2024/08/06
SS研HPCフォーラム2024

HPCシステムのインタラクティブ化技術
大規模並列アプリの即時実行・効率化

富士通株式会社
コンピューティング研究所
大辻　弘貴

*1 Some figures and texts are based on the SC23 Research Poster:
"Scalable Fine-Grained Gang Scheduling for HPC Systems with Unreliable Broadcast Synchronization Mechanisms"

自己紹介

FUJITSU

- 大辻 弘貴 (Hiroki Ohtsuji, Ph.D.)
- 富士通株式会社 研究本部 コンピューティング研究所
  - Compute Workload Brokerコアプロジェクト
- 領域
  - 高性能並列ファイルシステム
  - ジョブスケジューラ、大規模クラスタ設計・構築・運用
    - 研究用1056ノードFX700クラスタを新規設計・構築・運用（22年度～）
- 活動
  - 情報処理学会（HPC研究会運営委員）
  - IEEE会員

はじめに

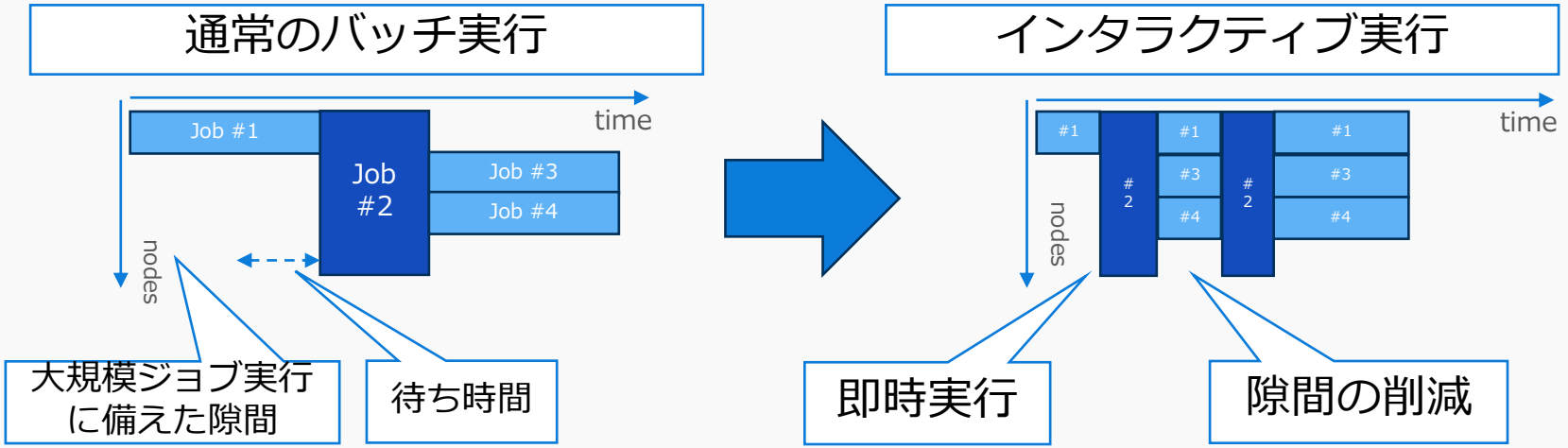
FUJITSU

- **HPCシステムにおいてインタラクティブ性の要求が高まっている**
  - 通常のHPCシステムはジョブ実行開始までの**待ち時間が不可避**
    - 原則として投入順に実行するバッチキューシステム由来の課題
  - AIをはじめとする新たなHPCユーザはバッチキューに慣れていない
    - Jupyter Notebookなど**インタラクティブなUI**を介した利用が主流
      - モデルが小さいうちはリソースを専有することができた
    - ところが、**モデルサイズの大規模化**などにより**並列計算が必須**になりつつある
      - 常にリソースを確保することは難しく、**HPCベースのシステム利用が増加**

大規模なジョブを効率的に即時実行する需要が生じている

HPCシステムのインタラクティブ化技術

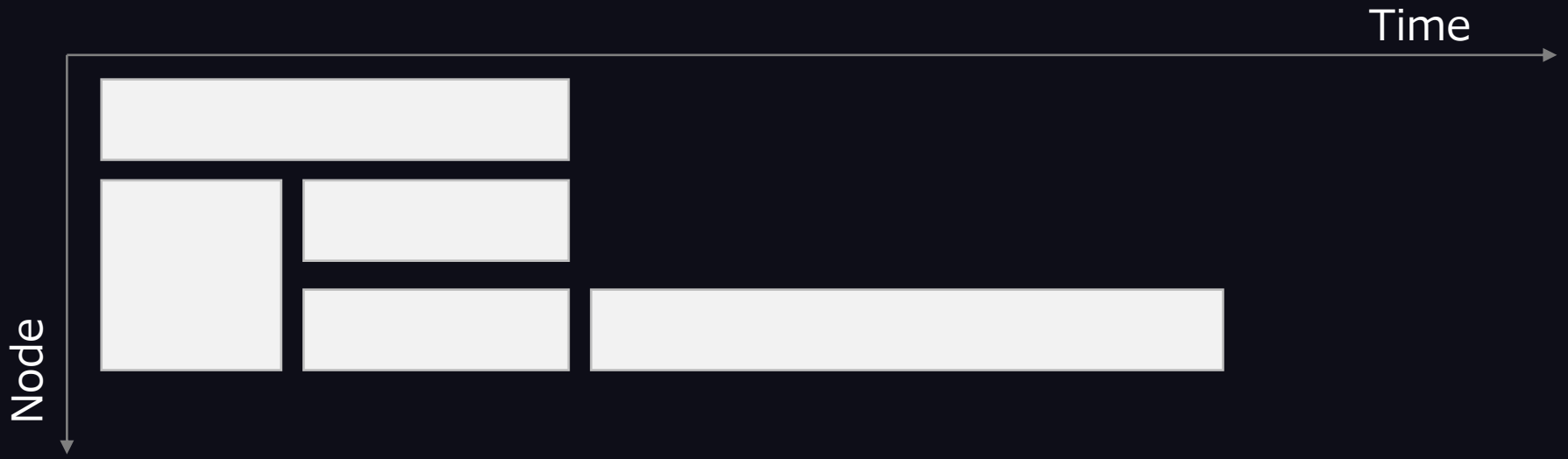
- 細粒度Gang schedulingによる並列アプリの同期並行実行
  - HPCアプリの時間分割実行を多ノード環境においても高効率に実現
    - ➔HPCシステムにおける高いインタラクティブ性の実現
      - ➔全系に近い大規模ジョブとそのほかのジョブを極めて少ない性能劣化で実現
    - ➔大規模ジョブにより生じる隙間を埋めることでシステム利用効率を向上



AI/ML Users and the Need for Responsive Job Execution

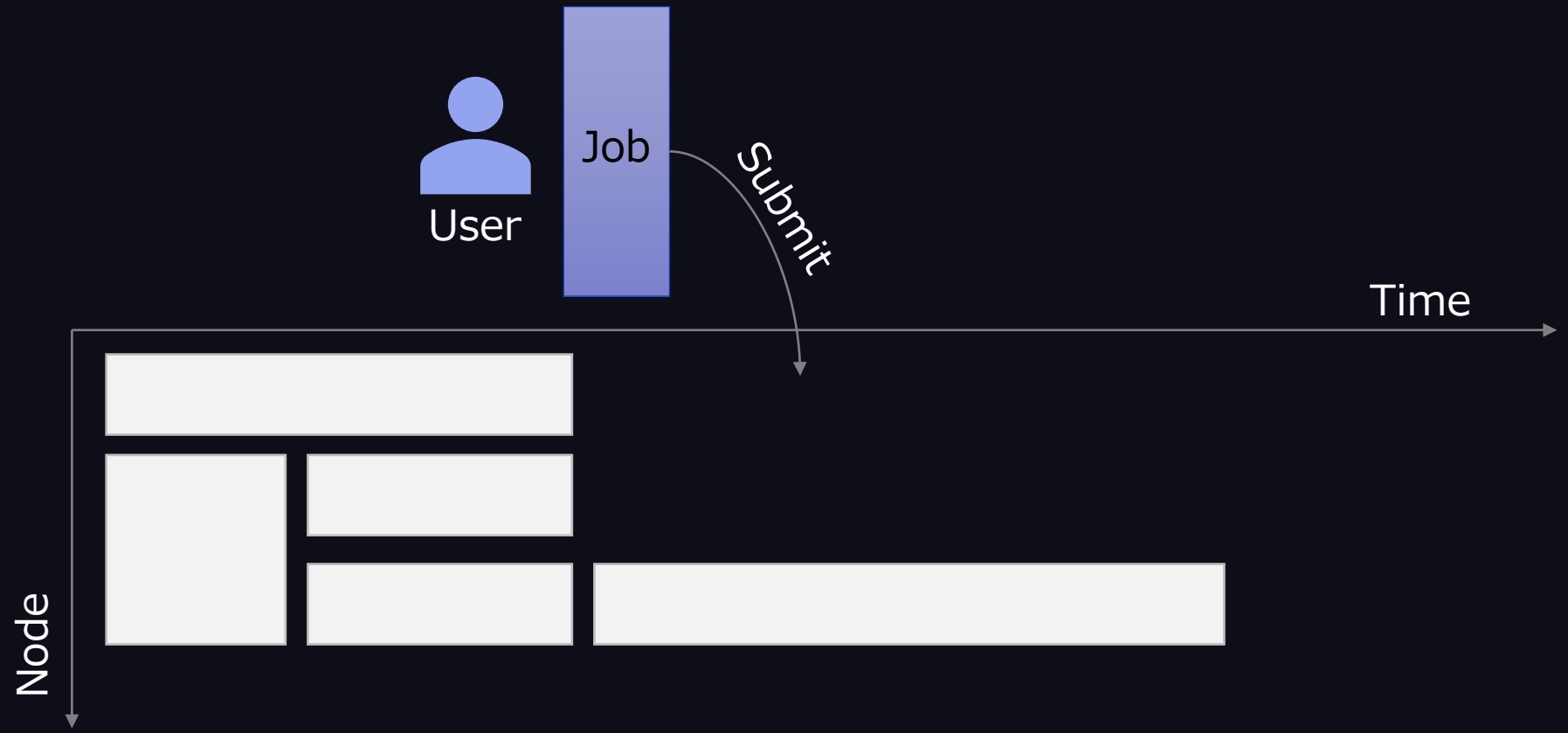
FUJITSU

- The demand for interactivity on HPC is increasing
  - Growing number of HPC users in the AI/ML fields



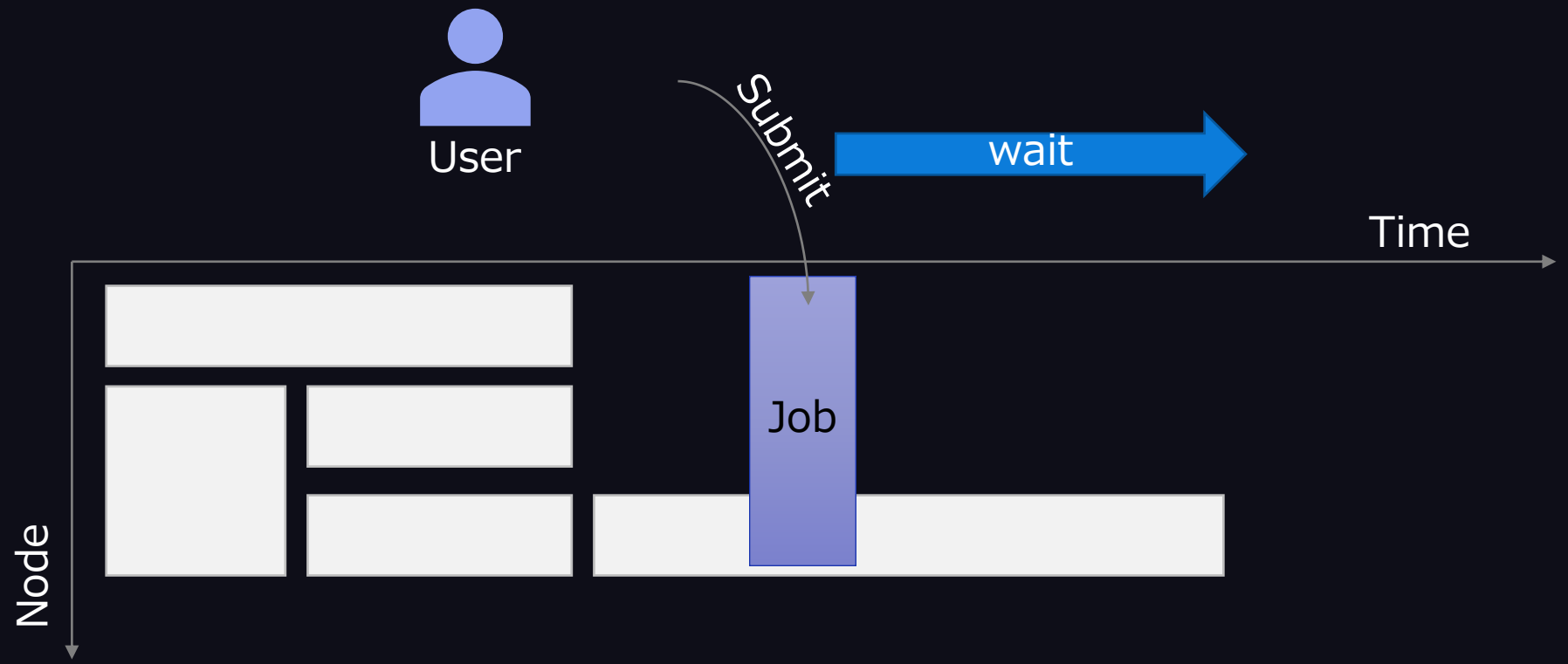
AI/ML Users and the Need for Responsive Job Execution

FUJITSU



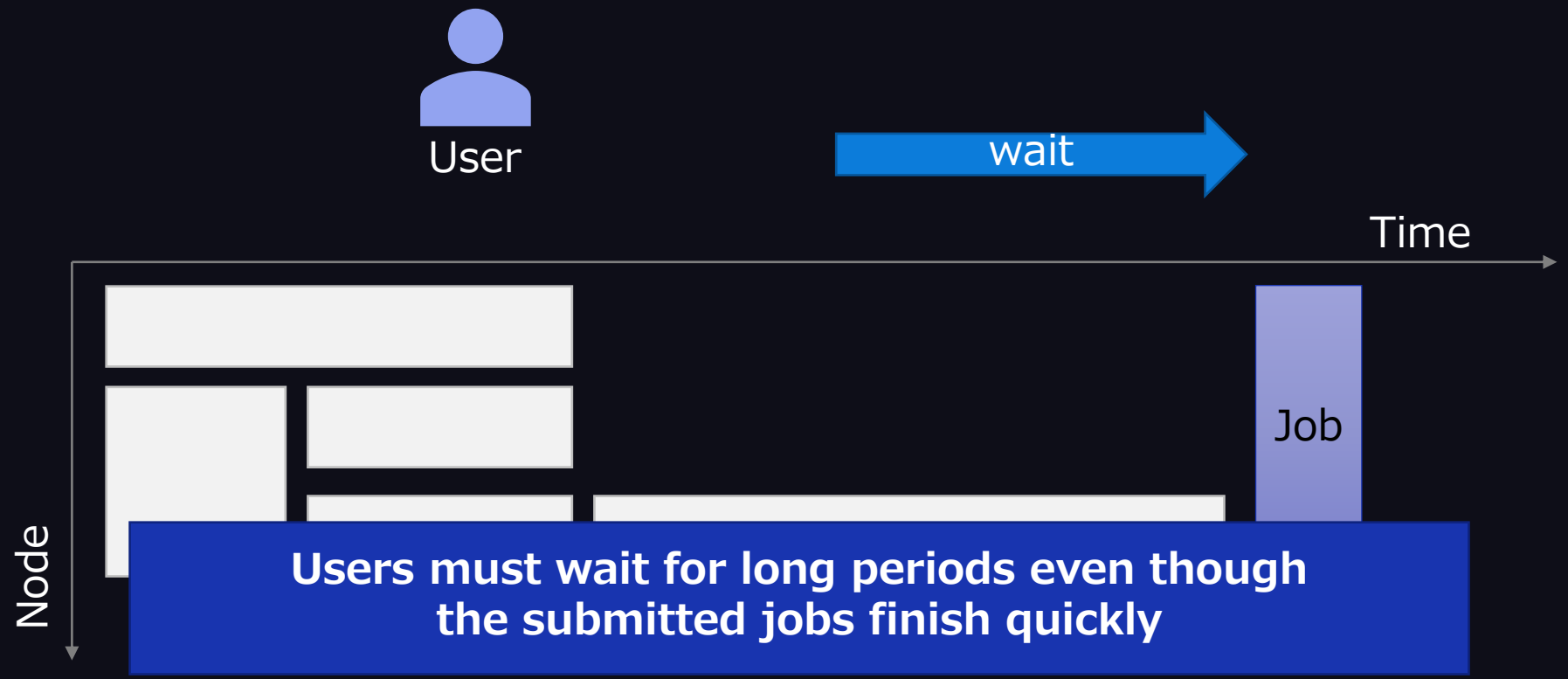
AI/ML Users and the Need for Responsive Job Execution

FUJITSU



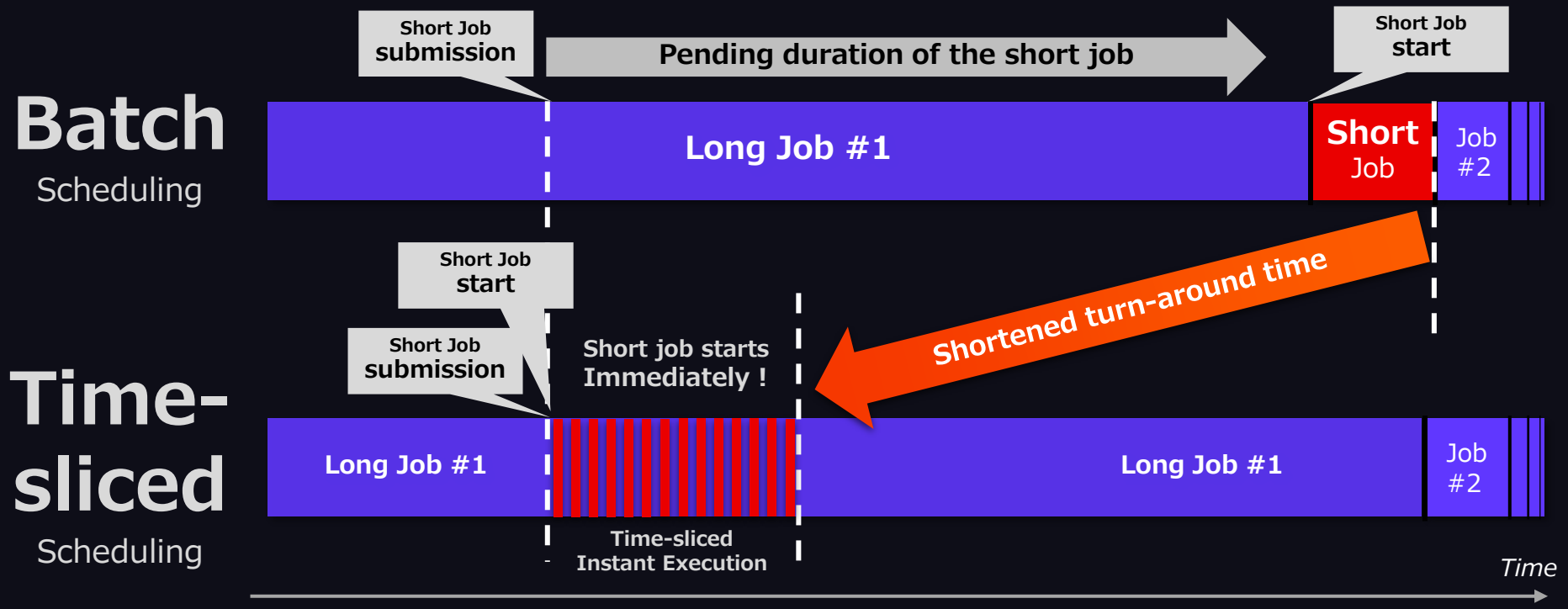
AI/ML Users and the Need for Responsive Job Execution

FUJITSU



Transition from Batch Scheduling

FUJITSU

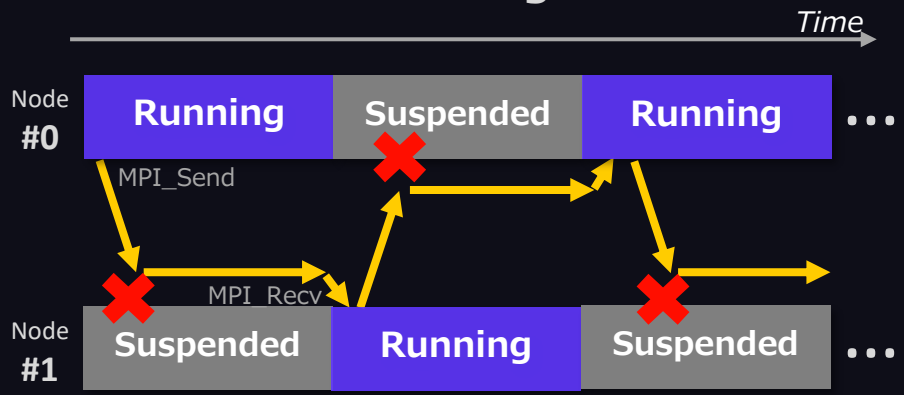


Requirements of Parallel Applications

FUJITSU

Unsynchronized

Scheduling



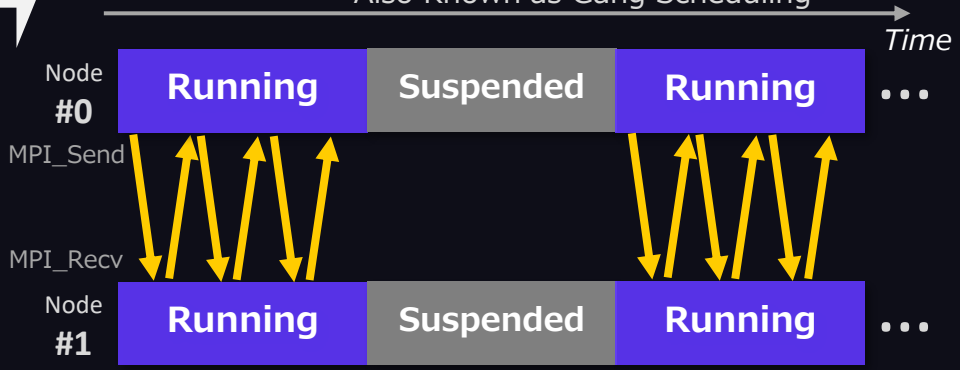
Poor Performance

Only one message can be sent per time-slice (the worst case)

Synchronized

Scheduling

Also Known as Gang Scheduling



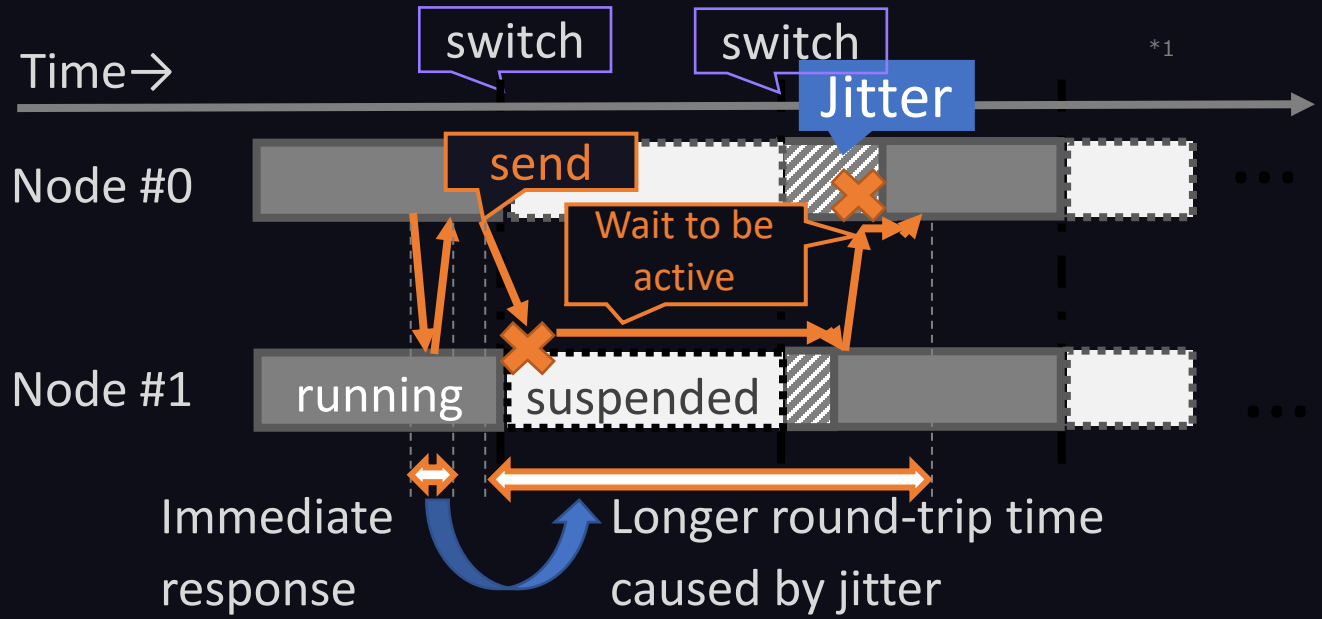
The Same Performance as usual

Multiple messages can be sent in a single time-slice

The Impact of Job Switching Jitter

FUJITSU

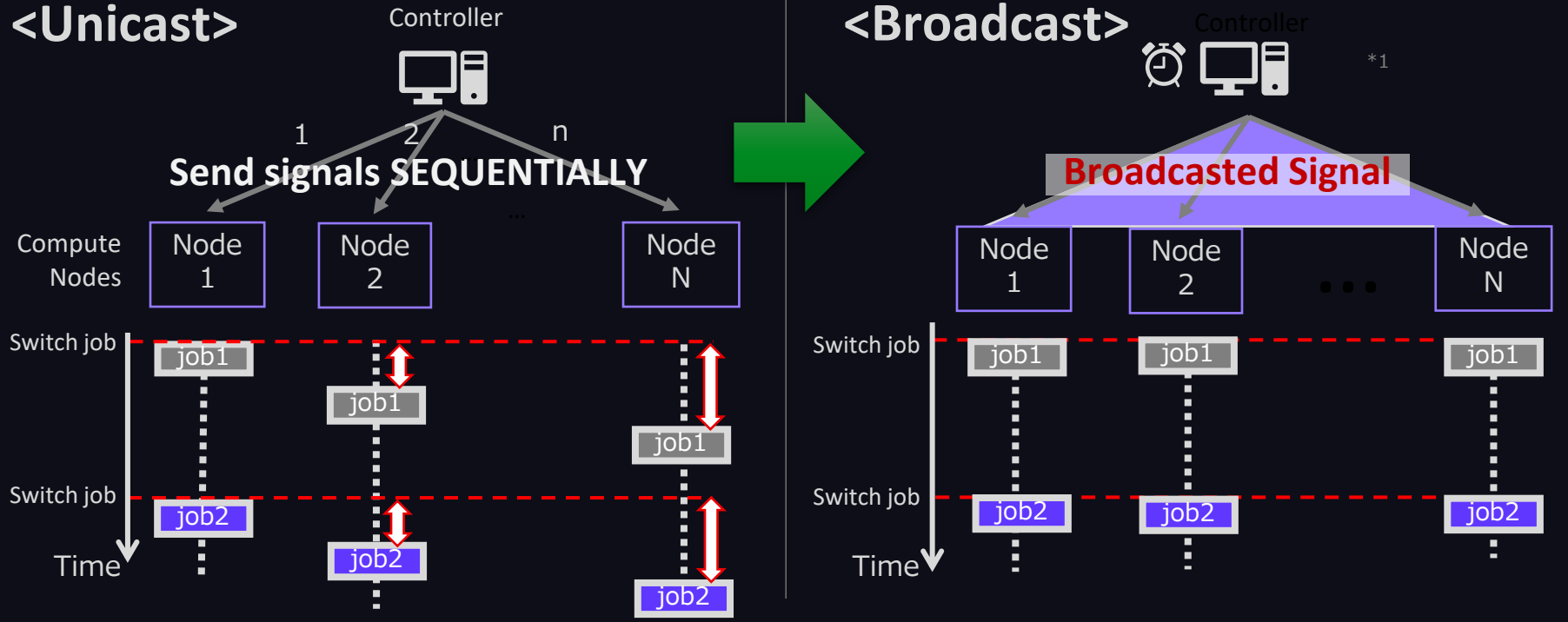
- Impact of jitter on communication performance



Parallel applications with network communication (MPI) require precise synchronization

Scalable Synchronization with Broadcast Control Signals

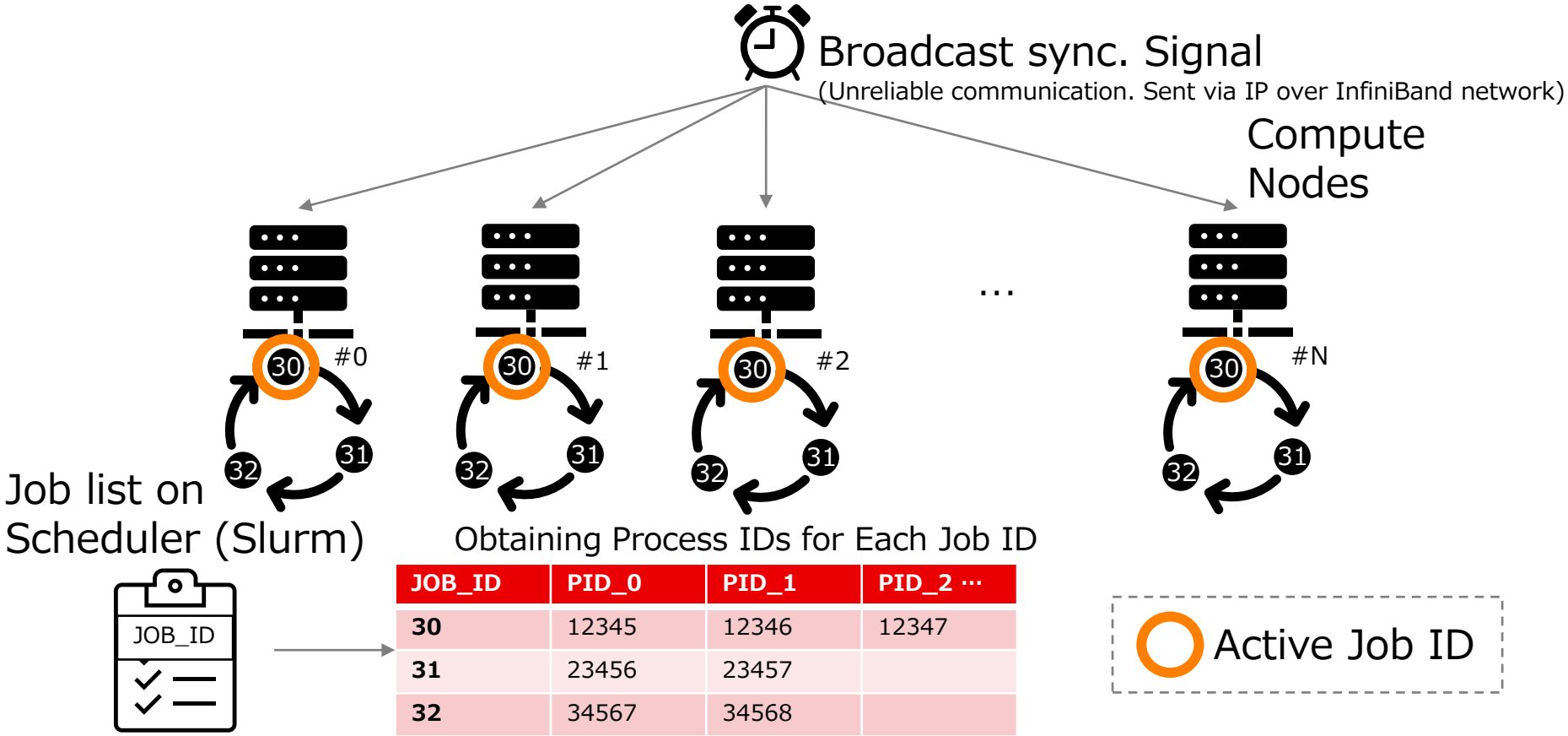
FUJITSU



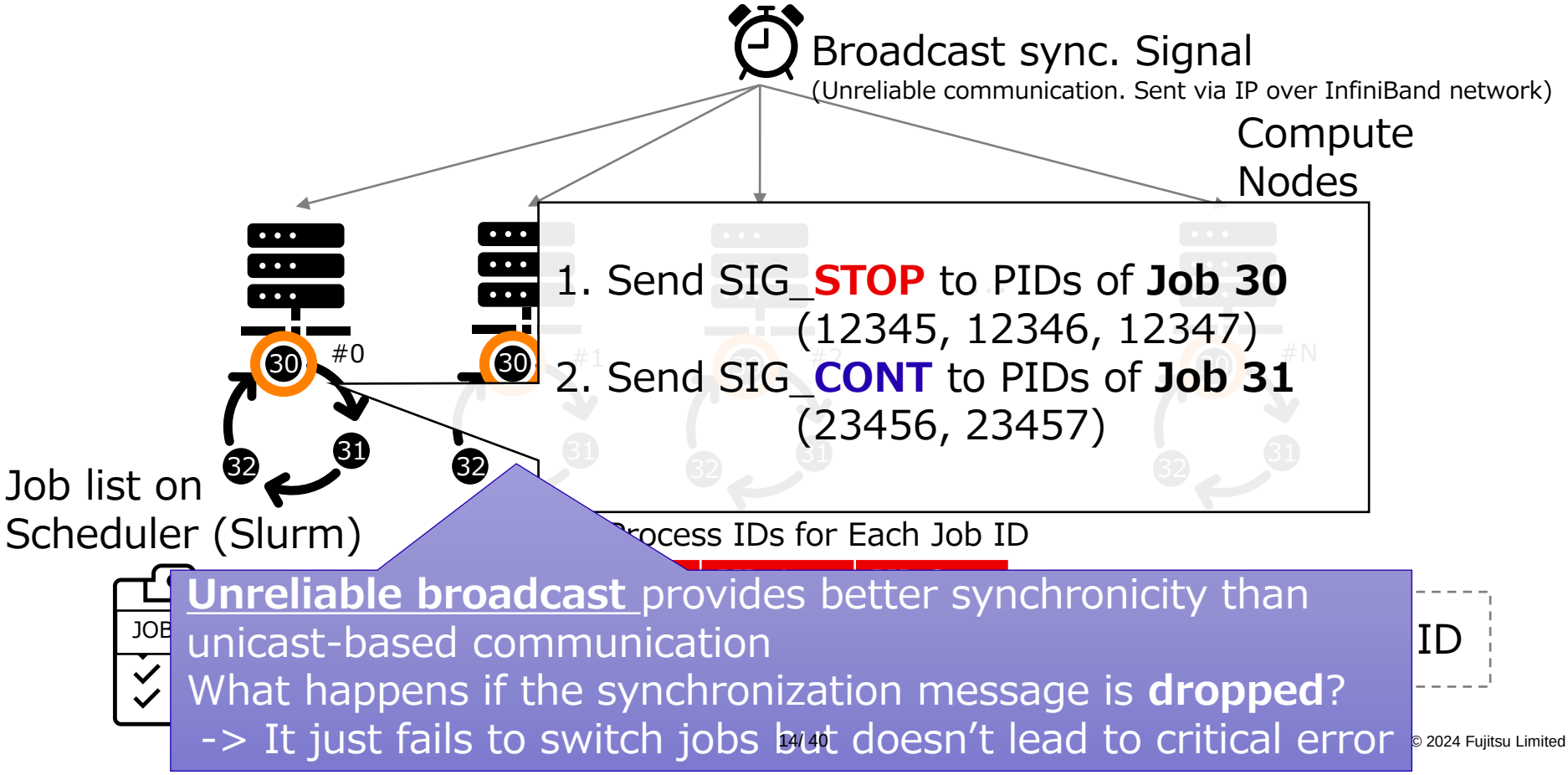
Jitter caused by sequential unicast communications

Good synchronization with a single broadcast communication

Unreliable Broadcast Synchronization in Large-Scale Gang Scheduling Systems



Unreliable Broadcast Synchronization in Large-Scale Gang Scheduling Systems



Unreliable broadcast provides better synchronicity than unicast-based communication

What happens if the synchronization message is **dropped**?

-> It just fails to switch jobs but doesn't lead to critical error

Broadcast Messages: Unreliable Yet Rarely Dropped in Practice

FUJITSU

- Paper [1] reports zero packet loss during the transfer of 50PB data on the InfiniBand Network
  - [1] Kalia et al, FaSST: Fast, Scalable and Simple Distributed Transactions with Two-Sided (RDMA) Datagram RPCs, USENIX OSDI'16

- Similarly, we observed no packet loss during our experimental testing of this system

What Happens if Broadcast Messages are Dropped?

- **Broadcast Control Message Drop:**
  - Leads to failure in job switching.
  - However, it does not cause any error, but rather continues the current job.

- **Performance degradation:**
  - The degradation caused by the drop is limited.
  - If 0.001% of messages are dropped, it results in just 0.001% of performance degradation.

Demos

FUJITSU

- **Running two rdbench* visualization jobs**
  - Simulation programs are running on a parallel cluster
  - Demonstrating how short jobs start immediately with Gang scheduling
- **Jupyter Notebook Interactive Demo**
- **Running Multiple 256-node MPI Jobs Concurrently**

* K. Hiraga, rdbench, https://github.com/range3/rdbench

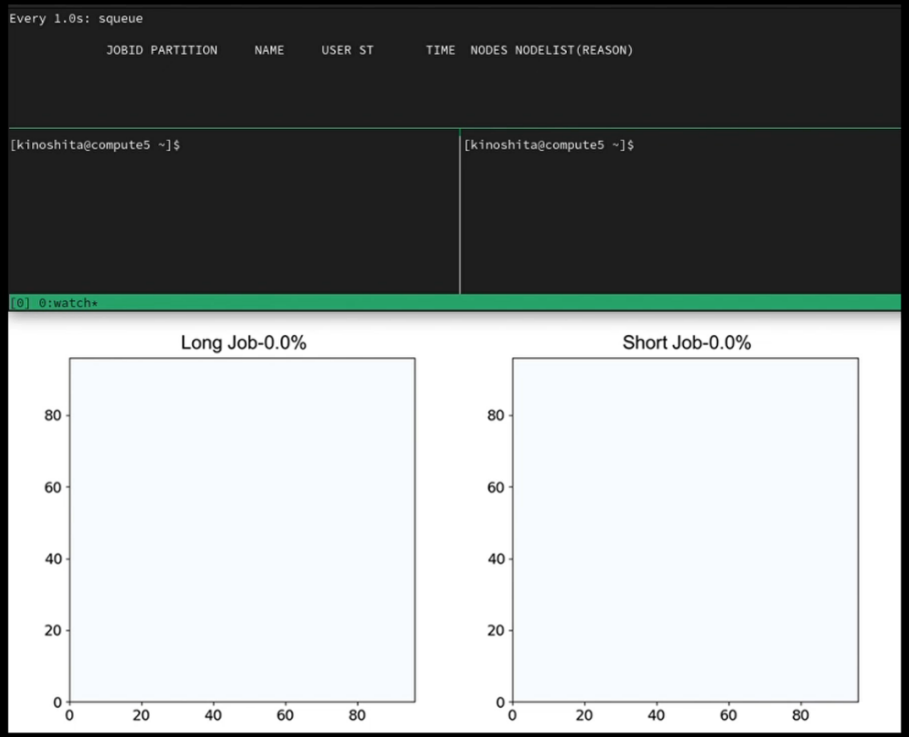
Demo #1

FUJITSU

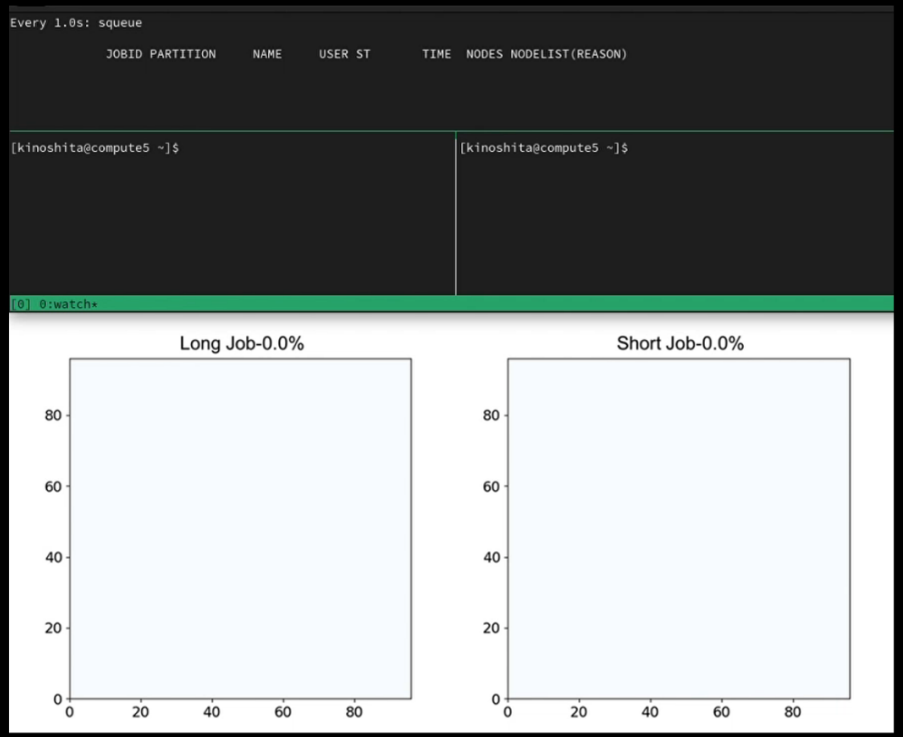
- **Running two rdbench* visualization jobs**
  - Simulation programs are running on a parallel cluster
  - Demonstrating how short jobs start immediately with Gang scheduling

* K. Hiraga, rdbench, https://github.com/range3/rdbench

Batch Scheduling

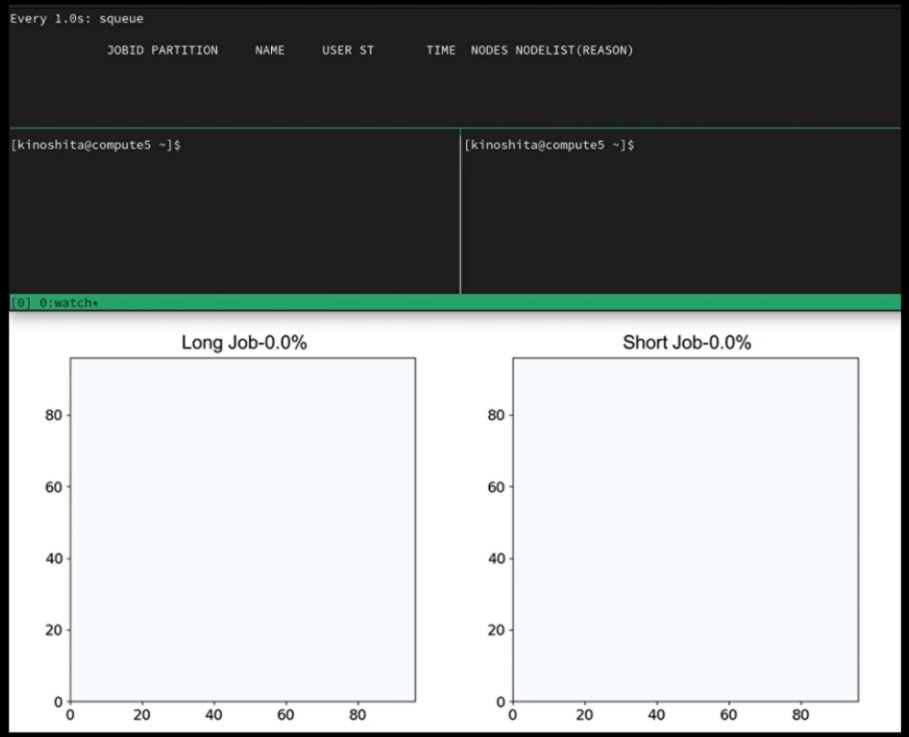


Gang Scheduling

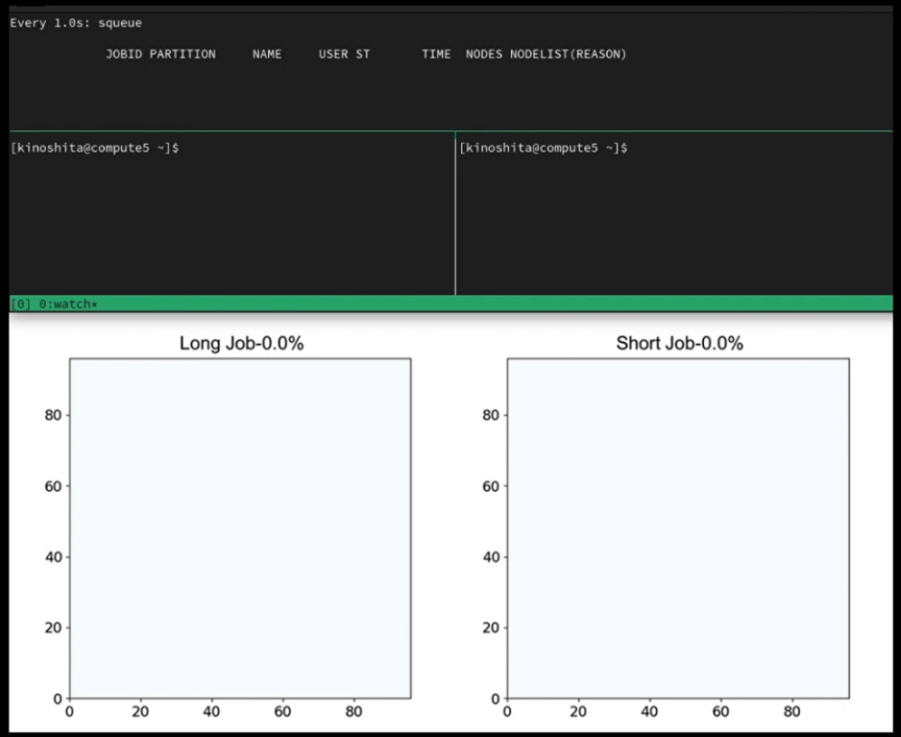


In this demonstration, we submit two jobs on a batch scheduling system and a gang scheduling system.

Batch Scheduling

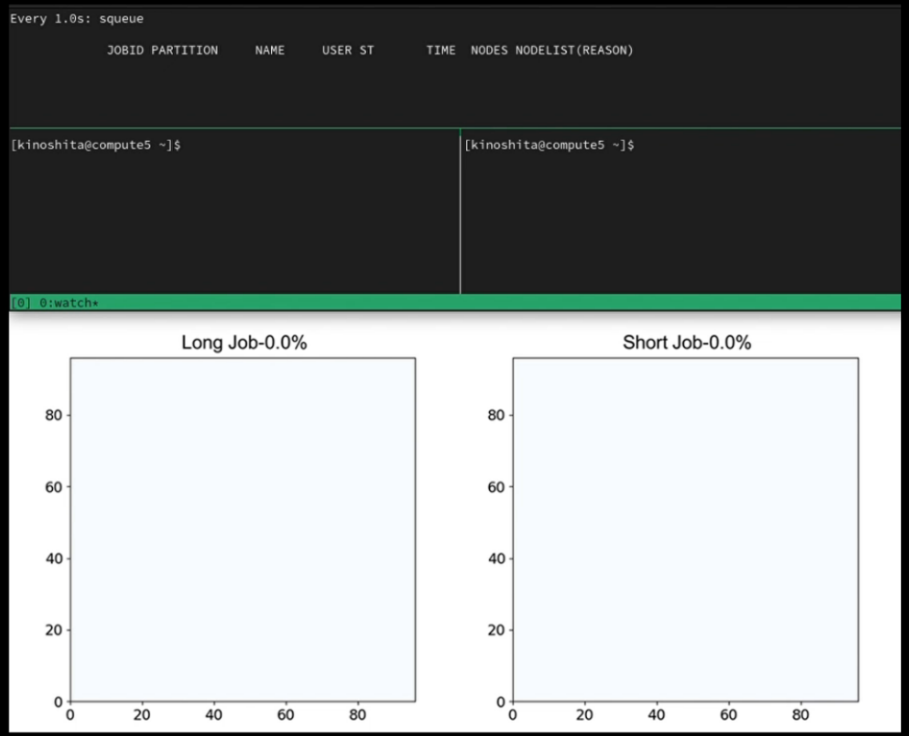


Gang Scheduling

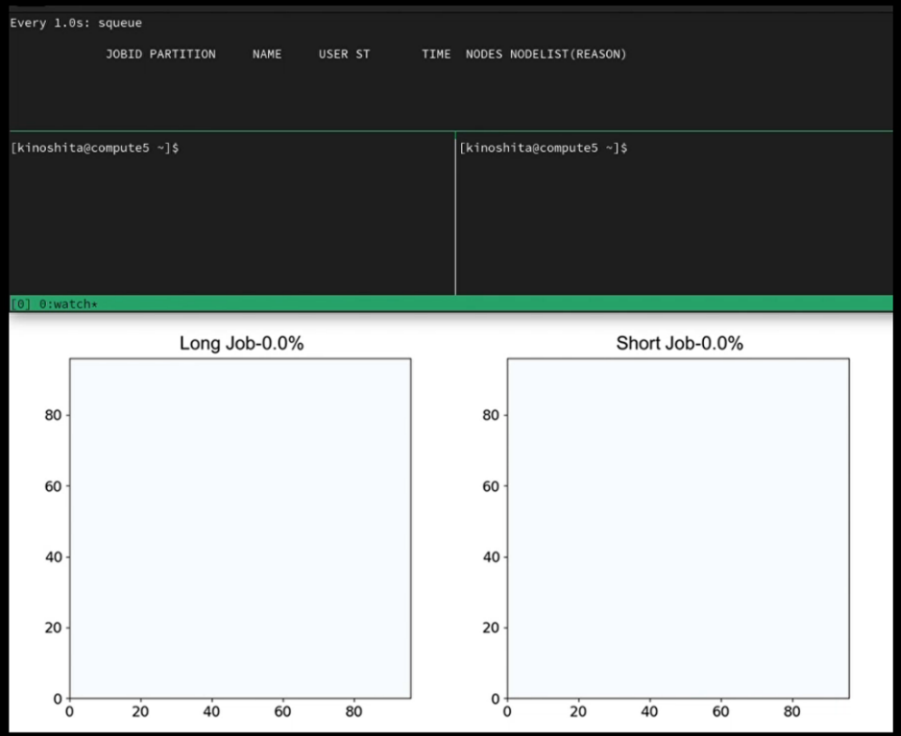


The short job starts instantly with the gang scheduling system while the batch scheduling system requires the completion of the long job.

Batch Scheduling

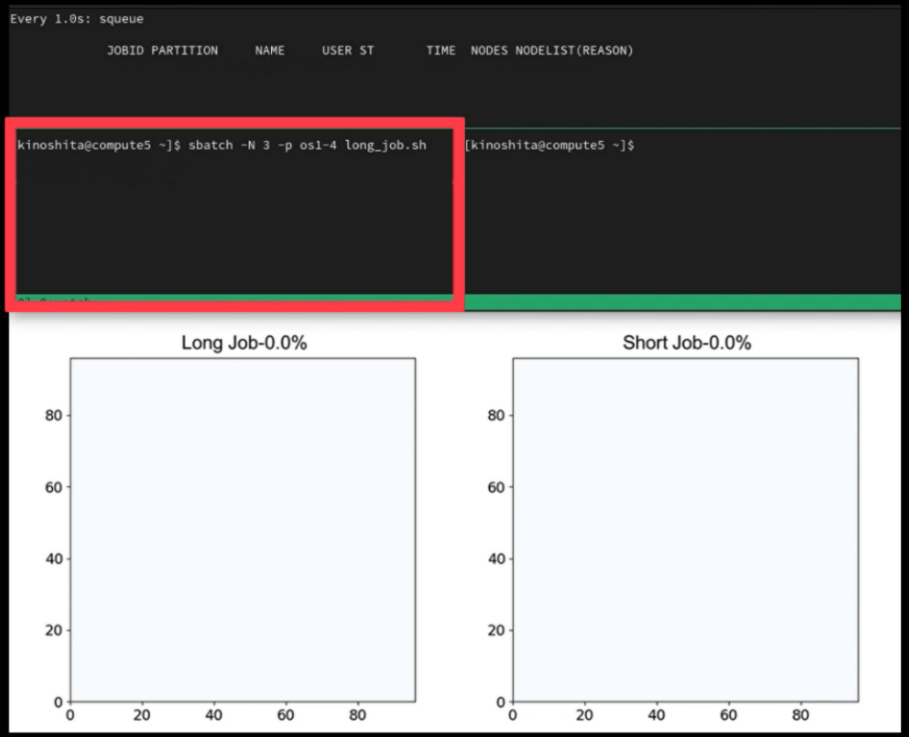


Gang Scheduling

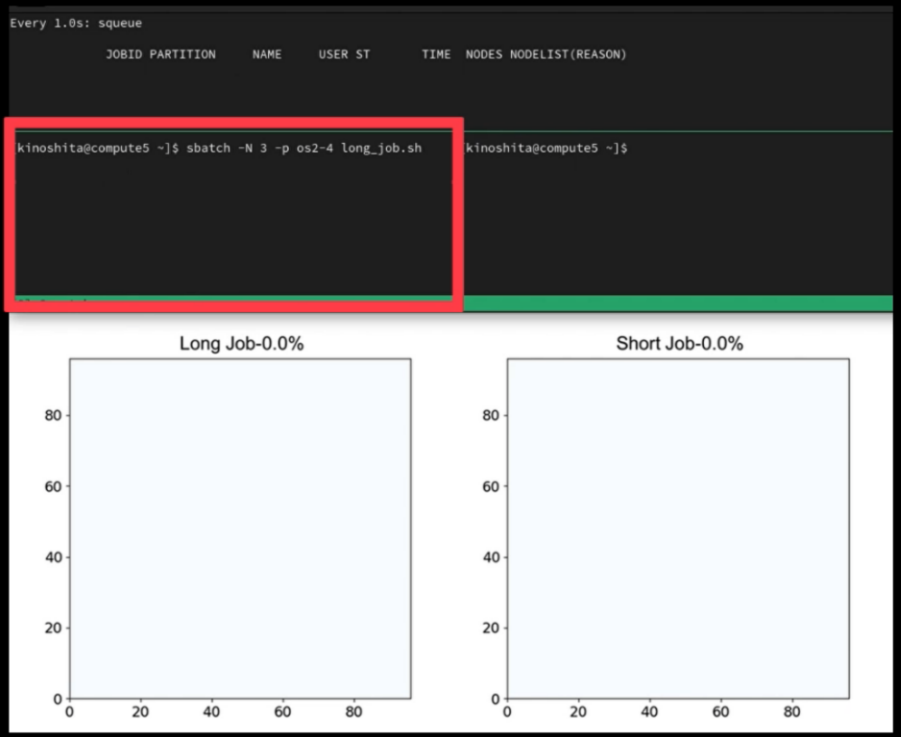


The results of the visualization for each job appear on the movie.

Batch Scheduling



Gang Scheduling

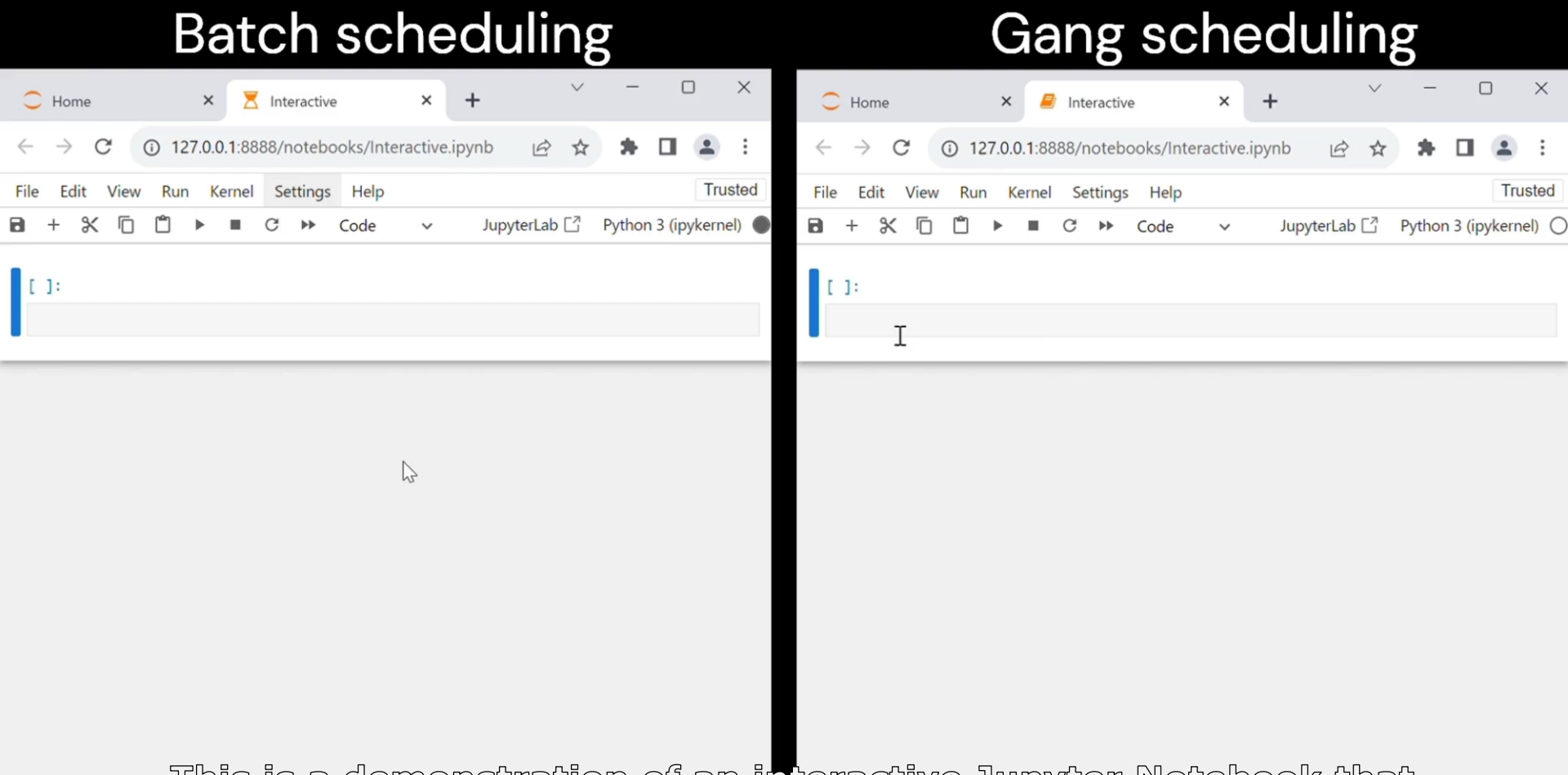


Submitting long jobs on both batch and gang scheduling systems

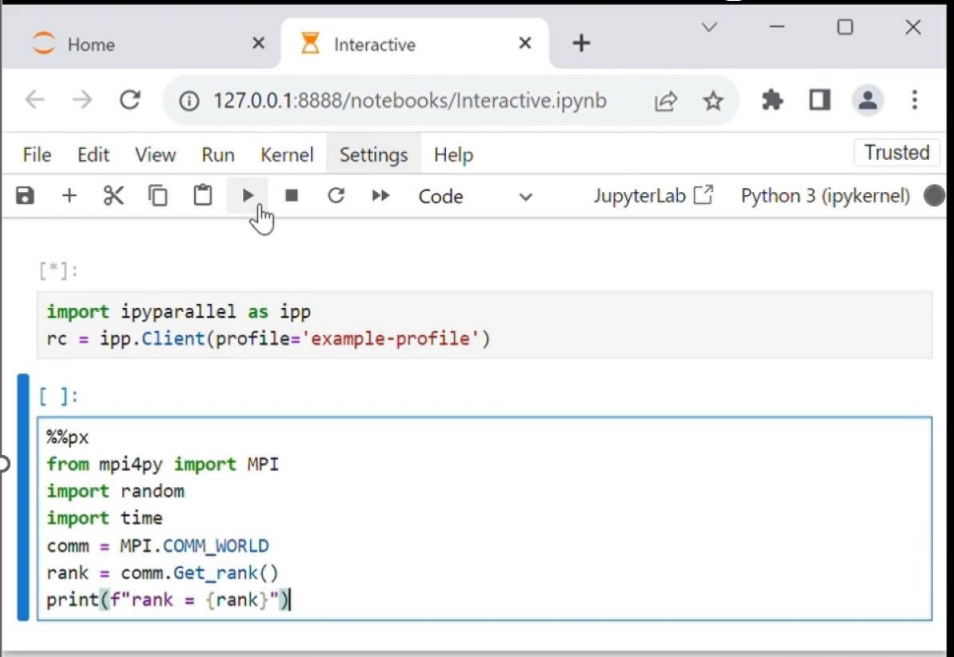
Demo #2

FUJITSU

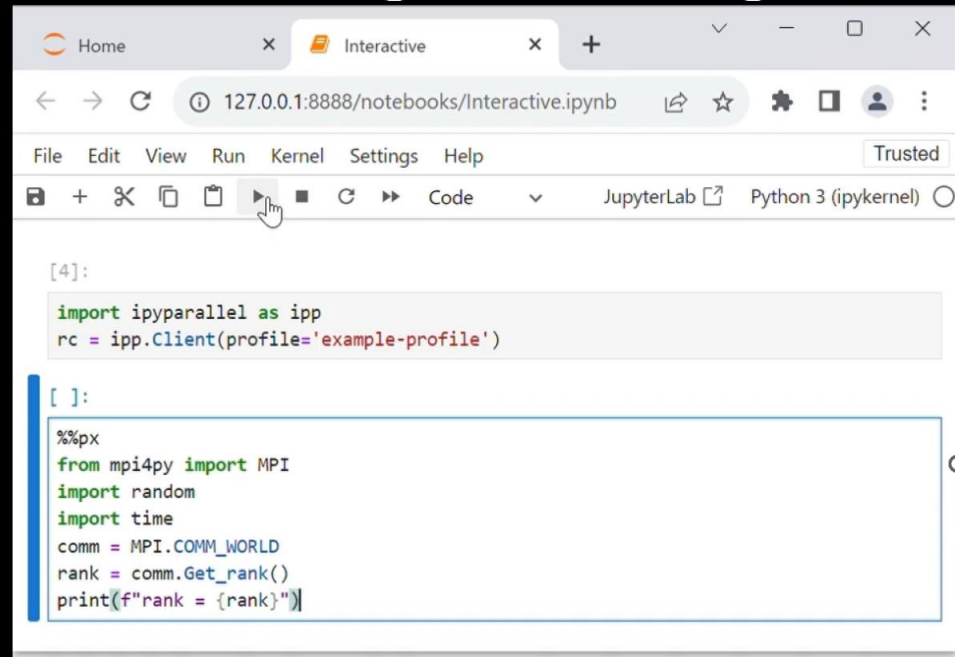
- **Jupyter Notebook Interactive Demo**
  - Instances of IPython parallel are running on a parallel cluster
  - Comparison between batch scheduling and gang scheduling



Batch scheduling

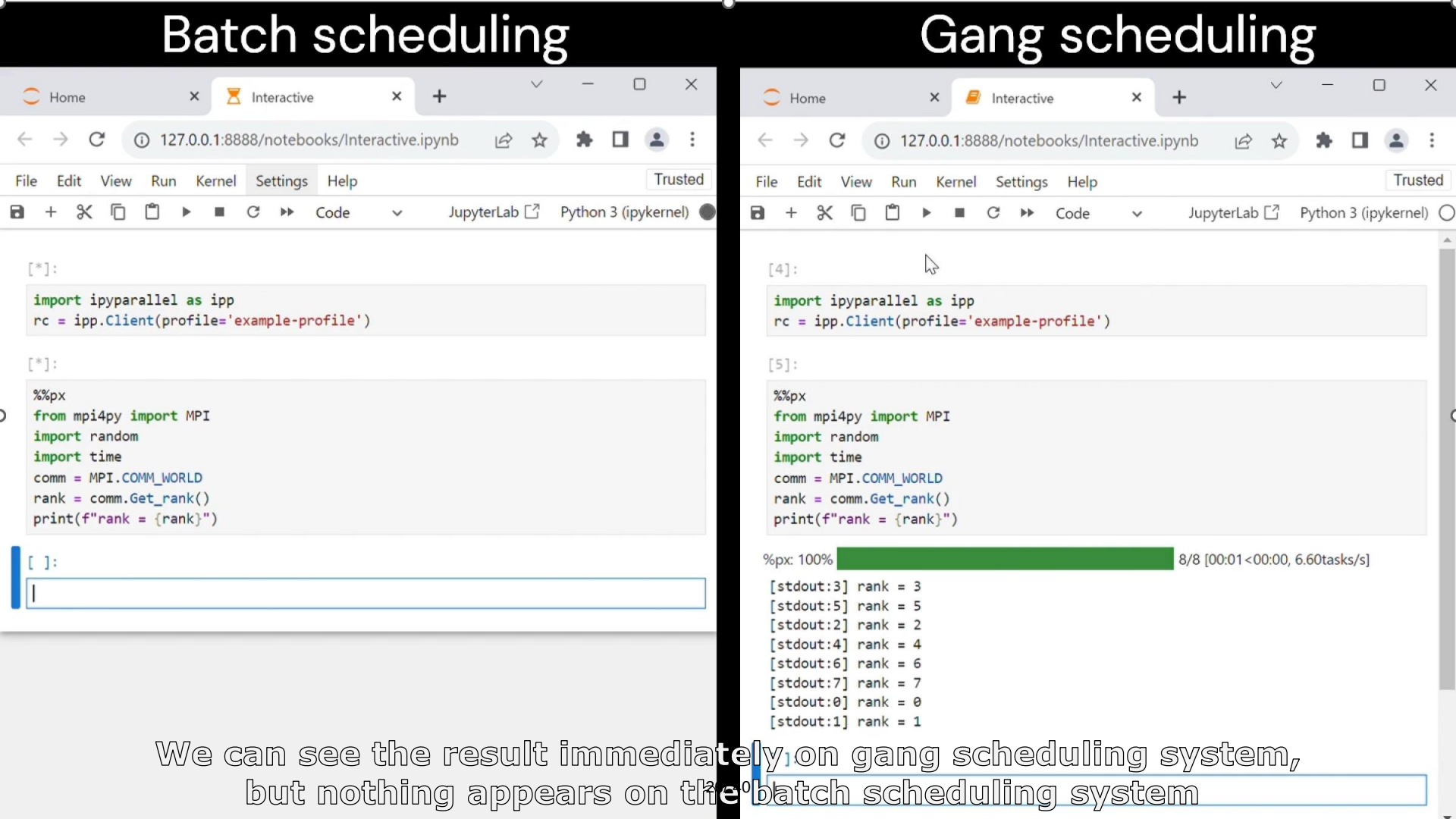


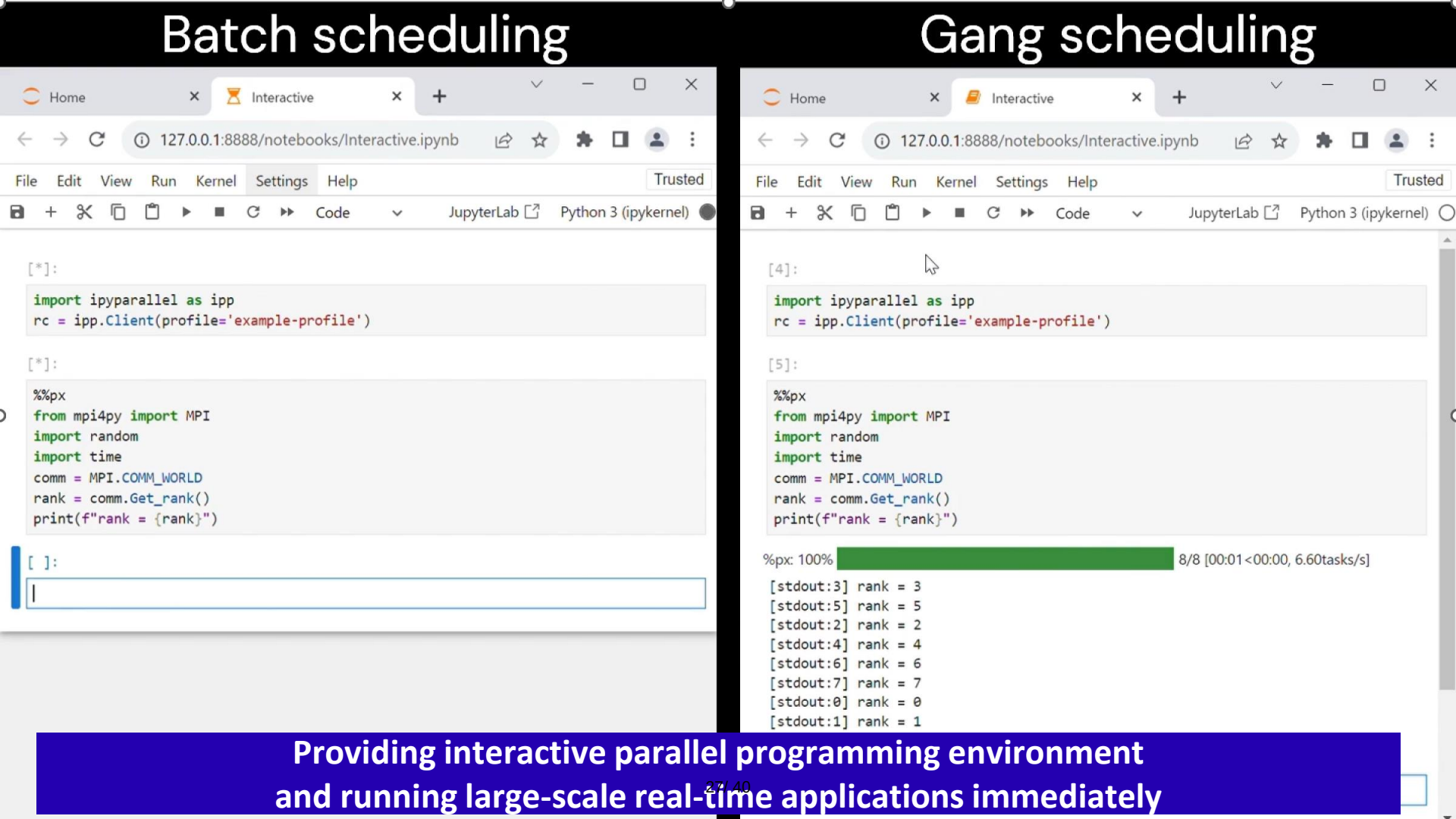
Gang scheduling



Preparing the parallel environment

Writing the parallel code to execute


Batch scheduling
Gang scheduling
Home
Interactive
127.0.0.1:8888/notebooks/Interactive.ipynb
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[*]:
import ipyparallel as ipp
rc = ipp.Client(profile='example-profile')
[*]:
%%px
from mpi4py import MPI
import random
import time
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
print(f"rank = {rank}")
[ ]:
[4]:
import ipyparallel as ipp
rc = ipp.Client(profile='example-profile')
[5]:
%%px
from mpi4py import MPI
import random
import time
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
print(f"rank = {rank}")
%px: 100%
8/8 [00:01<00:00, 6.60tasks/s]
[stdout:3] rank = 3
[stdout:5] rank = 5
[stdout:2] rank = 2
[stdout:4] rank = 4
[stdout:6] rank = 6
[stdout:7] rank = 7
[stdout:0] rank = 0
[stdout:1] rank = 1
[ ]:
We can see the result immediately on gang scheduling system,
but nothing appears on the batch scheduling system


Batch scheduling
Gang scheduling
Home
Interactive
127.0.0.1:8888/notebooks/Interactive.ipynb
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[*]:
import ipyparallel as ipp
rc = ipp.Client(profile='example-profile')
[*]:
%%px
from mpi4py import MPI
import random
import time
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
print(f"rank = {rank}")
[ ]:
[4]:
import ipyparallel as ipp
rc = ipp.Client(profile='example-profile')
[5]:
%%px
from mpi4py import MPI
import random
import time
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
print(f"rank = {rank}")
%px: 100%
8/8 [00:01<00:00, 6.60tasks/s]
[stdout:3] rank = 3
[stdout:5] rank = 5
[stdout:2] rank = 2
[stdout:4] rank = 4
[stdout:6] rank = 6
[stdout:7] rank = 7
[stdout:0] rank = 0
[stdout:1] rank = 1
Providing interactive parallel programming environment
and running large-scale real-time applications immediately

Demo #3 – large scale jobs: Running three 256-node MPI jobs

FUJITSU

- **Running multiple 256-node MPI sample programs**
  - Running on ARM-based compute nodes
  - Multiple jobs are submitted to a 256-node Slurm partition
  - Using Allreduce operations on each iteration
  - Time-slice duration is 100ms

Evaluation Environment (Demo 2 and evaluation)
Computing Node:
FUJITSU Supercomputer PRIMEHPC FX700
CPU: FUJITSU A64FX Arm8.2-A SVE
48cores @ 2.0GHz
RAM: 32 GiB HBM2
Network interface: InfiniBand HDR100
Interconnect: Full-bisection bandwidth fat-tree
OS: Rocky Linux 8.5
Slurm: 22.05
OpenMPI: 4.1.5

[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mp [ohtsuji@scheduler ~]$
i-pi 128 2>&1

[ohtsuji@scheduler ~]$

Submitting a long job

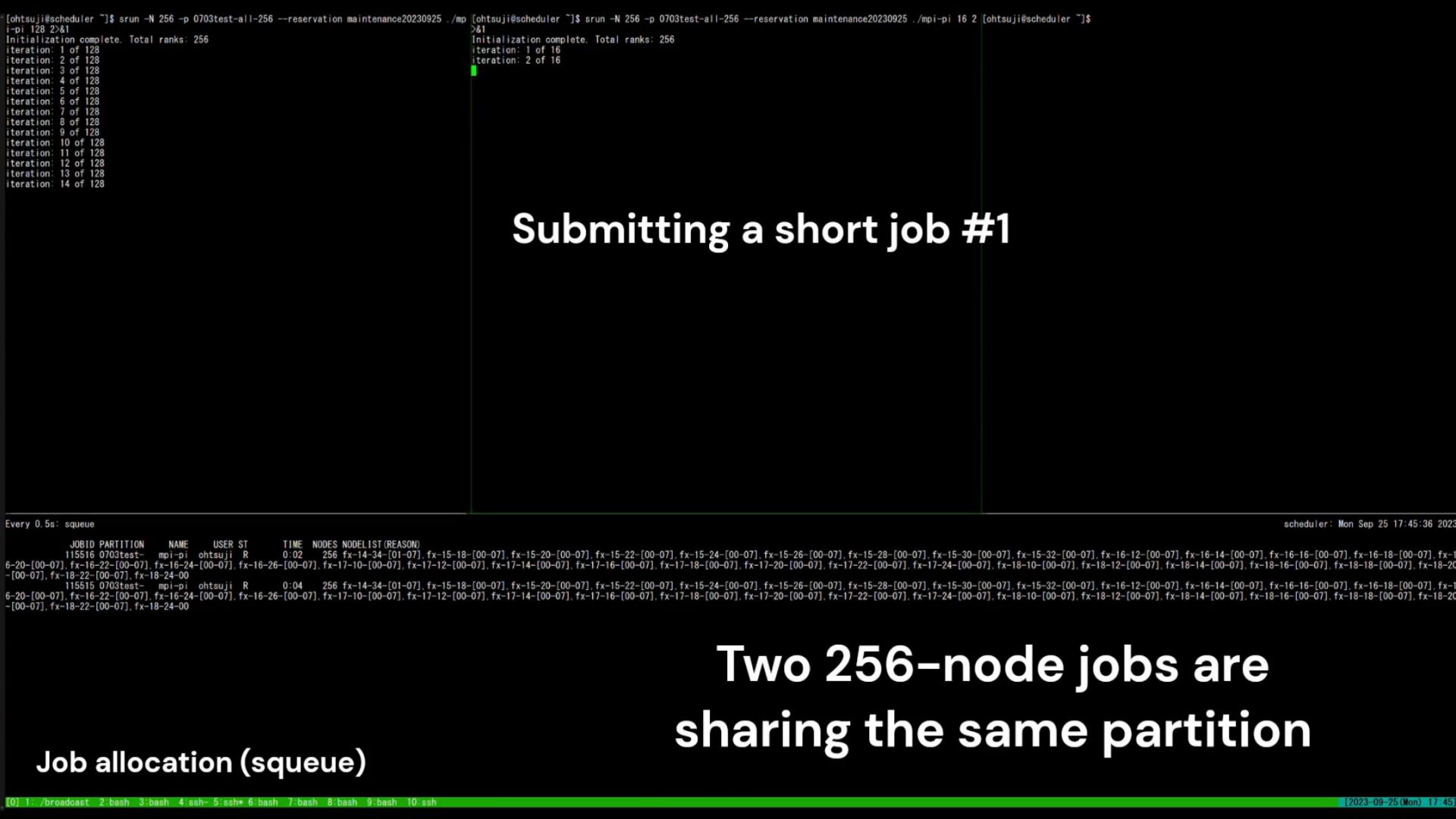
Every 0.5s: squeue scheduler: Mon Sep 25 17:45:33 2023

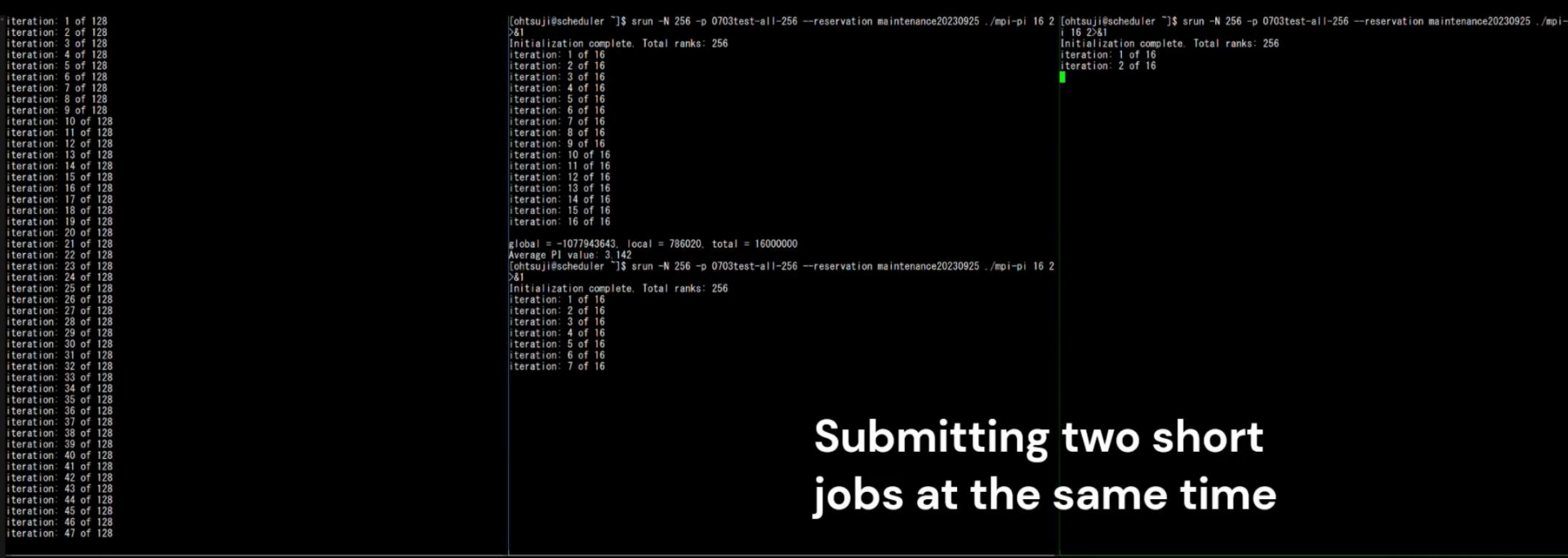
```
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            115515 0703test-   mpi-pi  ohtsuji  R       0:01    256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-1
6-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20
-[00-07],fx-18-22-[00-07],fx-18-24-00
```

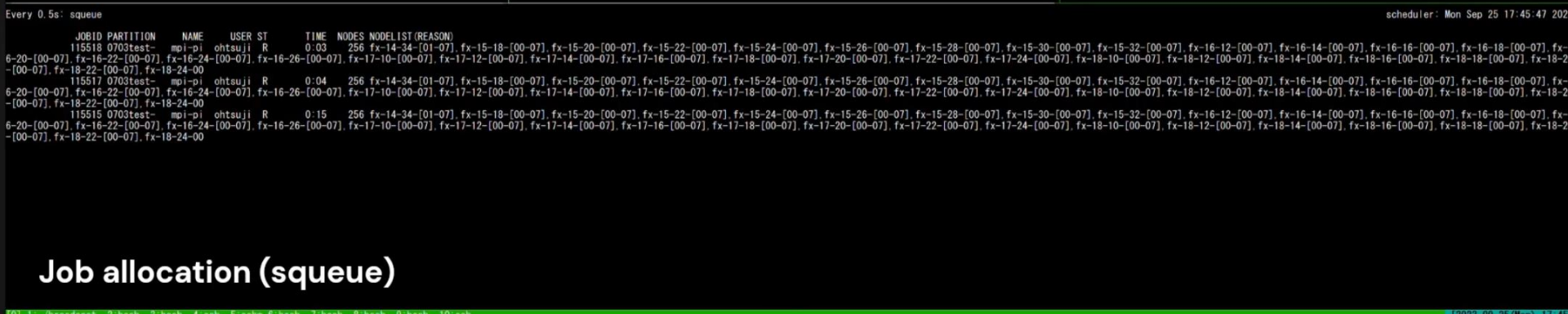
256 nodes allocated

Job allocation (squeue)

[0] 1:./broadcast 2:bash 3:bash 4:ssh- 5:ssh* 6:bash 7:bash 8:bash 9:bash 10:ssh [2023-09-25(Mon) 17:45


[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 128 2>&1
Initialization complete. Total ranks: 256
iteration: 1 of 128
iteration: 2 of 128
iteration: 3 of 128
iteration: 4 of 128
iteration: 5 of 128
iteration: 6 of 128
iteration: 7 of 128
iteration: 8 of 128
iteration: 9 of 128
iteration: 10 of 128
iteration: 11 of 128
iteration: 12 of 128
iteration: 13 of 128
iteration: 14 of 128
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
Submitting a short job #1
Every 0.5s: squeue
scheduler: Mon Sep 25 17:45:36 2023
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
115516 0703test- mpi-pi ohtsuji R 0:02 256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-16-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20-[00-07],fx-18-22-[00-07],fx-18-24-00
115515 0703test- mpi-pi ohtsuji R 0:04 256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-16-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20-[00-07],fx-18-22-[00-07],fx-18-24-00
Two 256-node jobs are sharing the same partition
Job allocation (squeue)
[0] 1:./broadcast 2:bash 3:bash 4:ssh- 5:ssh* 6:bash 7:bash 8:bash 9:bash 10:ssh
[2023-09-25(Mon) 17:45


iteration: 1 of 128
iteration: 2 of 128
iteration: 3 of 128
iteration: 4 of 128
iteration: 5 of 128
iteration: 6 of 128
iteration: 7 of 128
iteration: 8 of 128
iteration: 9 of 128
iteration: 10 of 128
iteration: 11 of 128
iteration: 12 of 128
iteration: 13 of 128
iteration: 14 of 128
iteration: 15 of 128
iteration: 16 of 128
iteration: 17 of 128
iteration: 18 of 128
iteration: 19 of 128
iteration: 20 of 128
iteration: 21 of 128
iteration: 22 of 128
iteration: 23 of 128
iteration: 24 of 128
iteration: 25 of 128
iteration: 26 of 128
iteration: 27 of 128
iteration: 28 of 128
iteration: 29 of 128
iteration: 30 of 128
iteration: 31 of 128
iteration: 32 of 128
iteration: 33 of 128
iteration: 34 of 128
iteration: 35 of 128
iteration: 36 of 128
iteration: 37 of 128
iteration: 38 of 128
iteration: 39 of 128
iteration: 40 of 128
iteration: 41 of 128
iteration: 42 of 128
iteration: 43 of 128
iteration: 44 of 128
iteration: 45 of 128
iteration: 46 of 128
iteration: 47 of 128
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2 >&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16
global = -1077943643, local = 786020, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2 >&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
Submitting two short jobs at the same time


Every 0.5s: squeue
scheduler: Mon Sep 25 17:45:47 202
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
115518 0703test- mpi-pi ohtsuji R 0:03 256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-16-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20-[00-07],fx-18-22-[00-07],fx-18-24-00
115517 0703test- mpi-pi ohtsuji R 0:04 256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-16-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20-[00-07],fx-18-22-[00-07],fx-18-24-00
115515 0703test- mpi-pi ohtsuji R 0:15 256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-16-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20-[00-07],fx-18-22-[00-07],fx-18-24-00
Job allocation (squeue)
[0] 1:./broadcast 2:bash 3:bash 4:ssh- 5:ssh* 6:bash 7:bash 8:bash 9:bash 10:ssh
[2023-09-25(Mon) 17:45

```
iteration: 11 of 128
iteration: 12 of 128
iteration: 13 of 128
iteration: 14 of 128
iteration: 15 of 128
iteration: 16 of 128
iteration: 17 of 128
iteration: 18 of 128
iteration: 19 of 128
iteration: 20 of 128
iteration: 21 of 128
iteration: 22 of 128
iteration: 23 of 128
iteration: 24 of 128
iteration: 25 of 128
iteration: 26 of 128
iteration: 27 of 128
iteration: 28 of 128
iteration: 29 of 128
iteration: 30 of 128
iteration: 31 of 128
iteration: 32 of 128
iteration: 33 of 128
iteration: 34 of 128
iteration: 35 of 128
iteration: 36 of 128
iteration: 37 of 128
iteration: 38 of 128
iteration: 39 of 128
iteration: 40 of 128
iteration: 41 of 128
iteration: 42 of 128
iteration: 43 of 128
iteration: 44 of 128
iteration: 45 of 128
iteration: 46 of 128
iteration: 47 of 128
iteration: 48 of 128
iteration: 49 of 128
iteration: 50 of 128
iteration: 51 of 128
iteration: 52 of 128
iteration: 53 of 128
iteration: 54 of 128
iteration: 55 of 128
iteration: 56 of 128
iteration: 57 of 128
```

```
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2
>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16

global = -1077943643, local = 786020, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2
>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16

global = -1077981451, local = 785194, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$
```

```
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-p
i 16 2>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
```

```
Every 0.5s: squeue                                                                      scheduler: Mon Sep 25 17:45:52 2023

           JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
          115517 0703test-   mpi-pi  ohtsuji CG       0:09    255 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-1
6-20-[00-07],fx-16-22-[00-01,03-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx
-18-20-[00-07],fx-18-22-[00-07],fx-18-24-00
          115518 0703test-   mpi-pi  ohtsuji  R       0:08    256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-1
6-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20
-[00-07],fx-18-22-[00-07],fx-18-24-00
          115515 0703test-   mpi-pi  ohtsuji  R       0:20    256 fx-14-34-[01-07],fx-15-18-[00-07],fx-15-20-[00-07],fx-15-22-[00-07],fx-15-24-[00-07],fx-15-26-[00-07],fx-15-28-[00-07],fx-15-30-[00-07],fx-15-32-[00-07],fx-16-12-[00-07],fx-16-14-[00-07],fx-16-16-[00-07],fx-16-18-[00-07],fx-1
6-20-[00-07],fx-16-22-[00-07],fx-16-24-[00-07],fx-16-26-[00-07],fx-17-10-[00-07],fx-17-12-[00-07],fx-17-14-[00-07],fx-17-16-[00-07],fx-17-18-[00-07],fx-17-20-[00-07],fx-17-22-[00-07],fx-17-24-[00-07],fx-18-10-[00-07],fx-18-12-[00-07],fx-18-14-[00-07],fx-18-16-[00-07],fx-18-18-[00-07],fx-18-20
-[00-07],fx-18-22-[00-07],fx-18-24-00
```

Job allocation (squeue)

Three 256-node jobs are sharing the same partition

```
[0] 1:./broadcast  2:bash  3:bash  4:ssh-  5:ssh* 6:bash  7:bash  8:bash  9:bash  10:ssh          [2023-09-25(Mon) 17:45
```

```
iteration: 85 of 128
iteration: 86 of 128
iteration: 87 of 128
iteration: 88 of 128
iteration: 89 of 128
iteration: 90 of 128
iteration: 91 of 128
iteration: 92 of 128
iteration: 93 of 128
iteration: 94 of 128
iteration: 95 of 128
iteration: 96 of 128
iteration: 97 of 128
iteration: 98 of 128
iteration: 99 of 128
iteration: 100 of 128
iteration: 101 of 128
iteration: 102 of 128
iteration: 103 of 128
iteration: 104 of 128
iteration: 105 of 128
iteration: 106 of 128
iteration: 107 of 128
iteration: 108 of 128
iteration: 109 of 128
iteration: 110 of 128
iteration: 111 of 128
iteration: 112 of 128
iteration: 113 of 128
iteration: 114 of 128
iteration: 115 of 128
iteration: 116 of 128
iteration: 117 of 128
iteration: 118 of 128
iteration: 119 of 128
iteration: 120 of 128
iteration: 121 of 128
iteration: 122 of 128
iteration: 123 of 128
iteration: 124 of 128
iteration: 125 of 128
iteration: 126 of 128
iteration: 127 of 128
iteration: 128 of 128

global = -33885674, local = 785836, total = 128000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$
```

The long job completed

```
global = -1077943643, local = 786020, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2
>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16

global = -1077981451, local = 785194, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-pi 16 2
>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16

global = -1077959275, local = 785317, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$
```

```
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-
i 16 2>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16

global = -1077988639, local = 784990, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$ srun -N 256 -p 0703test-all-256 --reservation maintenance20230925 ./mpi-
i 16 2>&1
Initialization complete. Total ranks: 256
iteration: 1 of 16
iteration: 2 of 16
iteration: 3 of 16
iteration: 4 of 16
iteration: 5 of 16
iteration: 6 of 16
iteration: 7 of 16
iteration: 8 of 16
iteration: 9 of 16
iteration: 10 of 16
iteration: 11 of 16
iteration: 12 of 16
iteration: 13 of 16
iteration: 14 of 16
iteration: 15 of 16
iteration: 16 of 16

global = -1077997074, local = 785302, total = 16000000
Average PI value: 3.142
[ohtsuji@scheduler ~]$
```

```
Every 0.5s: squeue                                                                    scheduler: Mon Sep 25 17:46:17 202

             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

Job allocation (squeue)

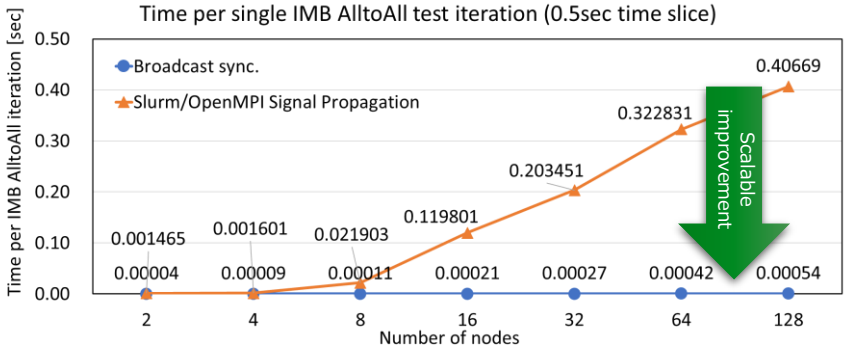
```
[0] 1:./broadcast  2:bash  3:bash  4:ssh-  5:ssh*  6:bash  7:bash  8:bash  9:bash  10:ssh                    [2023-09-25(Mon) 17:46
```

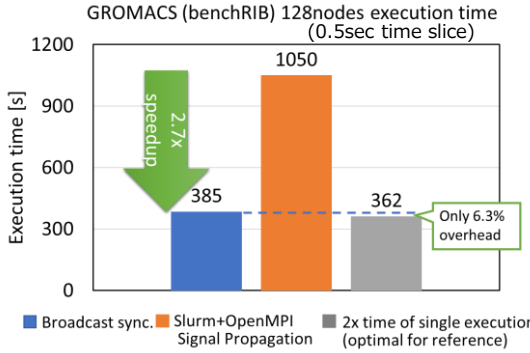
Evaluation

FUJITSU

* SC23 Research Poster, Scalable Fine-Grained Gang Scheduling for HPC Systems with Unreliable Broadcast Synchronization Mechanisms

- Performance of communication and a real application
  - Comparison: Our broadcast sync and Slurm/OpenMPI signal propagation
  - Communication performance
    - MPI AlltoAll: Improved scalability
  - Application performance
    - **2.7x speedup** on GROMACS (128nodes)
    - The overhead caused by time-sliced execution was only **6.3%**





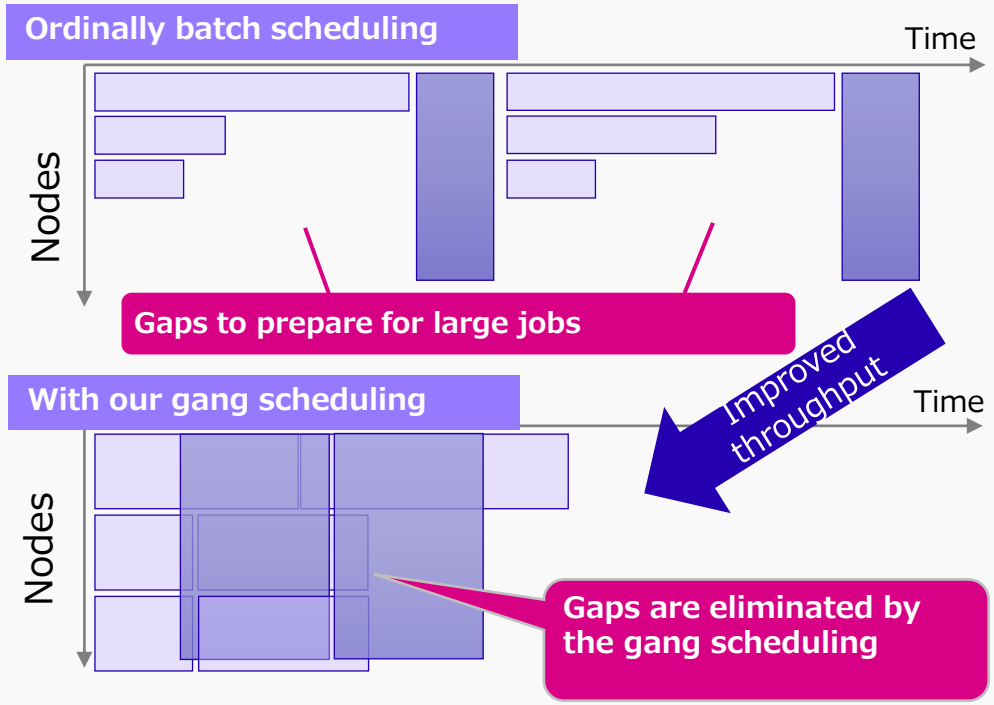
Unreliable Broadcast synchronization enables scalable interactivity on large-scale clusters

Improved Interactivity and Throughput

FUJITSU

- We have already deployed the mechanism on the actual system
  - Integrated with Slurm Scheduler
- Successfully eliminated gaps caused by large jobs
- We confirmed
  - Shortened waiting time
  - Job throughput acceleration

  on our on-premise cluster

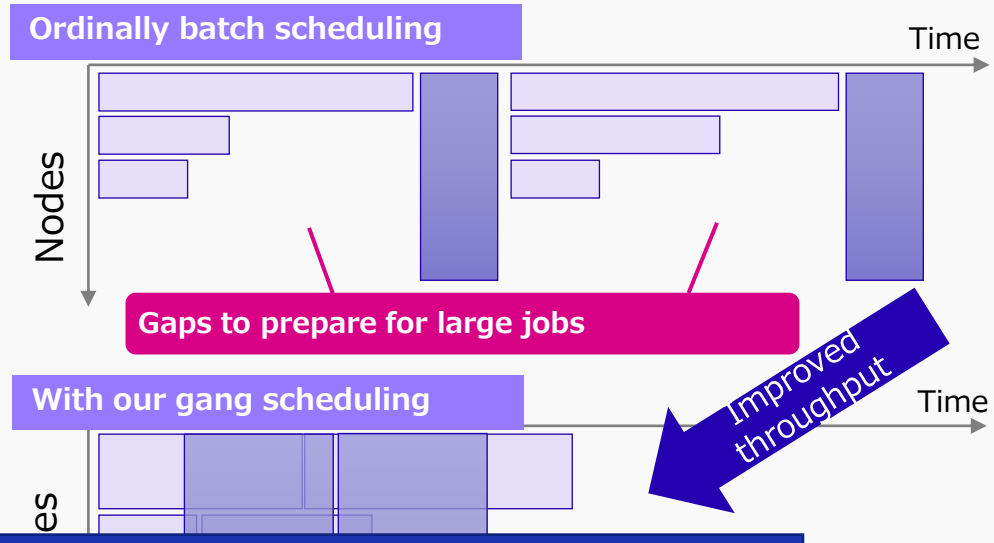


Improved Interactivity and Throughput

FUJITSU

- We have already deployed the mechanism on the actual system
  - Integrated with Slurm Scheduler
- Successfully eliminated gaps caused by large jobs
- We confirmed
  - Shortened waiting time
  - Job throughput acceleration

on



We have implemented this mechanism on the 1056-nodes ARM-based cluster. More results will be available soon!

ated by uling

実システムへの適用および開発状況

FUJITSU

- 富士通の量子コンピューティングシミュレータに適用済
  - FX700 1056ノードのクラスタシステム
    - 詳細はPCCC AI/HPC OSS活用ワークショップ（2024/2/5）における発表を参照
- インタラクティブHPC技術の動作環境
  - 現時点においてはSlurm環境に適用可能
    - その他のスケジューラも仕組み上は対応可能。詳細はお問合せください。
- お試しパッケージ
  - システム全体に適用することなく、ユーザ権限でお試しいただけるパッケージの提供を準備中です
  - **PoC先募集中！**

FUJITSU

Conclusion

Conclusion

FUJITSU

- **インタラクティブHPC技術**
  - HPCシステムのインタラクティブ性と効率を向上するための技術
    - 汎用ハード・既存システムに上乗せで使用可能なGang Scheduling
  - 全系高精度同期技術により、並列アプリケーションの時間分割実行を高効化
  - 全系ジョブによる実行遅延を減らすとともに、システム利用効率を向上

- 実システムへの適用実績および開発状況
  - 1056ノードのFX700クラスタにおいて実運用中(2024年2月～)
  - お試しパッケージを準備中。ご興味のある方はご連絡ください。

- ご質問・ご連絡は以下までお願いいたします
  - ohtsuji.hiroki at fujitsu.com

Thank you