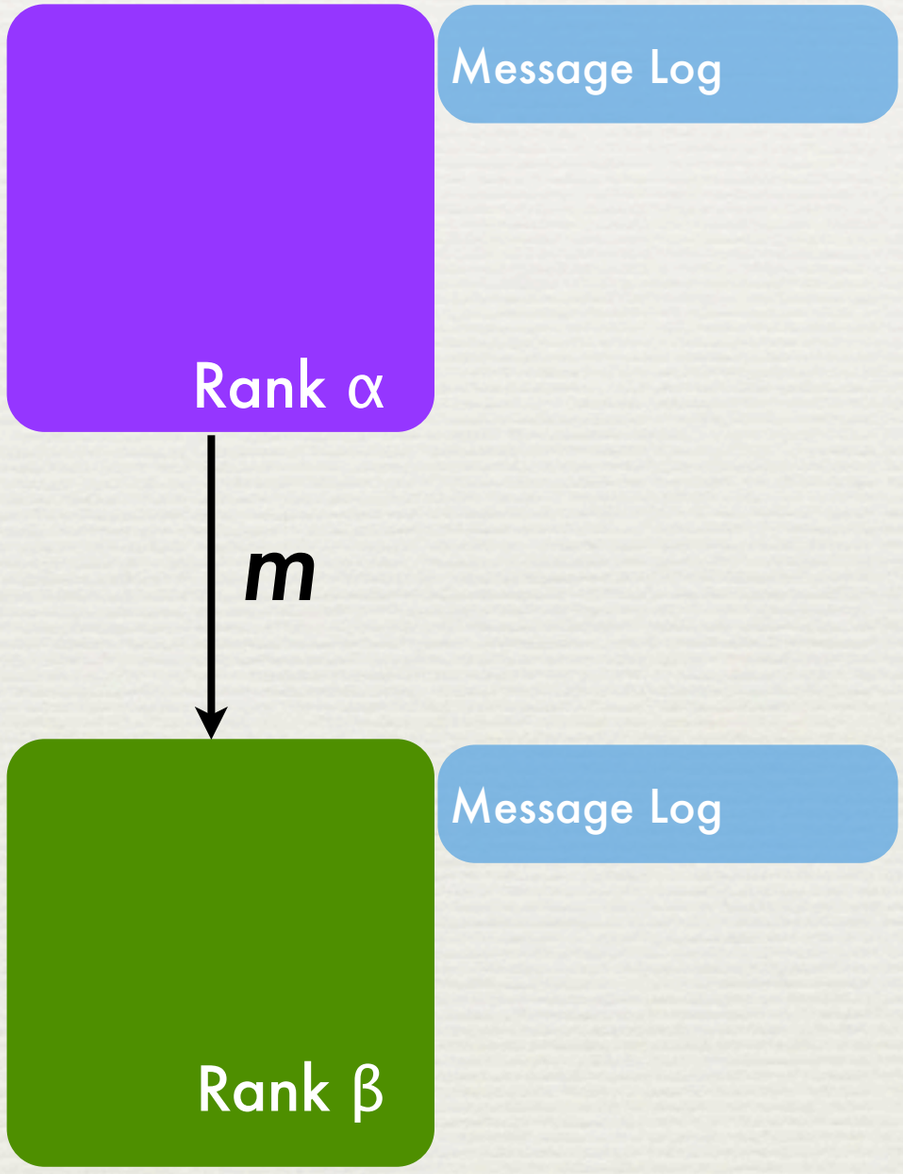
- Clusters and Message Logging.
- Static Clustering (MPI).
- Dynamic Clustering (Charm++).
- Future Work.

Message Logging



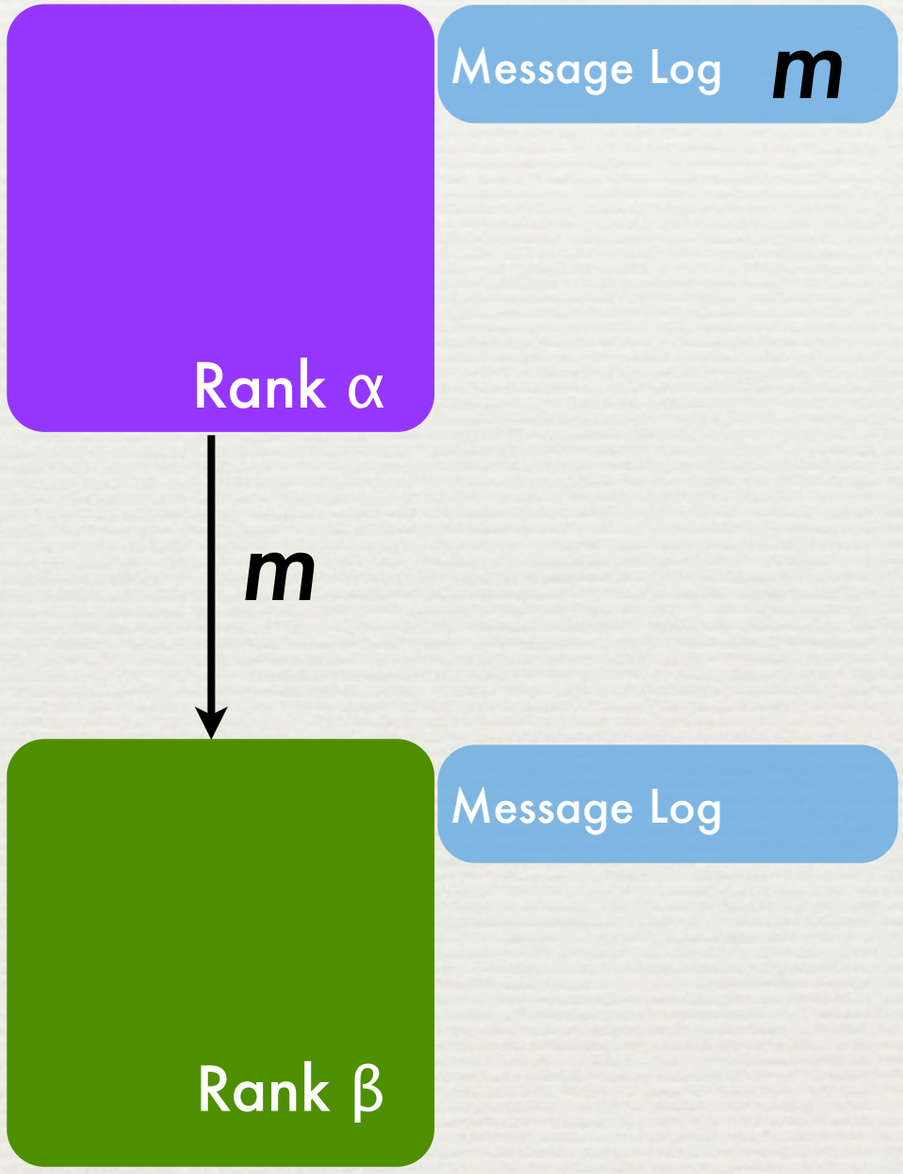
- Every message sent *may* be logged.
- **Advantage:** only the failed rank is rolled back.
- **Drawback**: memory overhead.

Message Logging



- Every message sent *may* be logged.
- **Advantage:** only the failed rank is rolled back.
- **Drawback**: memory overhead.

Message Logging



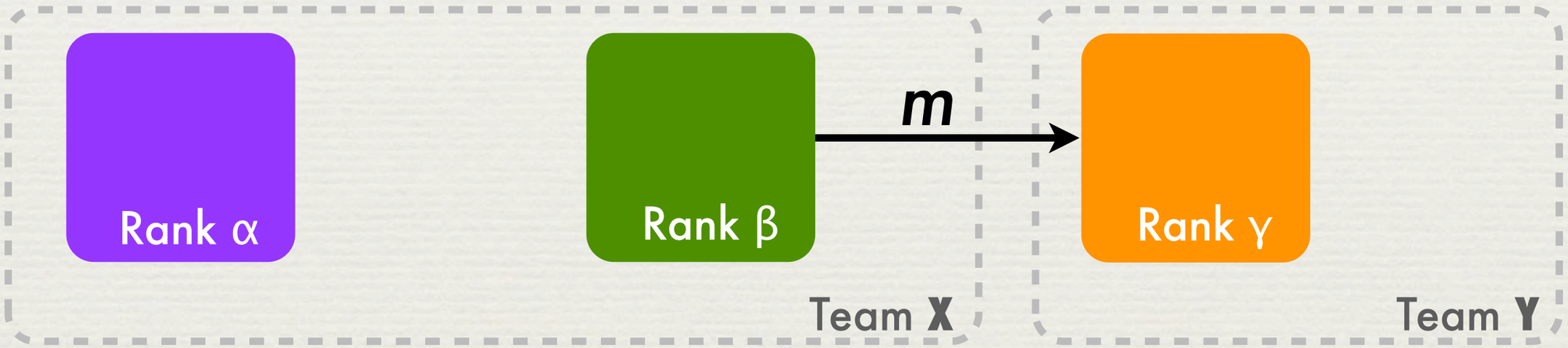
- Every message sent *may* be logged.
- **Advantage:** only the failed rank is rolled back.
- **Drawback**: memory overhead.

Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.

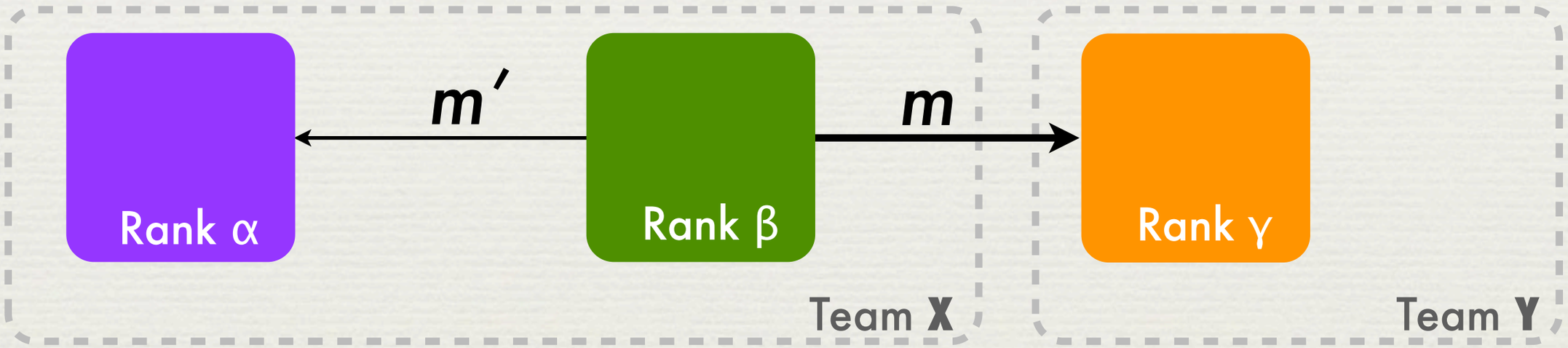
Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



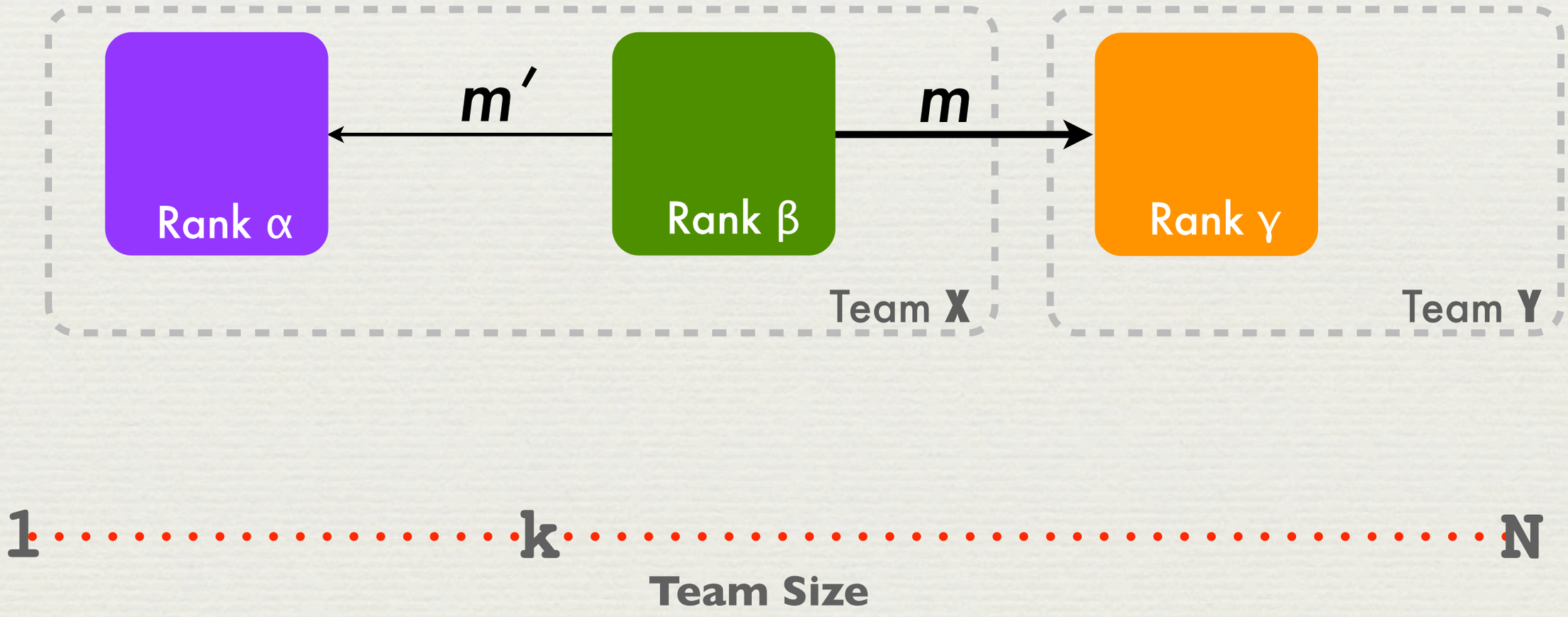
Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



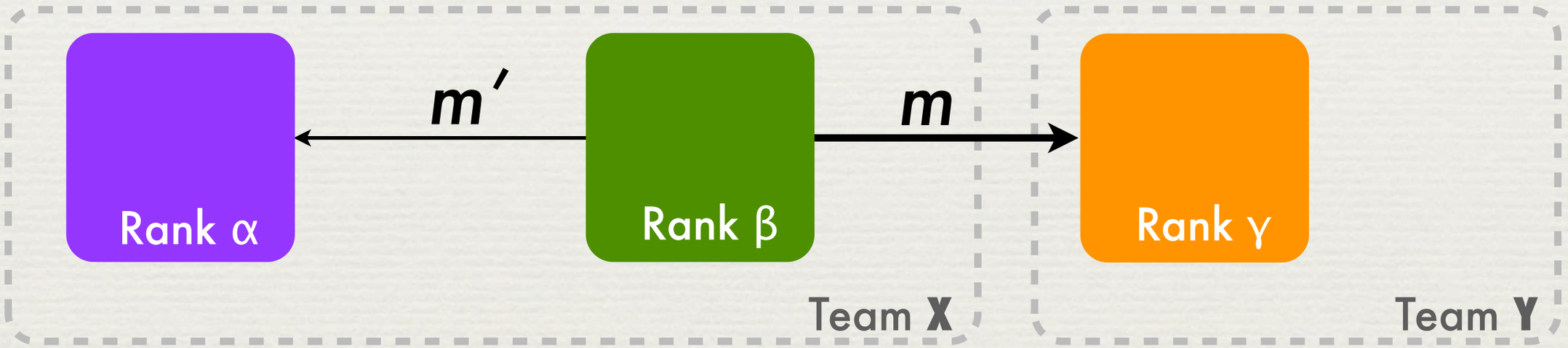
Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



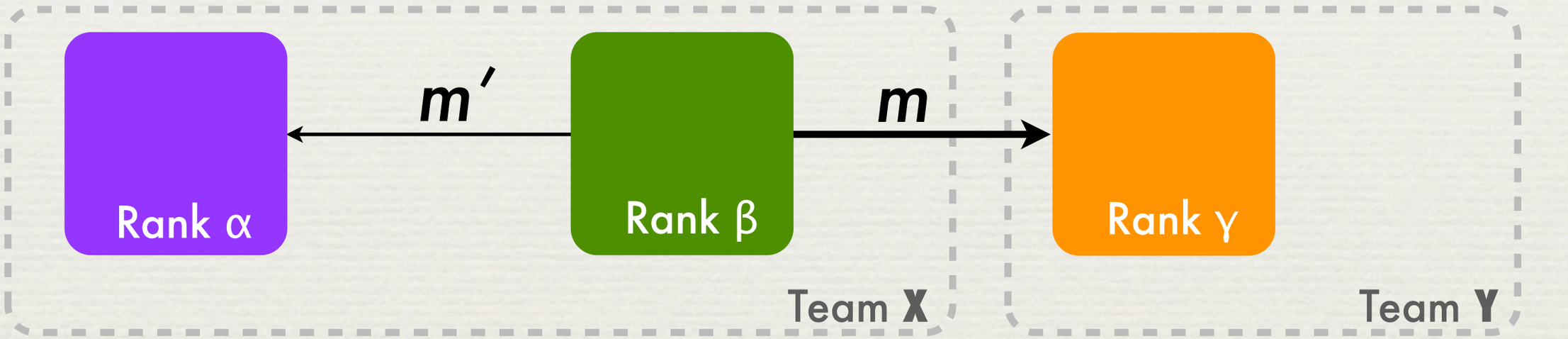
Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



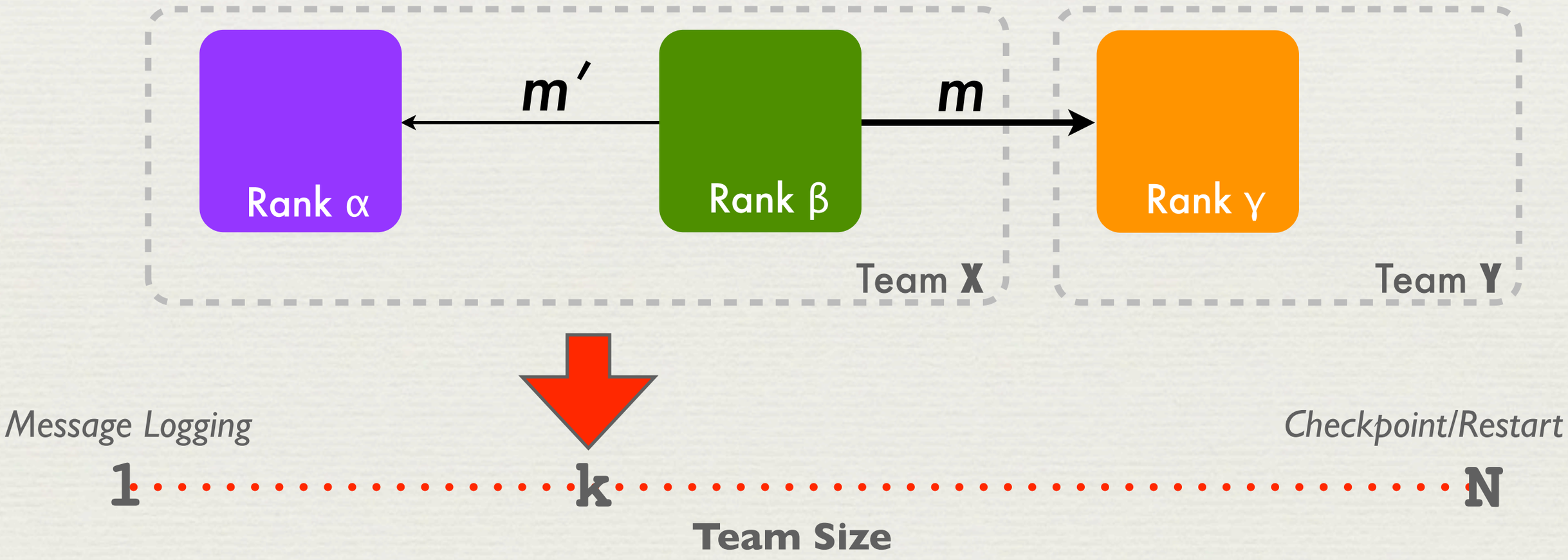
Message Logging — *Checkpoint/Restart*

1 . . . . . k . . . . . N

Team Size

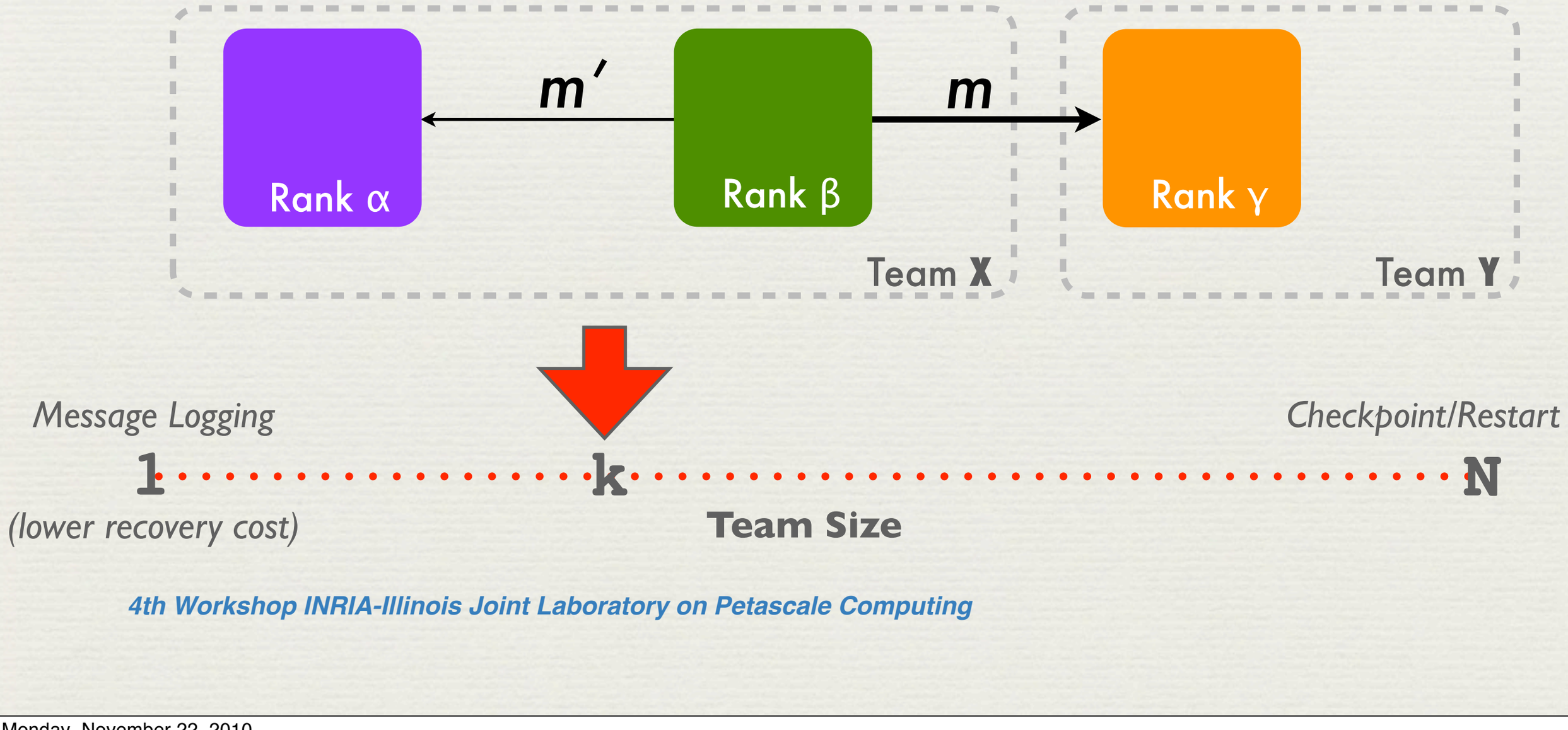
Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



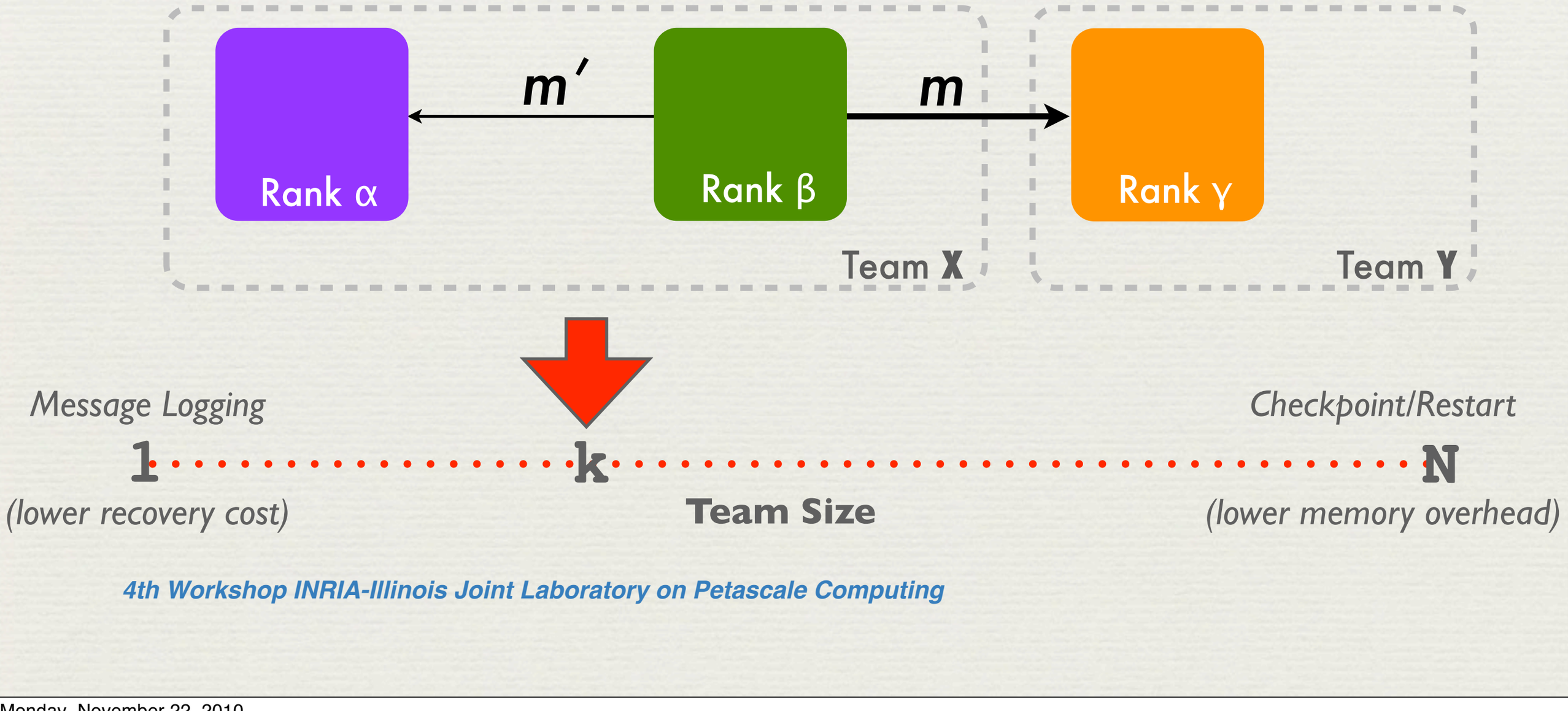
Teams

- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.

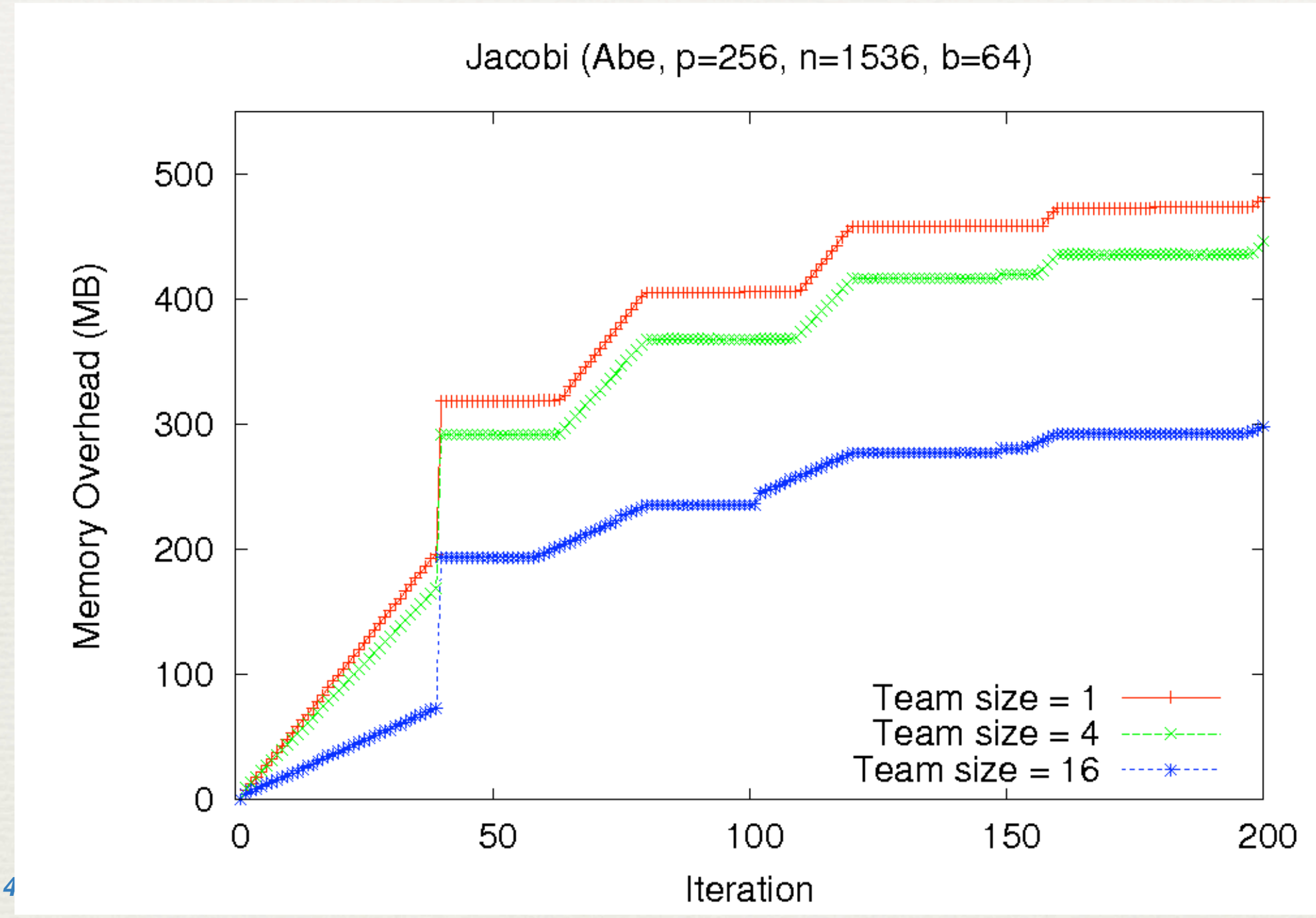


Teams

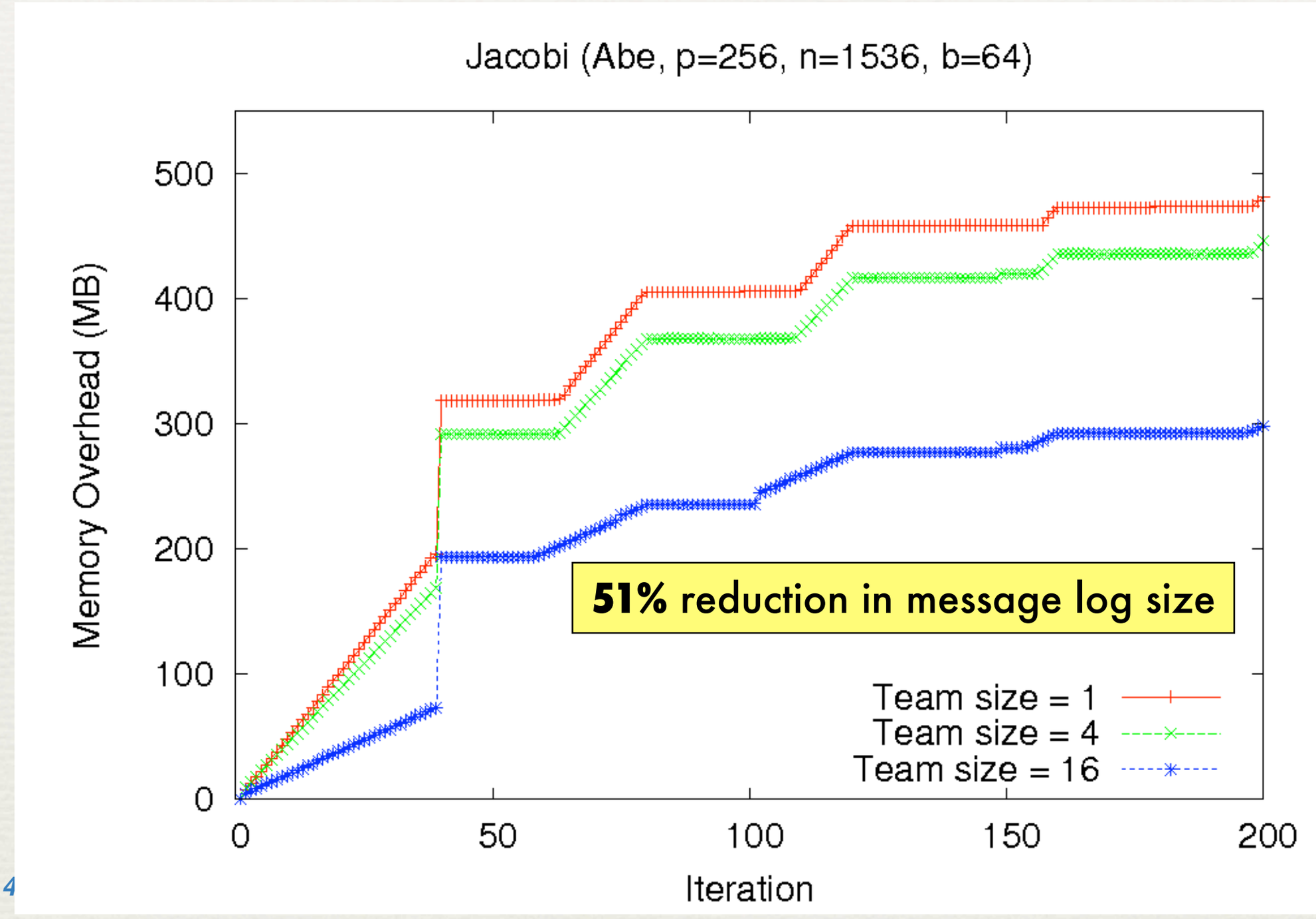
- **Goal**: reduce memory overhead of message log.
- Only messages crossing team **boundaries** are logged.



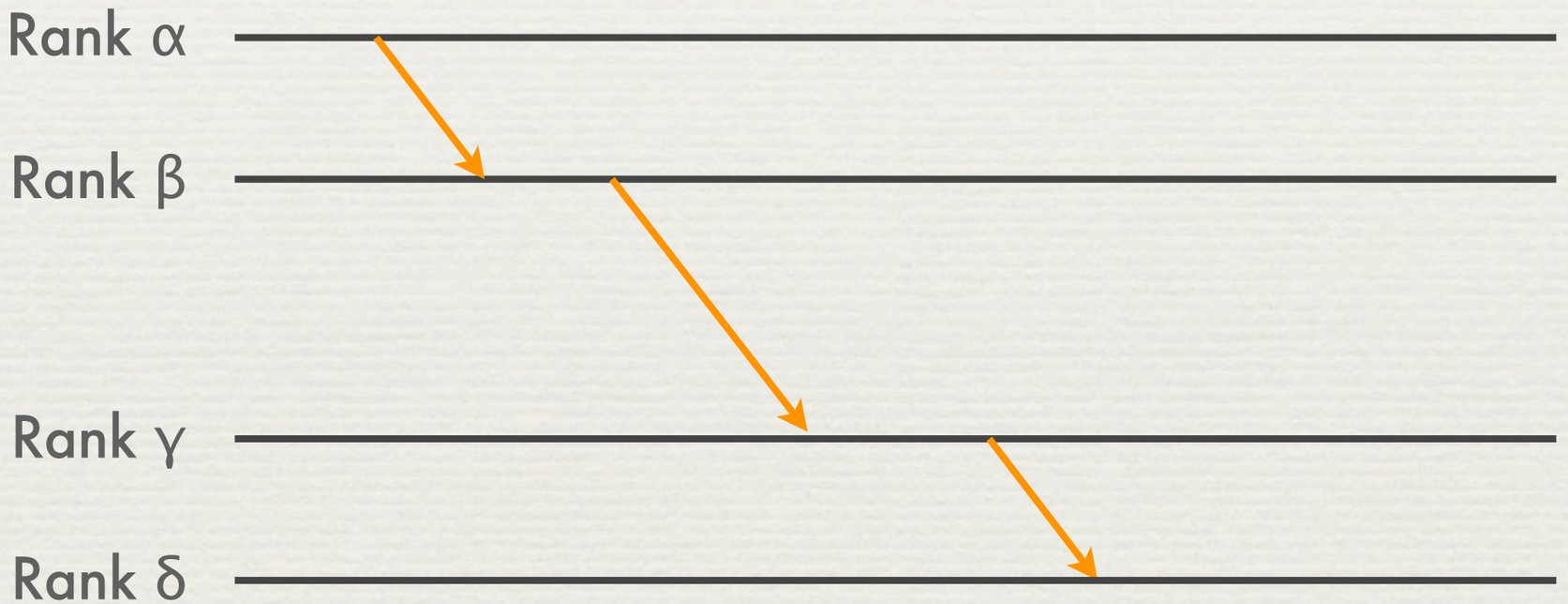
Reduce Memory Overhead



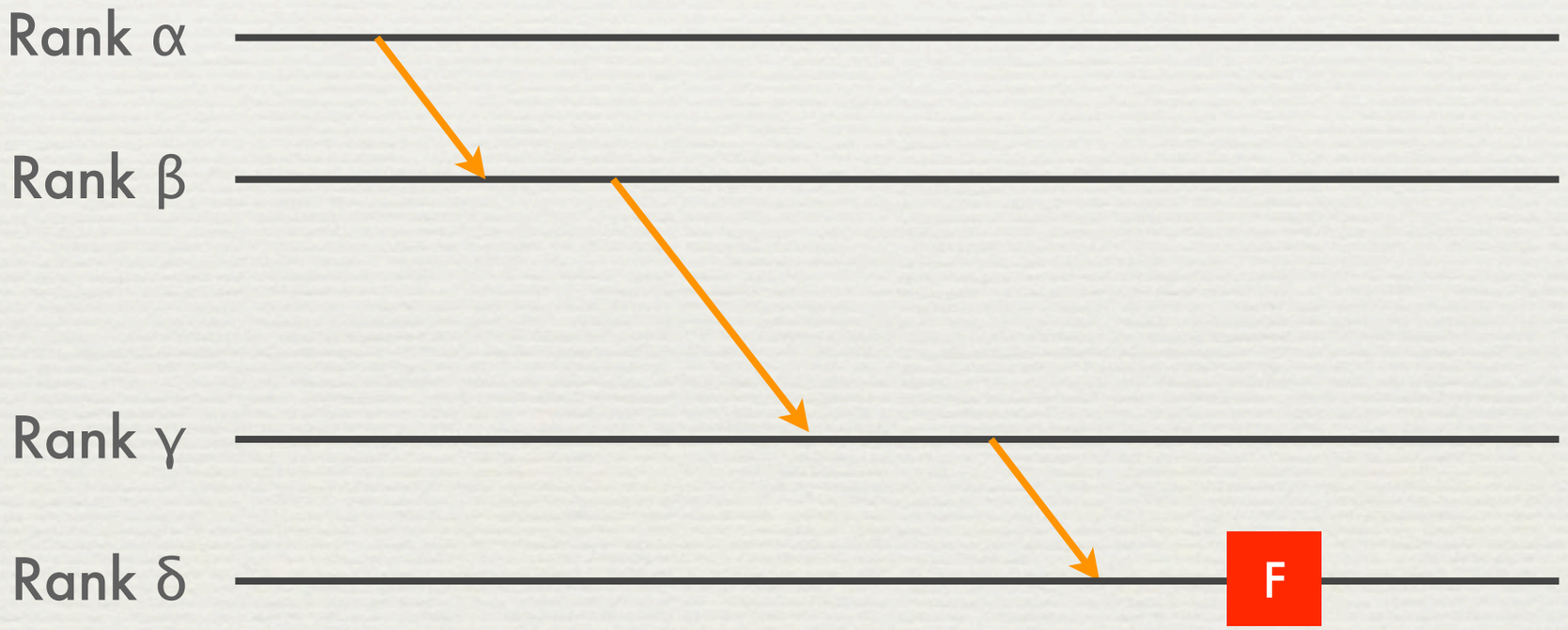
Reduce Memory Overhead



Bound Cascading Rollback

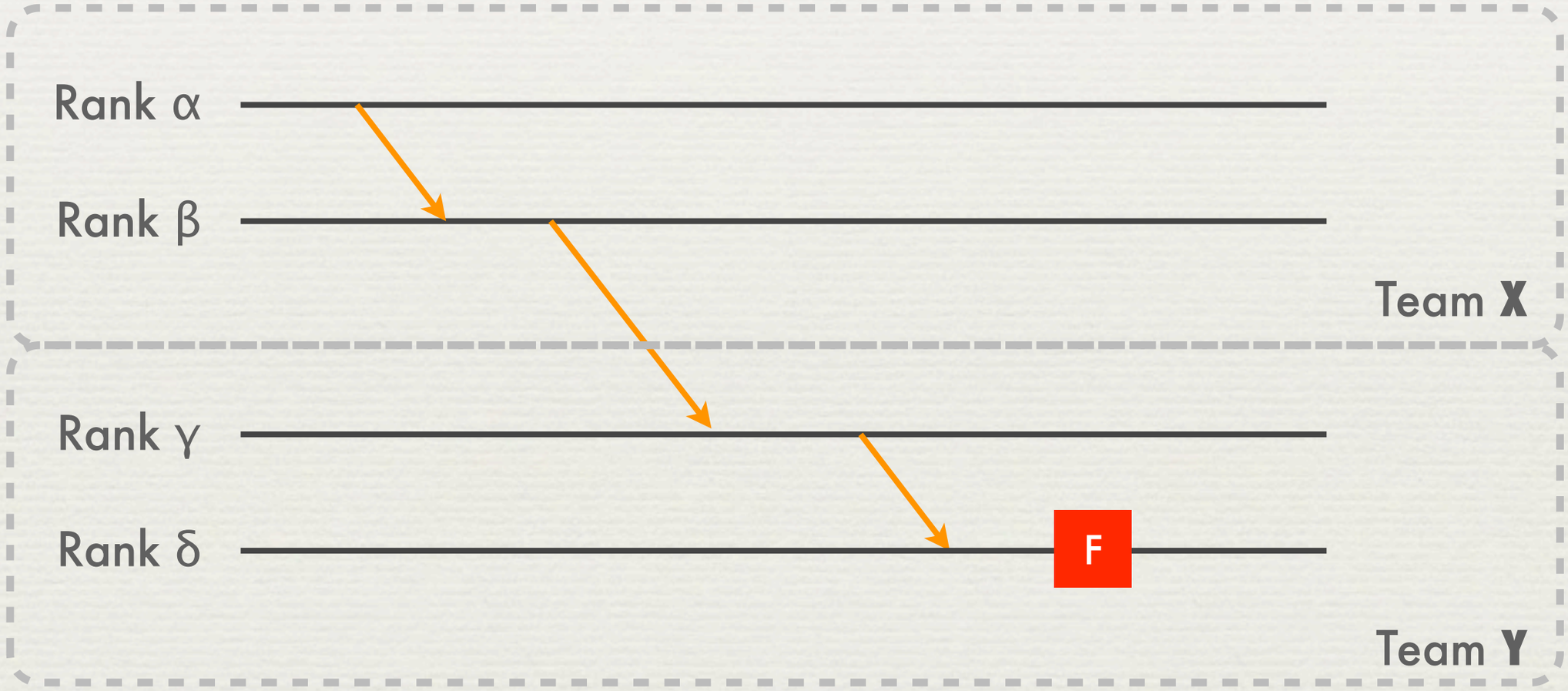


Bound Cascading Rollback



Bound Cascading Rollback

Bound Cascading Rollback



Bound Cascading Rollback

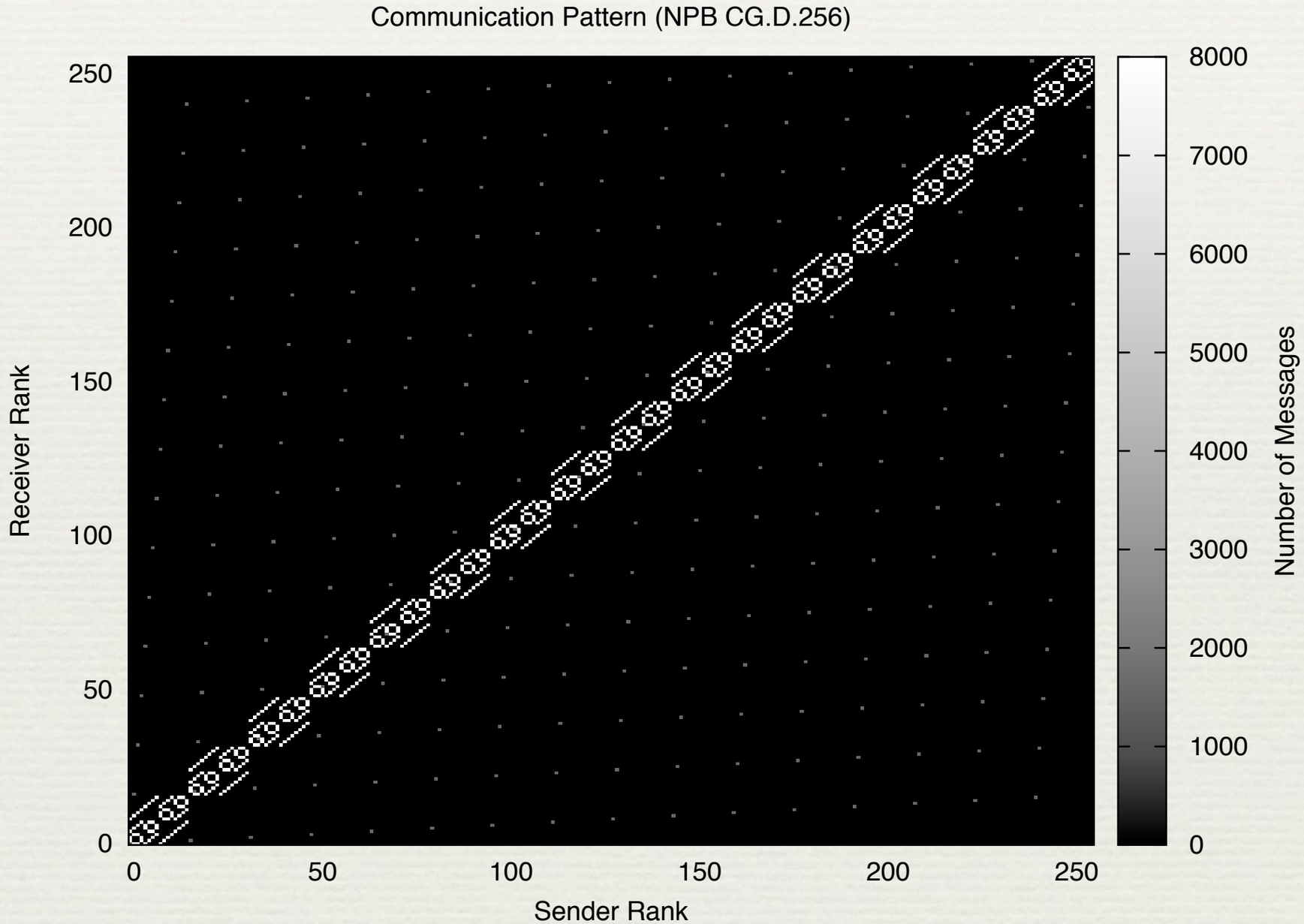
How to split the ranks to minimize the

communication volume?

Static Clustering

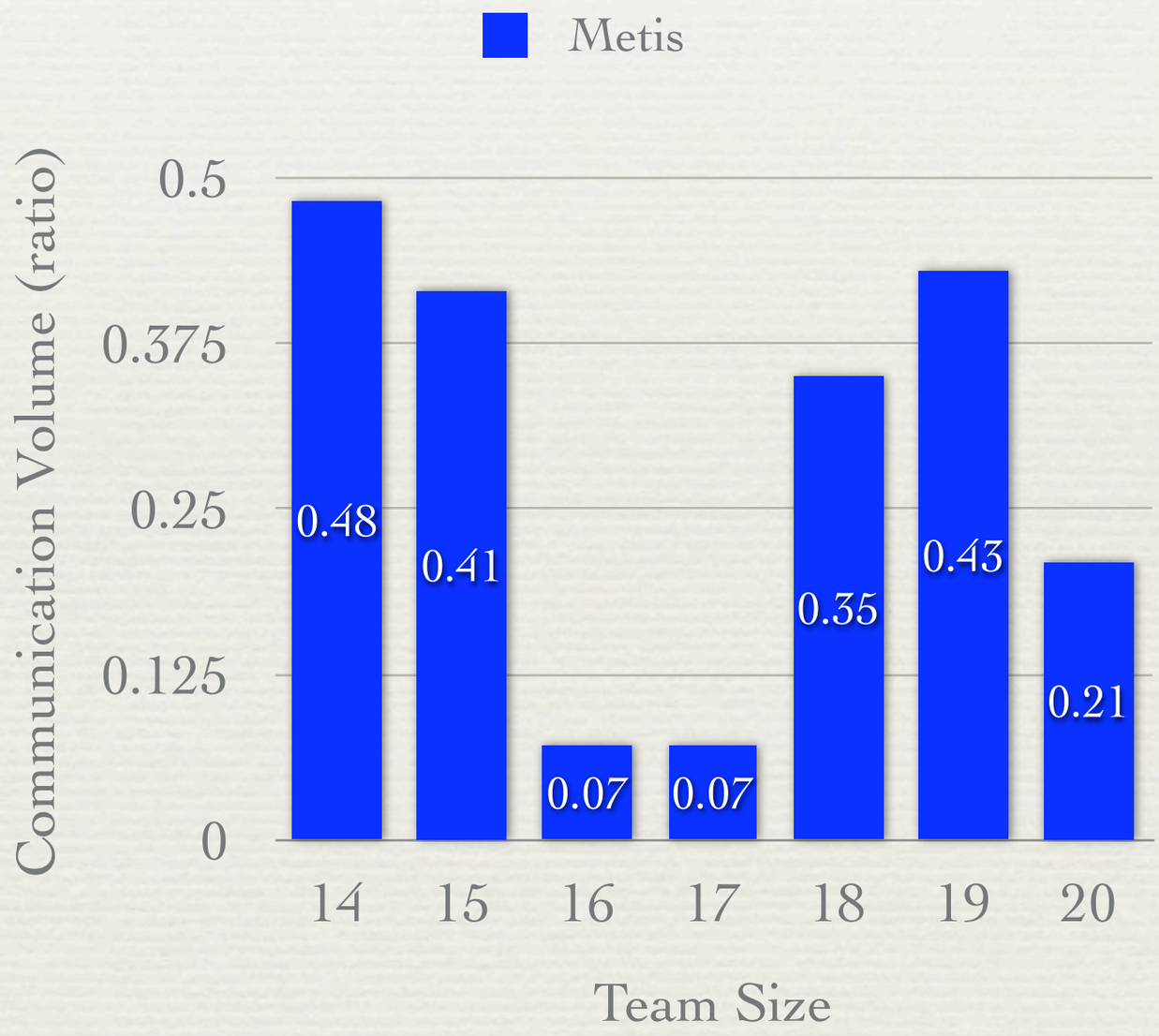
Amina Guermouche
Thomas Ropars
Prof. Franck Cappello
(INRIA)

Communication Pattern



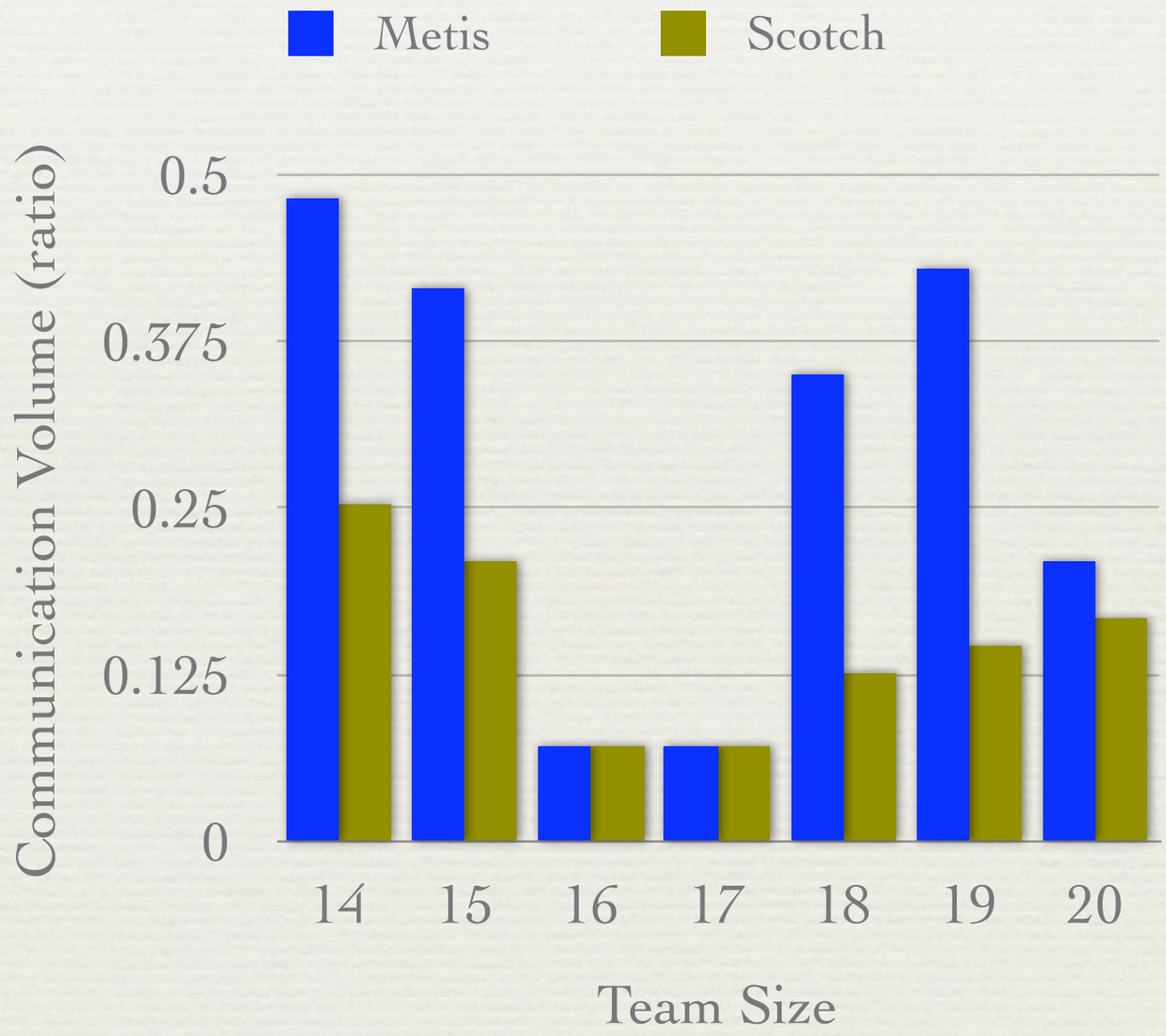
Team Size

- **Constraint**: maximum team size (t).
- **Graph partitioning** techniques with k clusters: k=⌈N/t⌉.
- **Example**: t=20.



Team Size

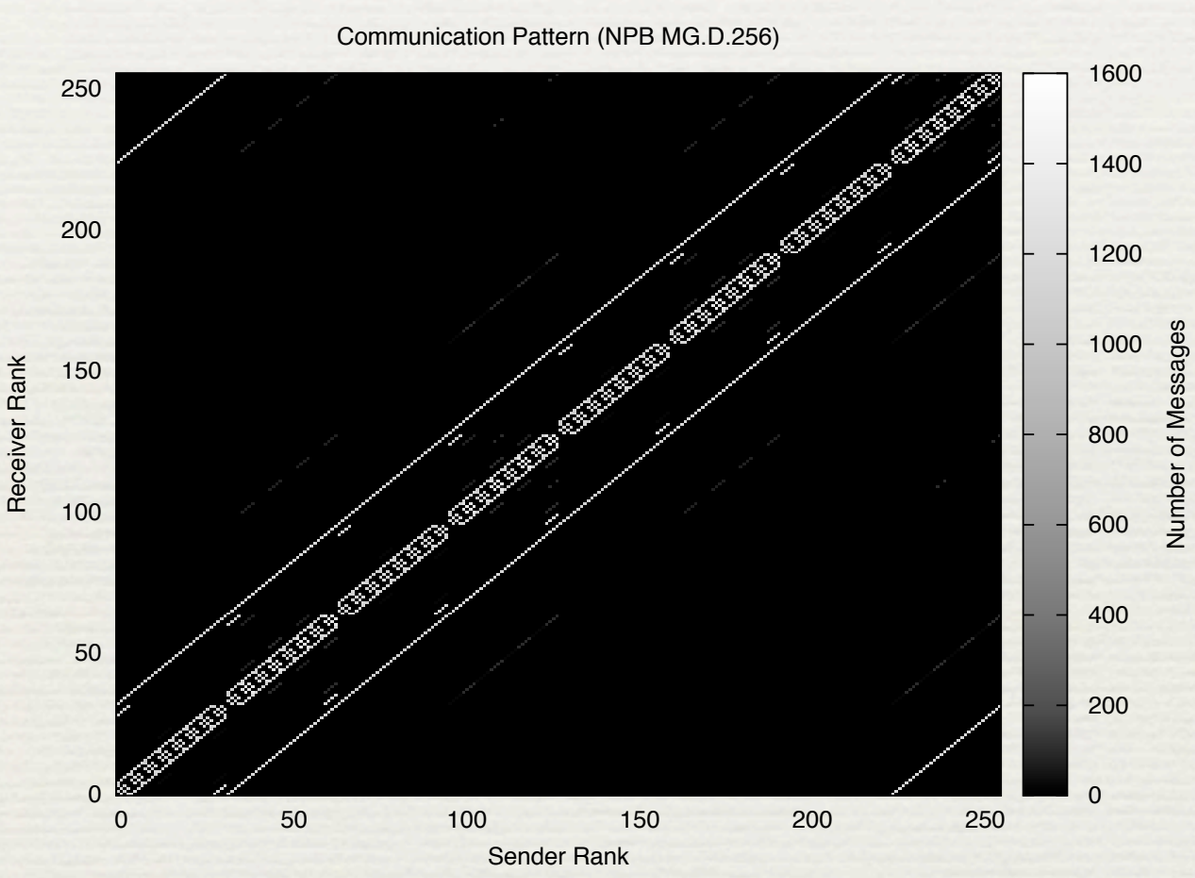
- **Constraint**: maximum team size (t).
- **Graph partitioning** techniques with k clusters: k=⌈N/t⌉.
- **Example**: t=20.



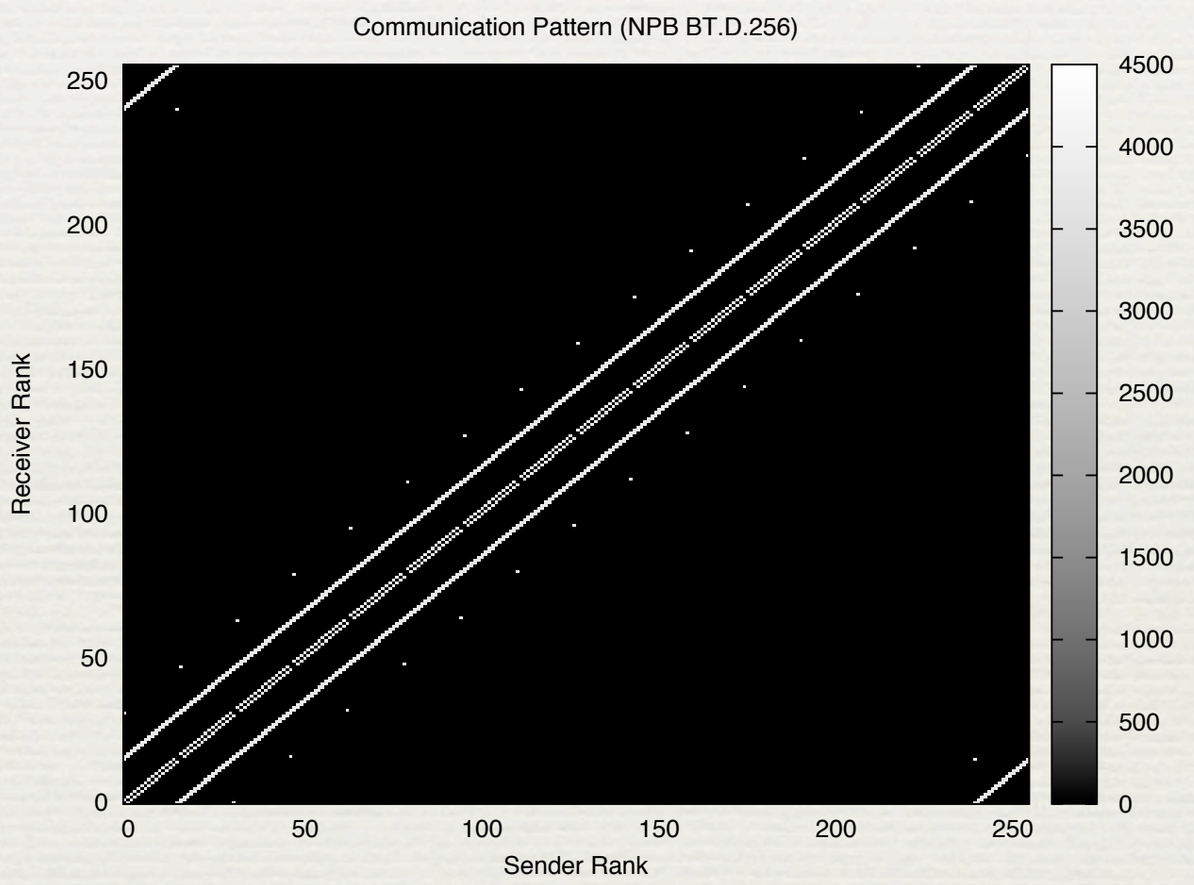
Team Size

- **Constraint**: maximum team size (t).
- **Graph partitioning** techniques with k clusters: k=⌈N/t⌉.
- **Example**: t=20.

Benchmarks



NPB-MG



NPB-BT

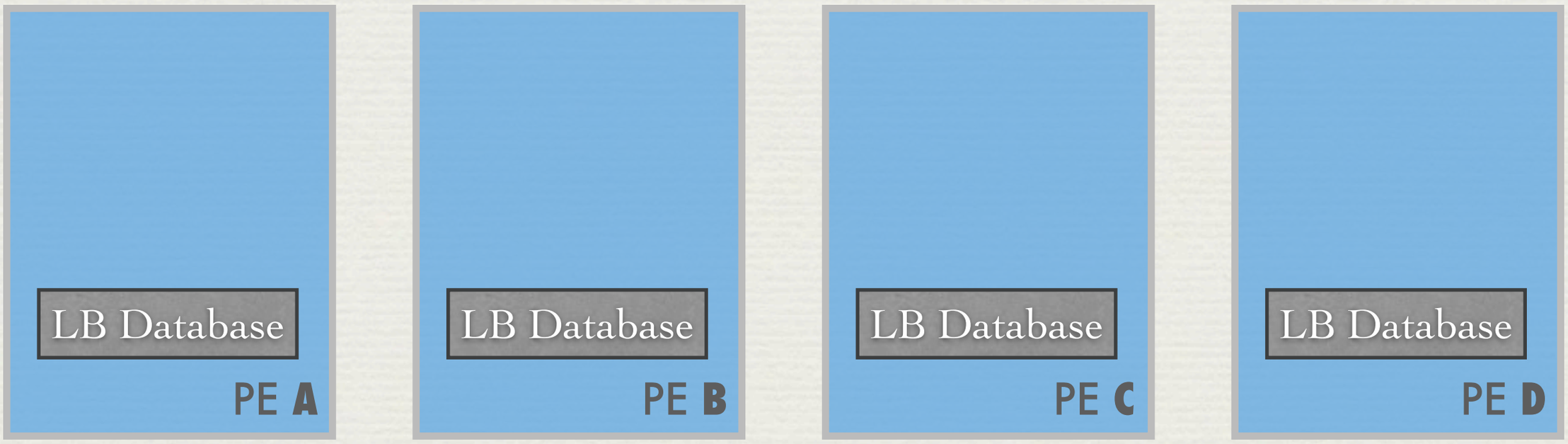
Graph Properties

Program	Average Path Length	Clustering Coefficient	Communication Volume (ratio)		
			Metis	Scotch	Random
NPB-CG (t=16)	4.49	0	0.07	0.07	0.93
NPB-MG (t=32)	3.82	0.09	0.27	-	0.87
NPB-BT (t=16)	6.24	0.40	0.35	0.33	0.93

Dynamic Clustering

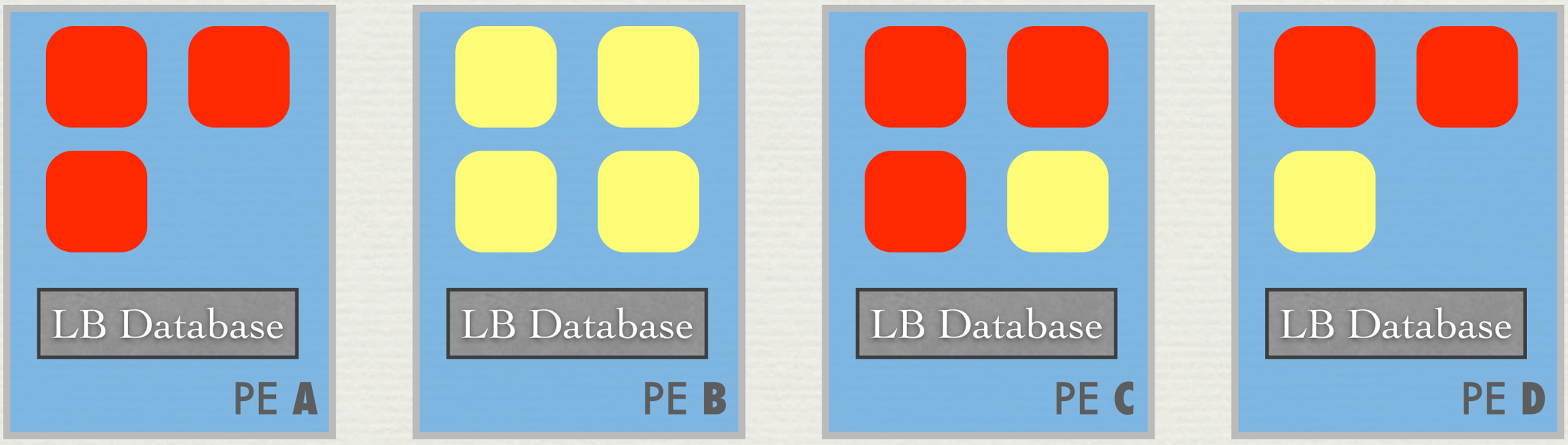
Load Balancing in Charm++

- **Migratable** objects, asynchronous method invocation.
- **Measurement-based** load balancing: collects computation load and communication structure.



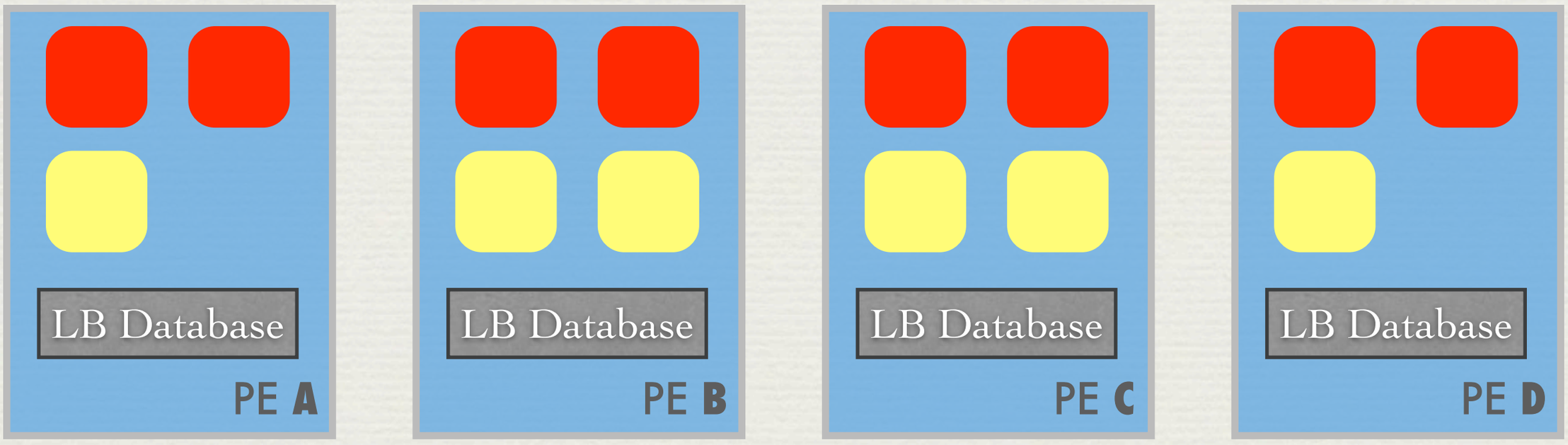
Load Balancing in Charm++

- **Migratable** objects, asynchronous method invocation.
- **Measurement-based** load balancing: collects computation load and communication structure.



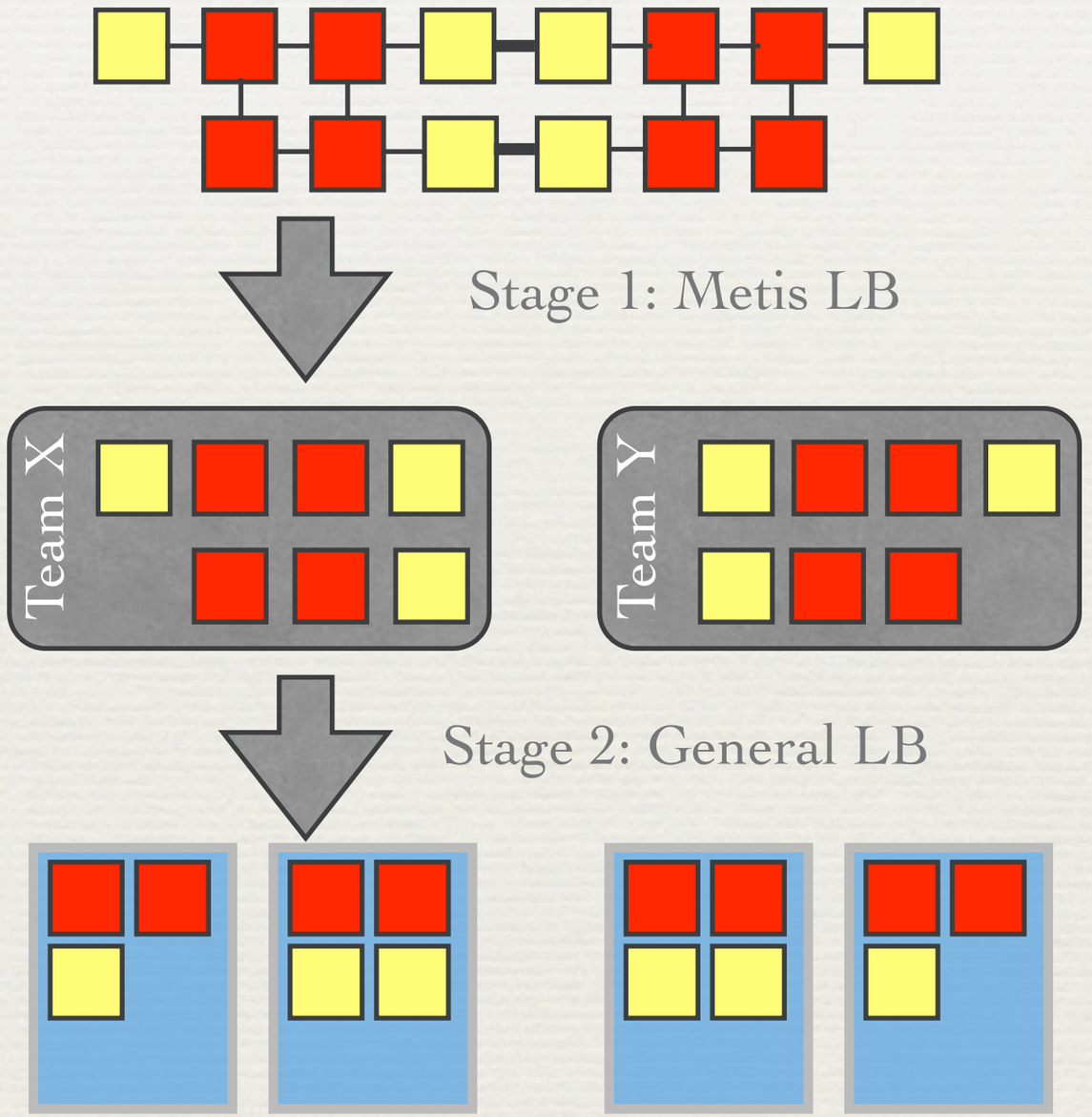
Load Balancing in Charm++

- **Migratable** objects, asynchronous method invocation.
- **Measurement-based** load balancing: collects computation load and communication structure.

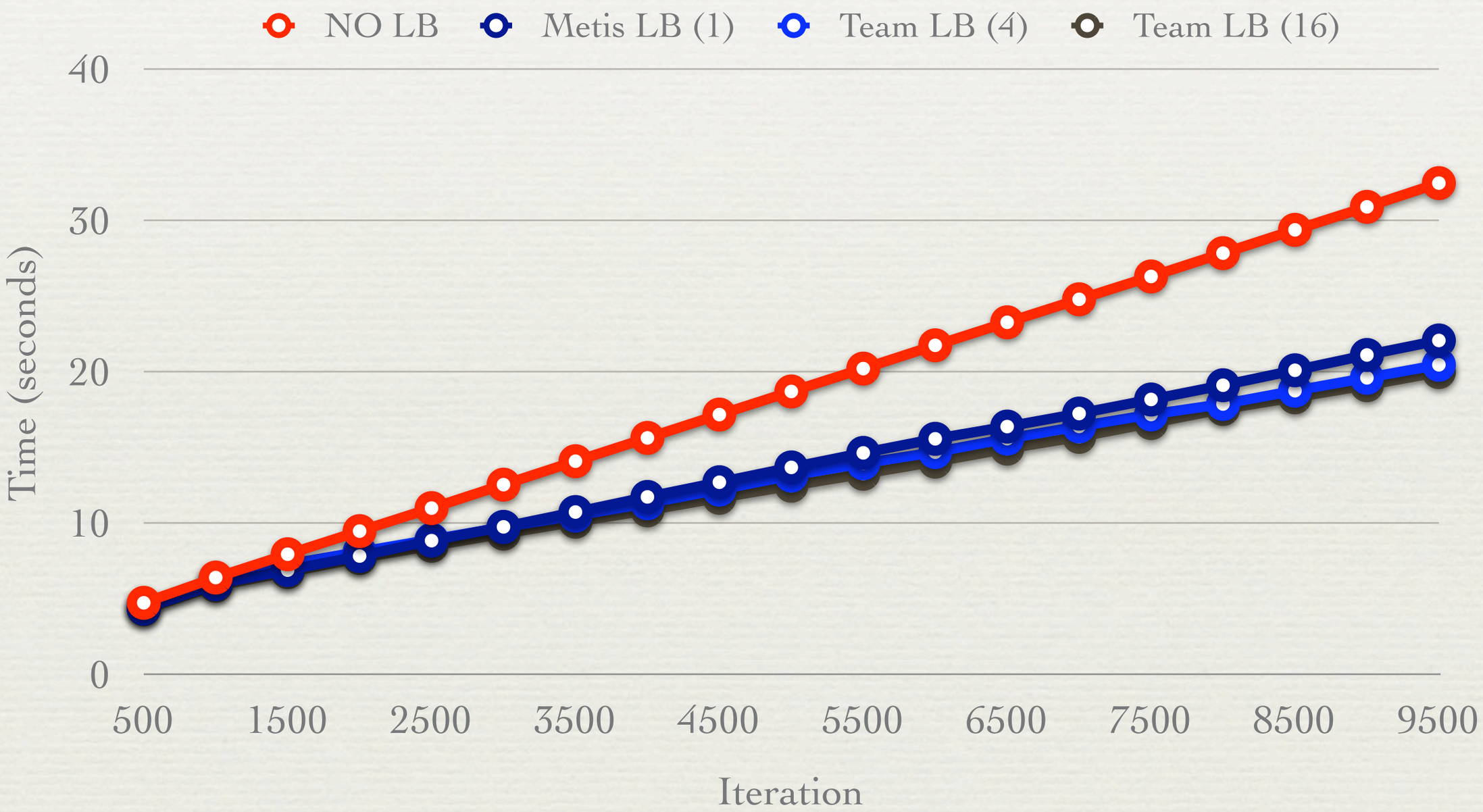


Team Load Balancer

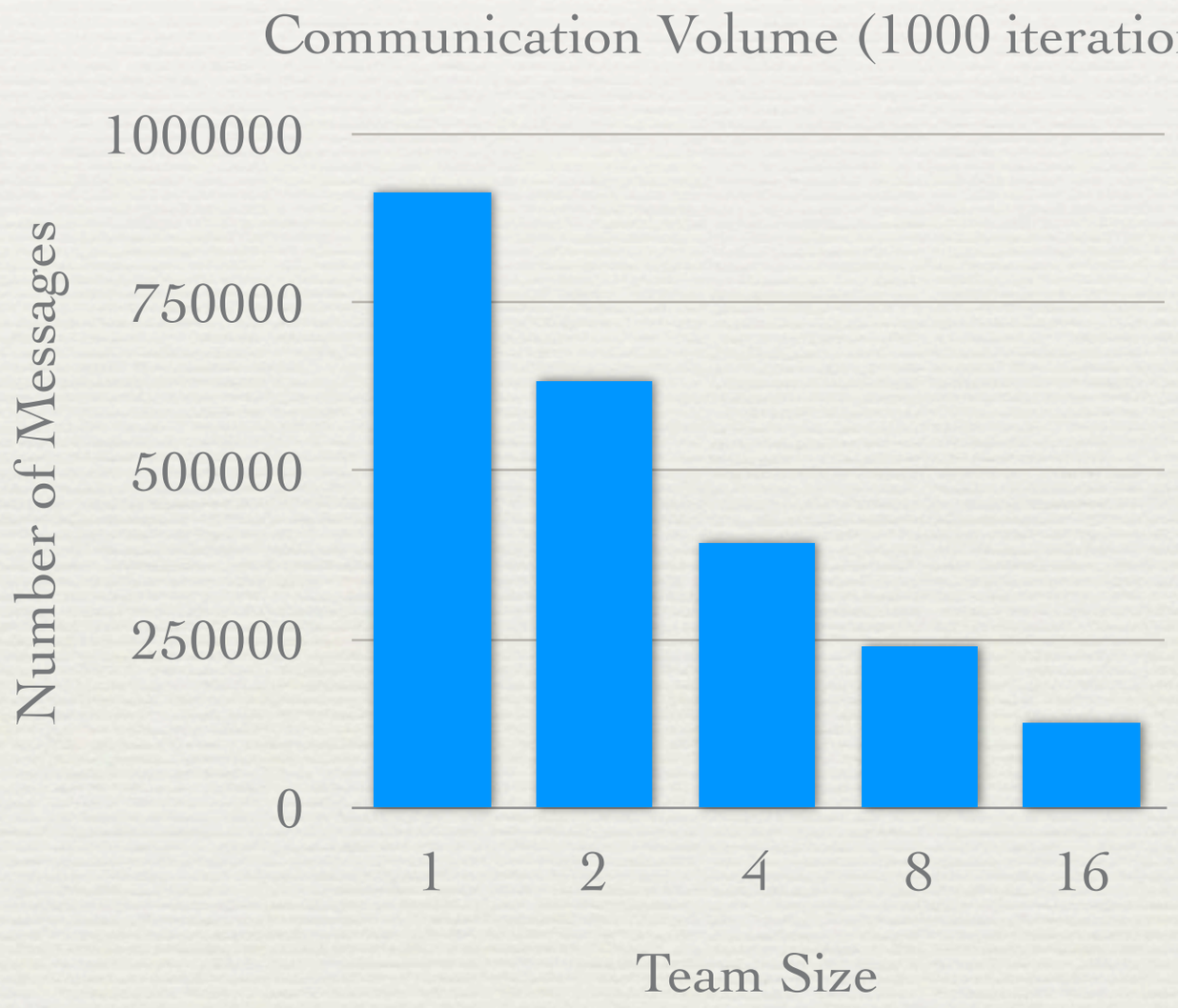
- Divides (evenly) the objects into teams while minimizing **communication volume**.
- Team LB (*t*), *t* is the team size (number of PEs).
- **Two stage** process:
  - Divide objects into teams.
  - Load balance each team.



Reducing Execution Time



Reducing Message Log Size



Conclusions

- Graph partitioning techniques are a promising alternative to cluster parallel applications.
- Message logging protocols benefit from team partitioning:
  - Reduce message log size.
  - Avoid cascading rollback.

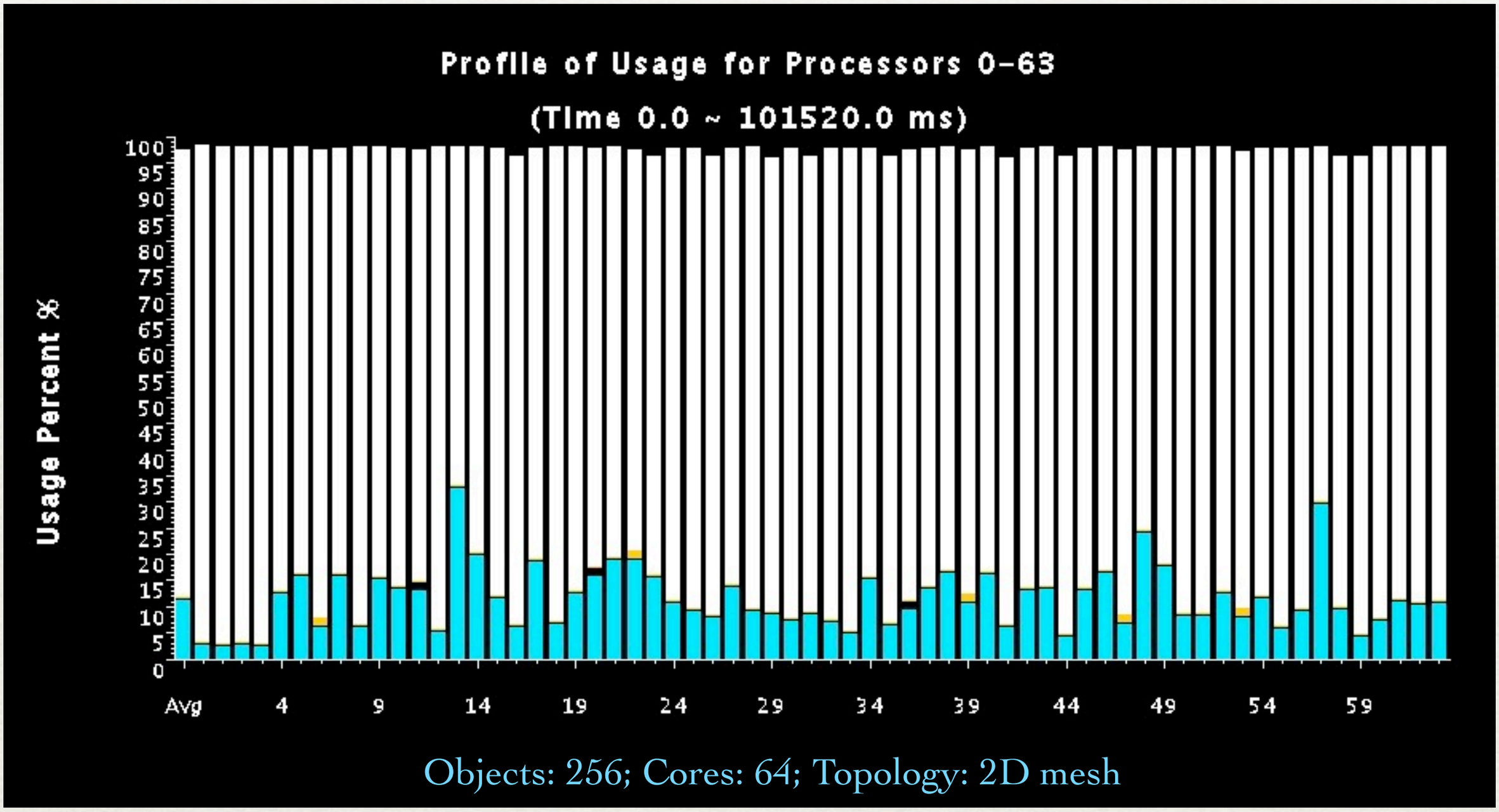
Future Work

- Scalable tool to collect communication information in MPI (collectives, notion of time).
- Evaluate more applications to inspect their clustering properties.
- Integration of clustering algorithms into parallel frameworks.

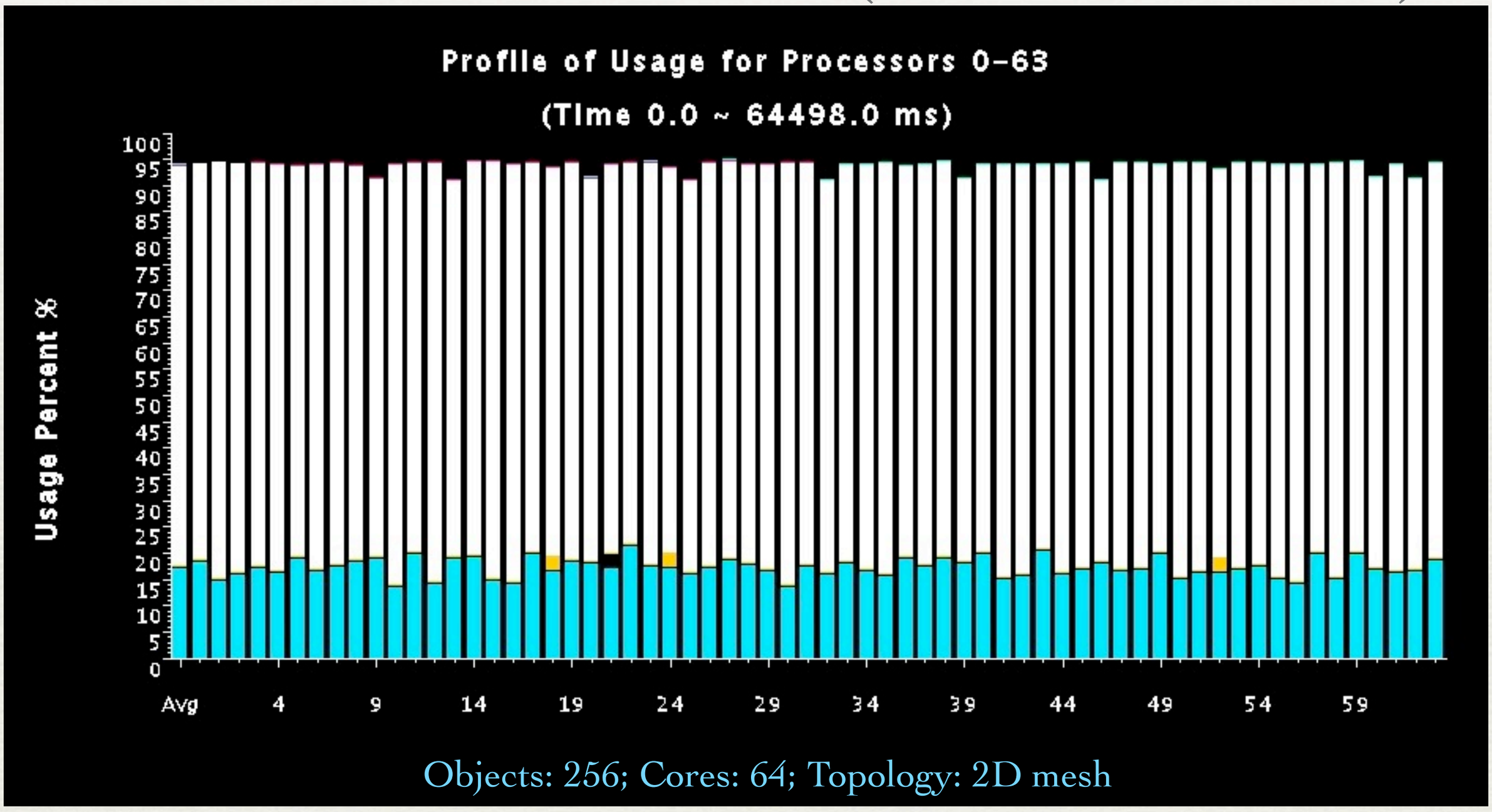
Q&A

Thank you!

LB Test


Profile of Usage for Processors 0-63
(Time 0.0 ~ 101520.0 ms)
Usage Percent %
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Avg
4
9
14
19
24
29
34
39
44
49
54
59
Objects: 256; Cores: 64; Topology: 2D mesh

Load Balance (Metis LB)



Load Balance (Team LB)