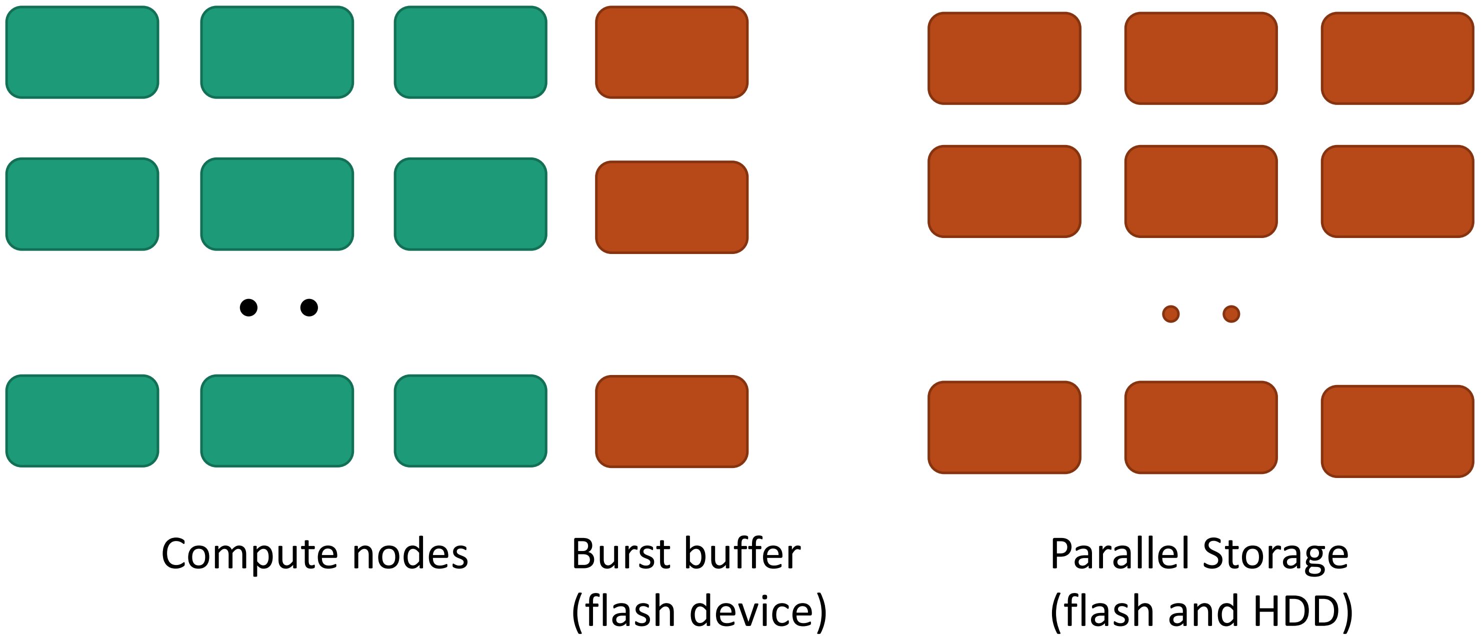
Bridging a gap between computing and storage

Osamu Tatebe

University of Tsukuba

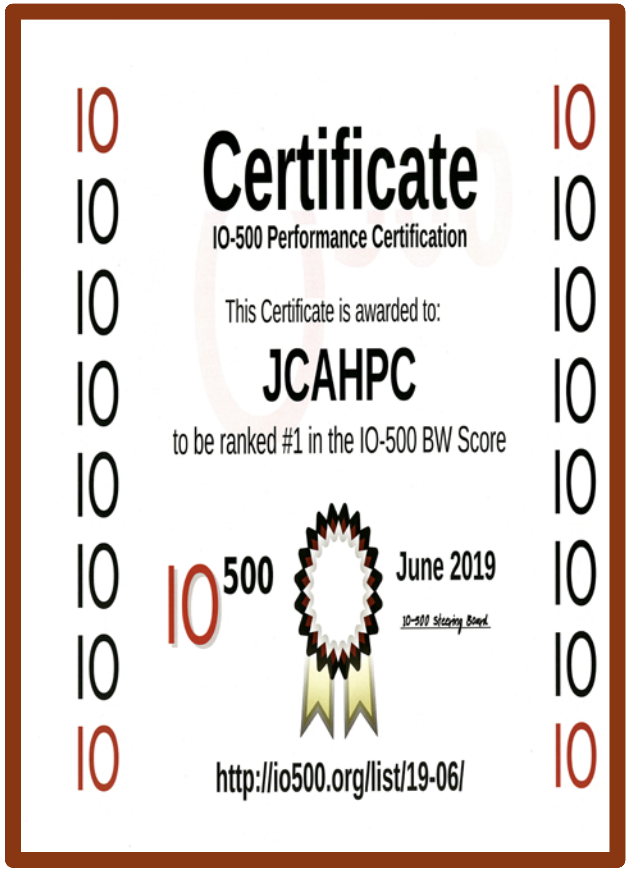
Gap between computing and storage

- CPU/GPU performance grows (Tesla V100 7 TFLOPS)
- HDD bandwidth cannot catch up (~ 250 MB/s)
- Flash and persistent memory may reduce the gap



Burst Buffer (BB)

- Intermediate storage layer to accommodate the burst HPC I/O traffic
  - Checkpointing, output at each time step
  - PLFS [SC09]
- Store temporal data between jobs, prefetch the input data, cache the input/output data
- JCAHPC Oakforest-PACS IME ranked #1 in IO-500 BW
  - More than 1 TB/s
- Exploitation of intermediate storage layer is a key to narrow the gap



Certificate

IO-500 Performance Certification

This Certificate is awarded to:

JCAHPC

to be ranked #1 in the IO-500 BW Score

IO500

June 2019

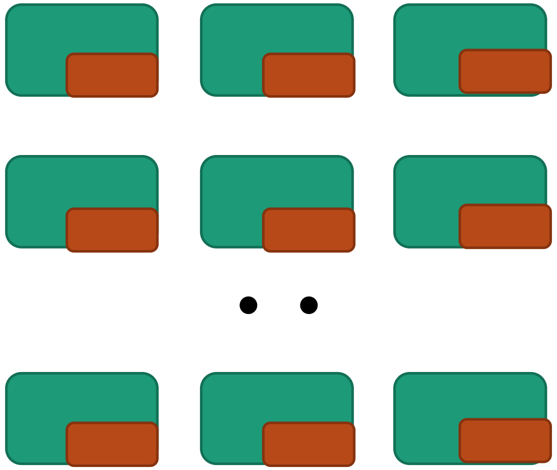
IO-500 Steering Board

http://io500.org/list/19-06/

Gfarm/BB burst buffer system [Tatebe 2019]

- On-demand temporal file system using node-local storage for burst buffer

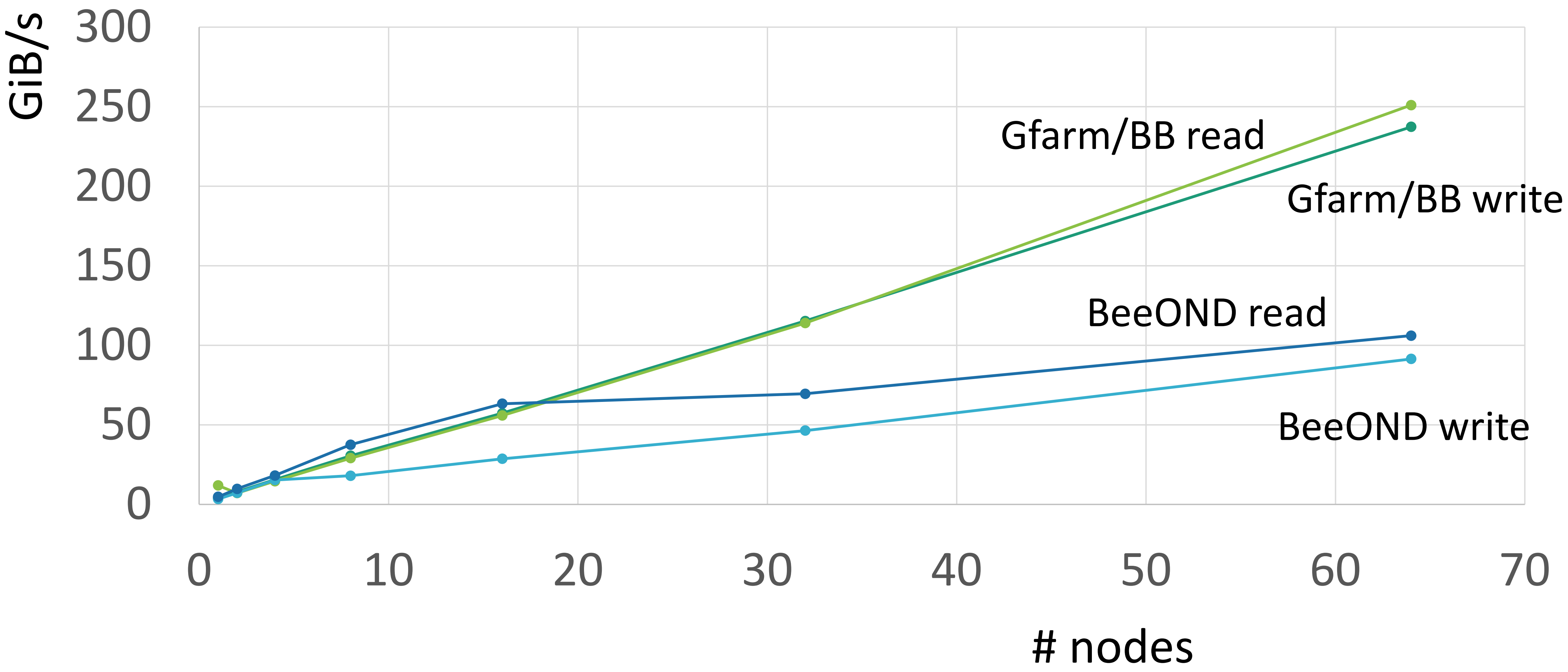
```
gfarmbb –h hostfile –m mount_point start
…
gfarmbb –h hostfile stop
```



Compute nodes

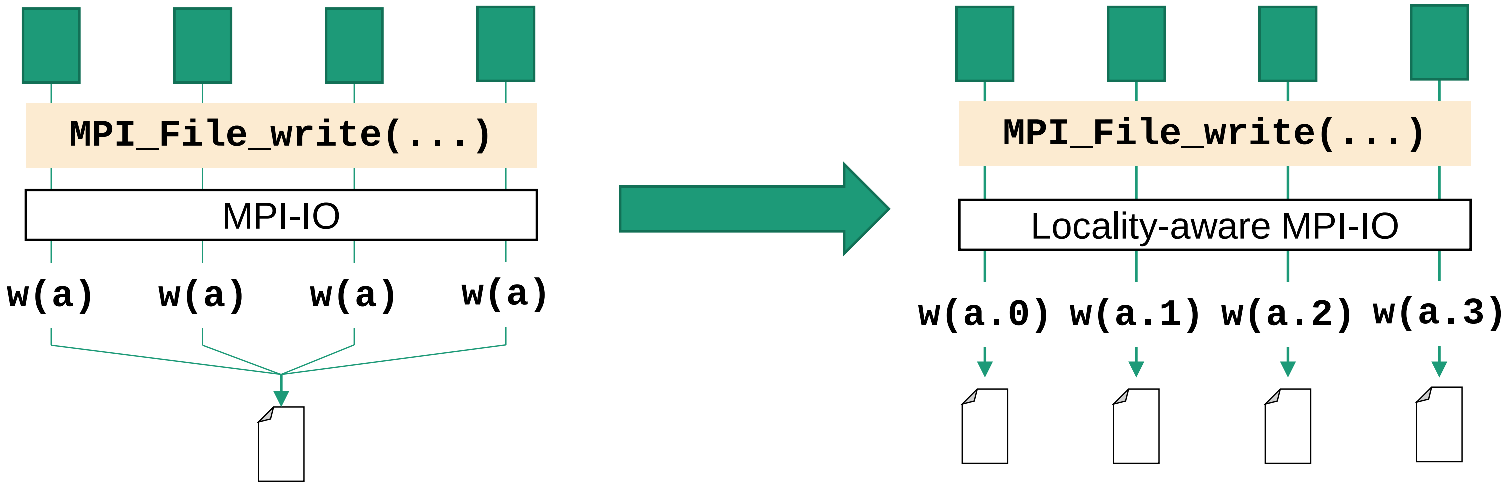
- Based on Gfarm file system that exploits the locality of I/O access
- RDMA data transfer, tradeoff between metadata performance and fault tolerance

IOR – file-per-process read/write bandwidth on Cygnus supercomputer

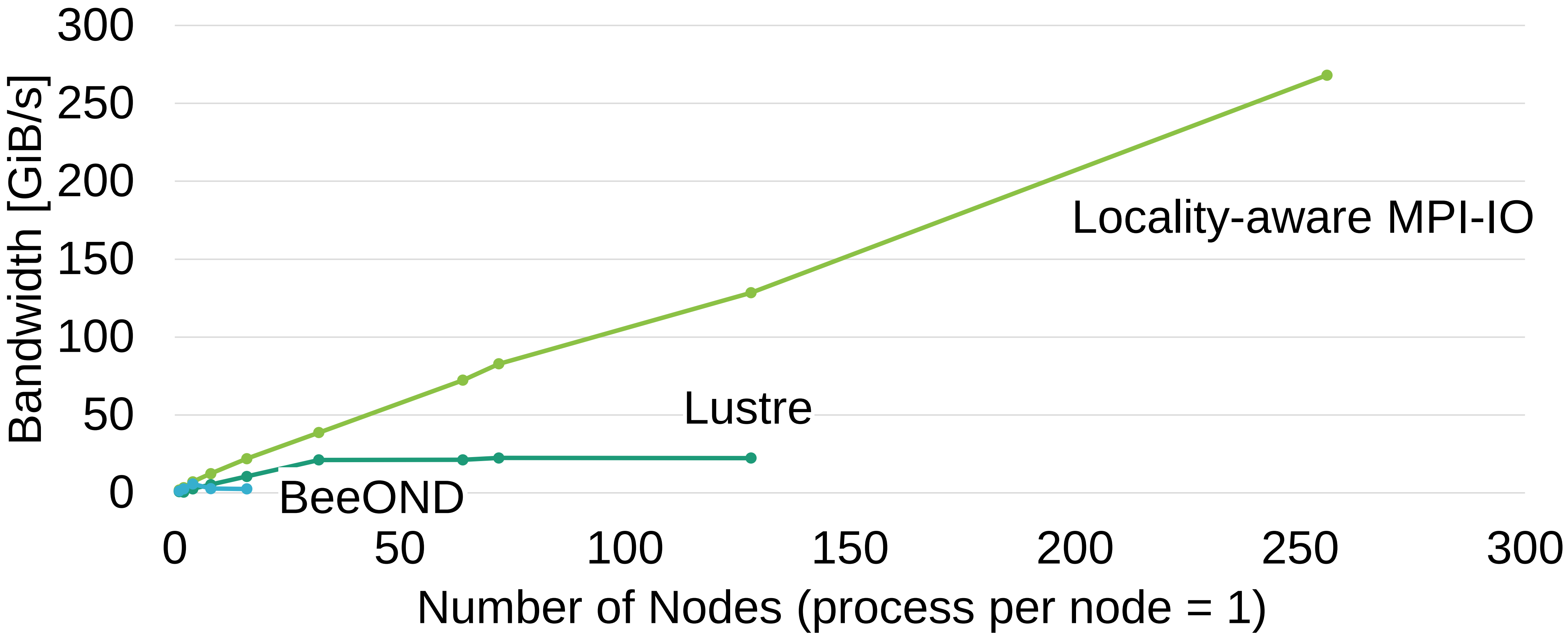


Locality aware MPI-IO [Sugihara]

- Single shared file access may kill the storage performance
- Locality aware MPI-IO converts to file-per-process access and stores to the intermediate storage layer (Gfarm/BB)
- When staging out to a parallel file system, they are copied to an expected single file

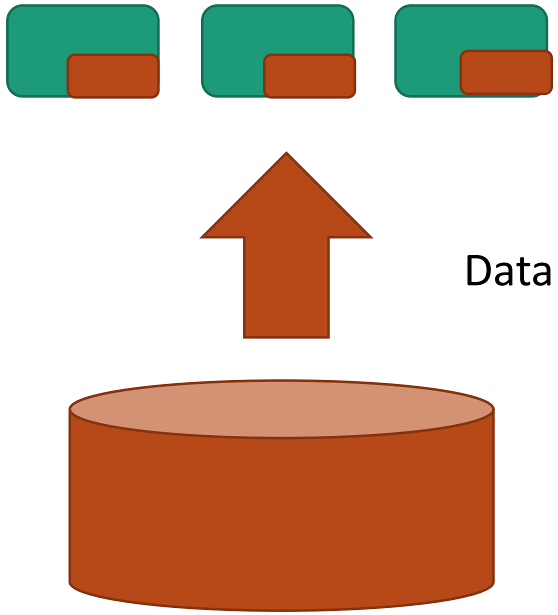


Preliminary evaluation of IOR single-shared-file write bandwidth on Tsubame 3.0 supercomputer



I/O performance for Large-scale Deep Learning [Serizawa BDCAT2019]

- Data staging to node-local storage is often required



Node-local storage

Data staging

Parallel file system

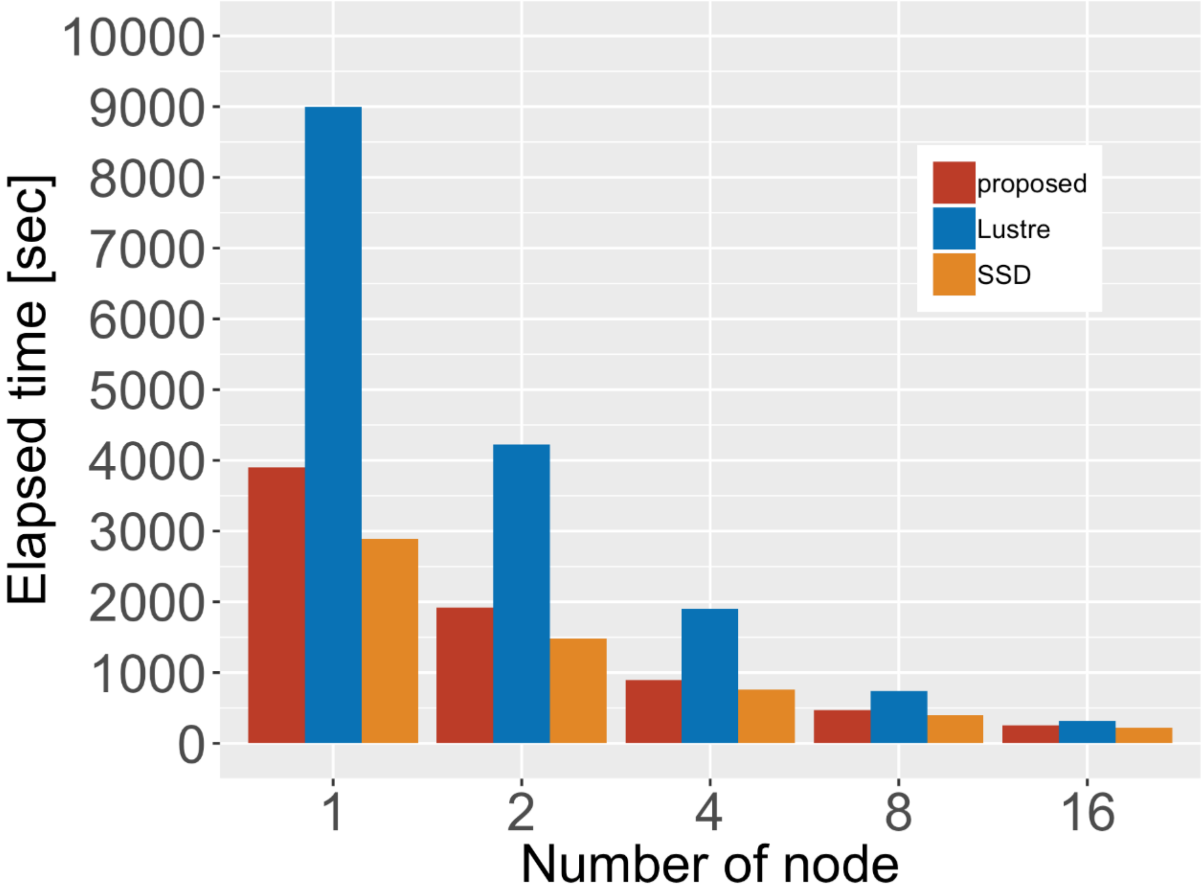
- This data staging takes time
  - Few times more than the training time

↓

- Data prefetching of small random files

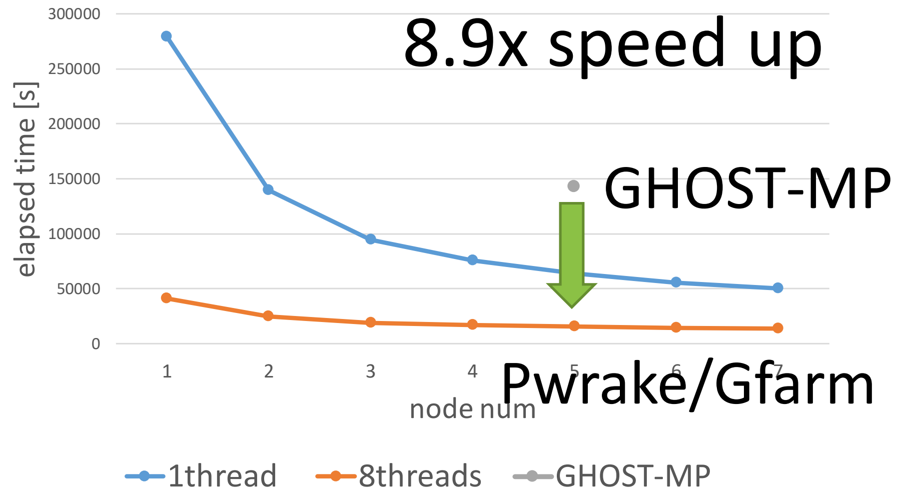
Evaluation of training time for two epochs on Cygnus supercomputer

- Implement using ChainerMN
- Comparison with Lustre and SSD
  - Lustre – no staging
  - SSD – data staged in already (ideal case)
- Proposed method shows close performance to the ideal case
  - Staging time can be almost hidden



Parallel and distributed meta-genome analysis using Gfarm/BB burst buffer [Machida]

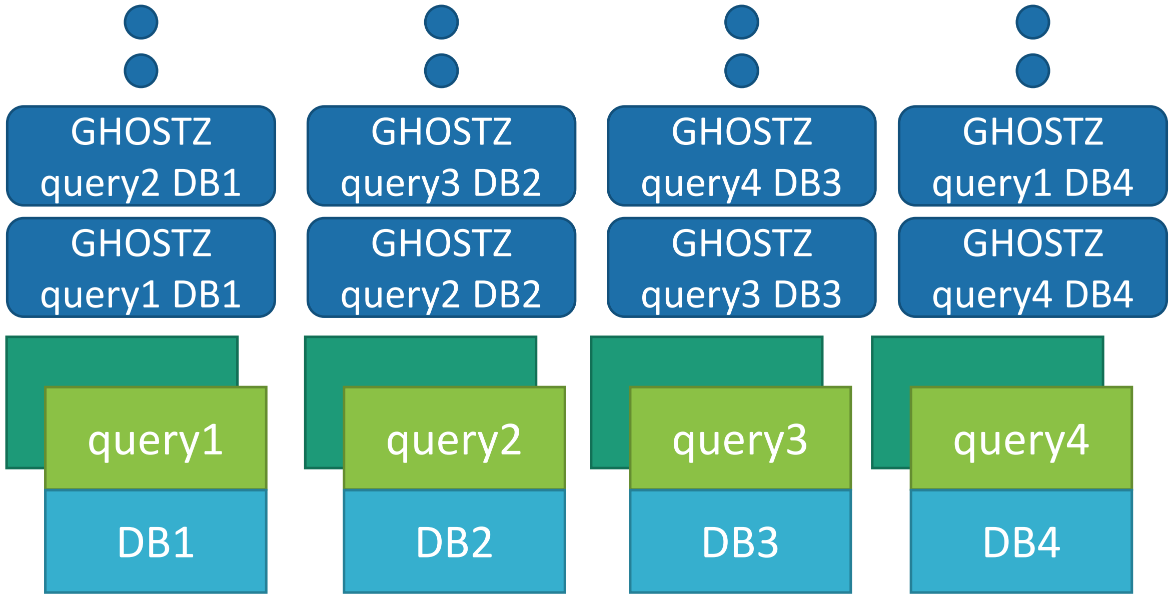
- Co-design of node-local storage and workflow system using data locality
  - No complex MPI codes and support for large data sets
- Data is distributed among node local storages
  - Both reference database and query data distributed
  - Data stored in a remote local storage can be accessed transpatently by Gfarm/BB
- Workflow execution by Pwrake using data locality
  - Execute a compute node where the reference database stored



8.9x speed up

GHOST-MP

Pwrake/Gfarm



GHOSTZ query2 DB1 | GHOSTZ query3 DB2 | GHOSTZ query4 DB3 | GHOSTZ query1 DB4

GHOSTZ query1 DB1 | GHOSTZ query2 DB2 | GHOSTZ query3 DB3 | GHOSTZ query4 DB4

query1 DB1 | query2 DB2 | query3 DB3 | query4 DB4

- Data distribution and input data creation are also executed using a workflow

General framework for scaleble large-scale meta-genome analysis using GPU clusters

Summary

- Node-local flash and persistent memory are a key to narrow performance gap between computing and storage
- Gfarm/BB burst buffer system construct an on-demand file system using node-local storage for burst buffer
- Locality-aware MPI exploits the data access locality for node-local storage for scalable performance
- Prefetching random small files hides the staging overhead for large-scale deep learning
- Gfarm/BB accelerates large-scale metagenome application
- Further exploitation of node-local storage is required