Automatic Topology Mapping on Blue Waters

Juan Galvez
PAID Project

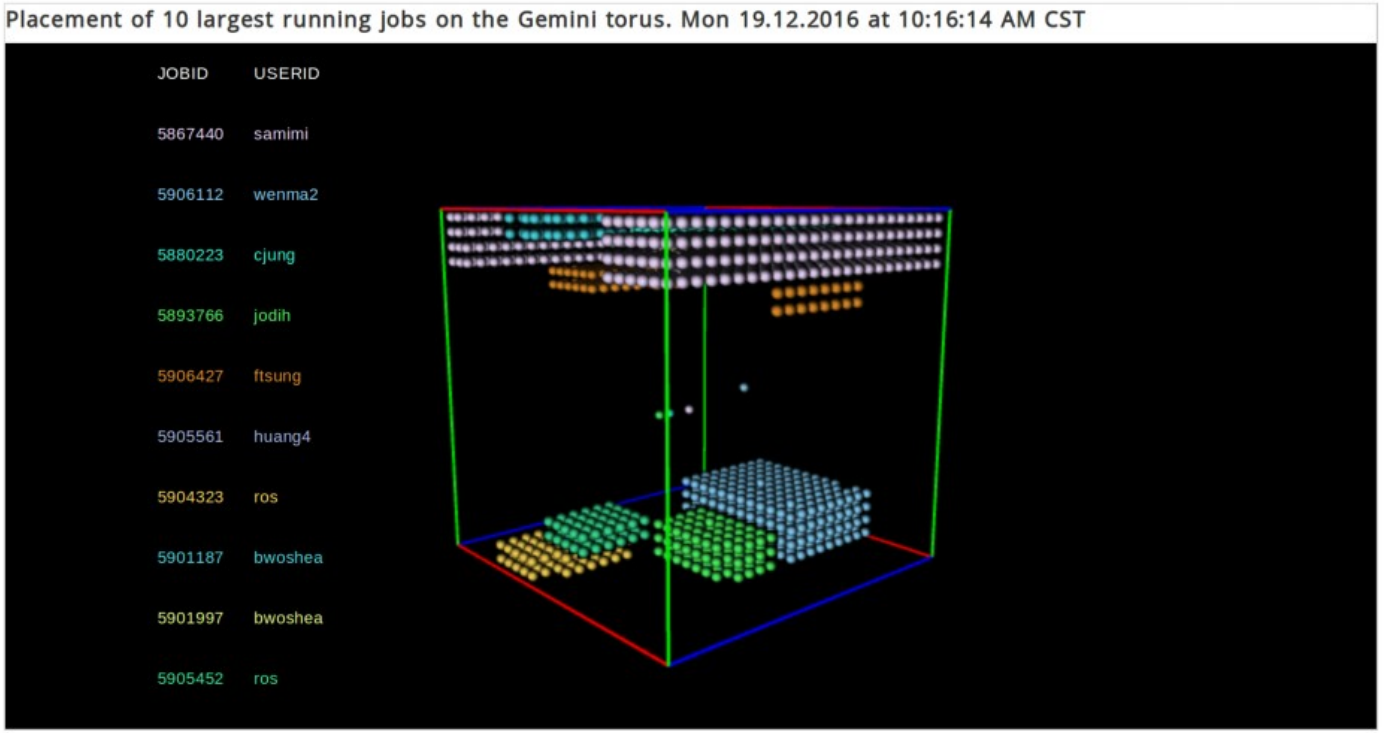
PPL @ UIUC
December 2016

Motivation

- Blue Waters has 3D torus topology of 24x24x24 nodes
- Task mapping refers to assignment of MPI tasks to nodes/processors when launching a job
  - Can have big impact on communication performance of an application
- Default task placement on Blue Waters assigns tasks in rank order to a list of processors
- If a geometry is not specified, scheduler will pick a shape based on available space

Motivation

- Default placement can sometimes be good enough, but won't be for every geometry shape, application

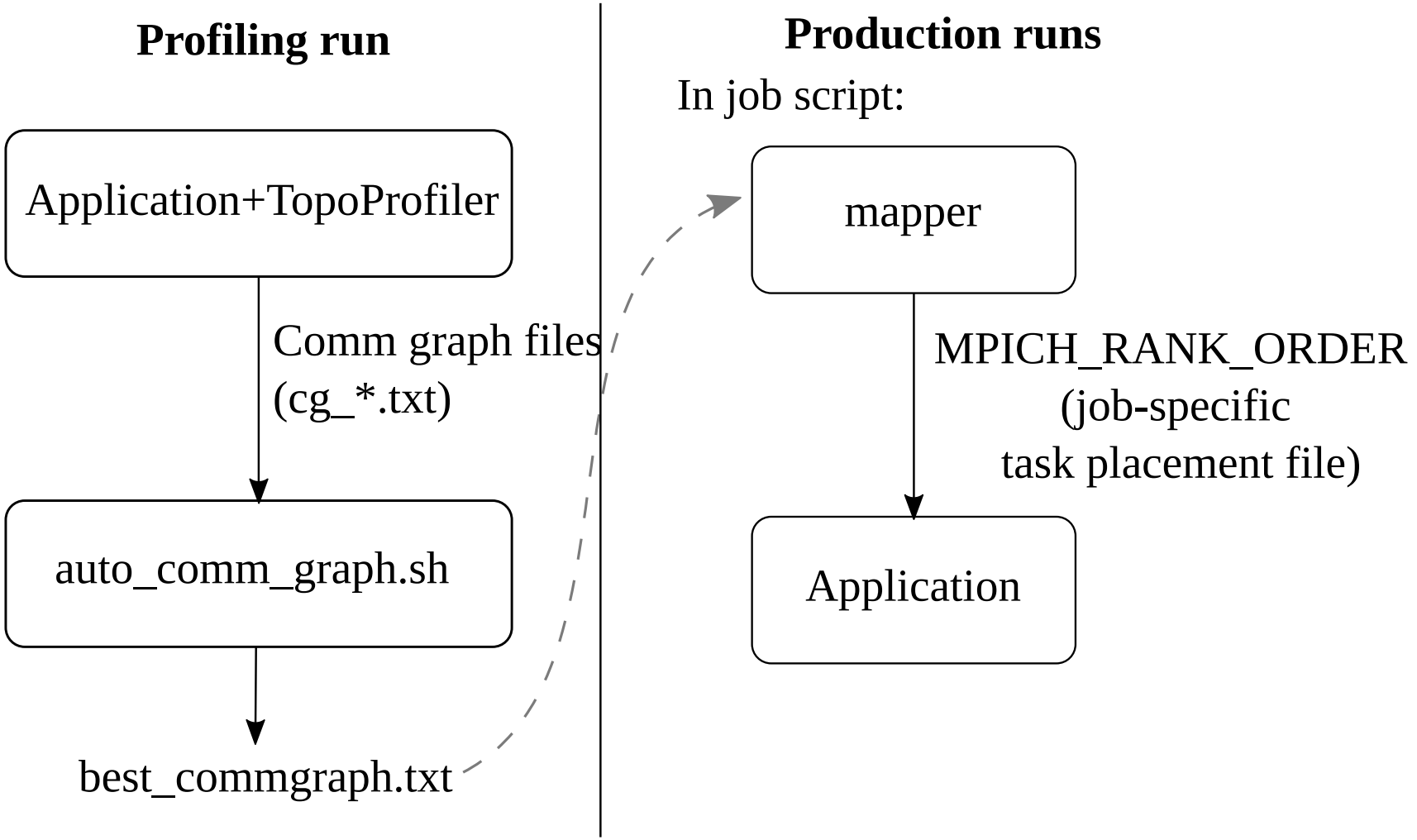


https://bluewaters.ncsa.illinois.edu/torus-viewer

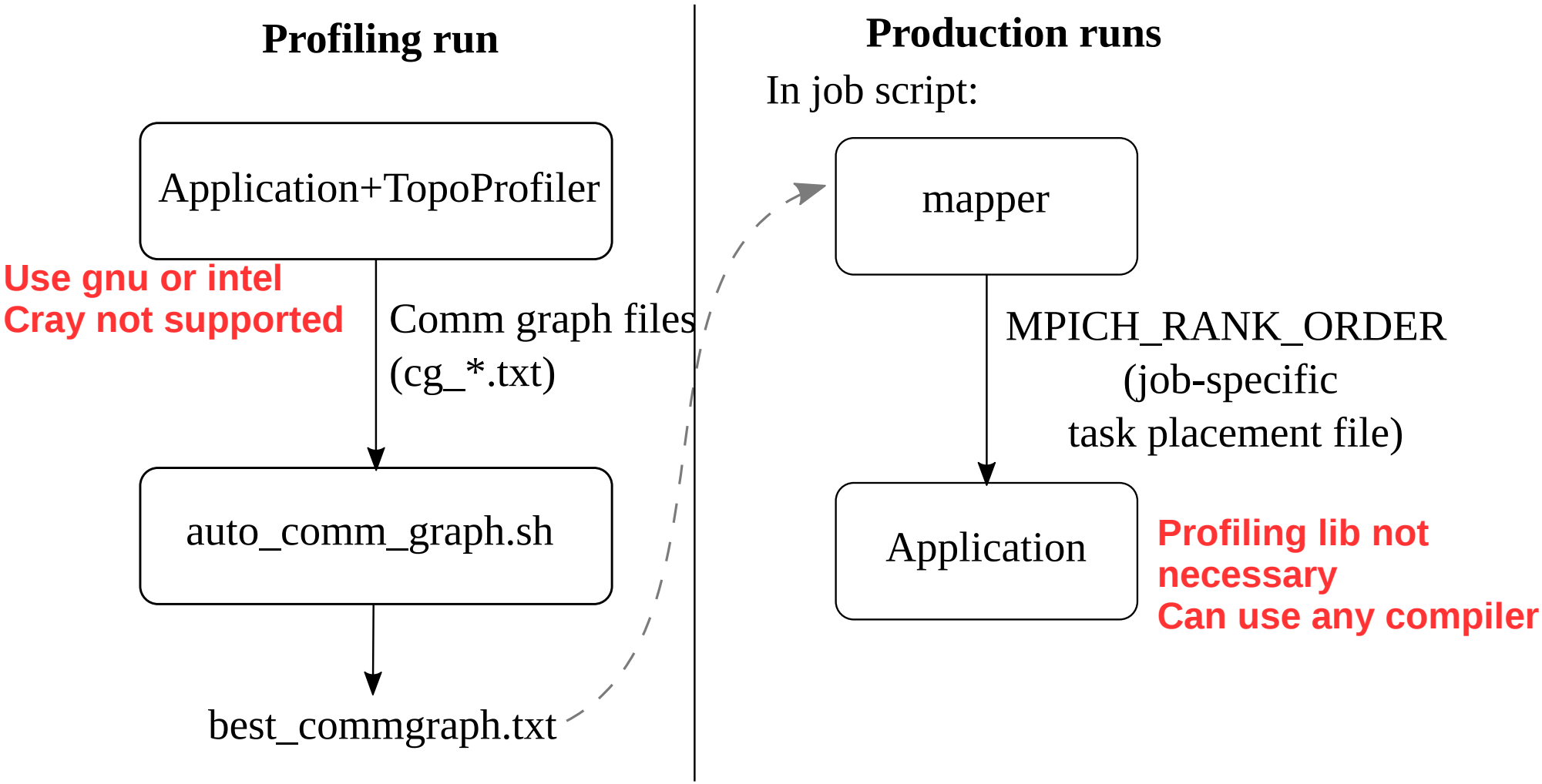
Motivation

- Determining best shape for your application can be expensive (trial and error) process
- Requesting a specific shape is not ideal, because job might take longer to start
  - More efficient to let scheduler place job in available space
- We designed automatic topology-aware task mapping tools to optimize placement for any shape

Overview


Profiling run
Application+TopoProfiler
Comm graph files
(cg_*.txt)
auto_comm_graph.sh
best_commgraph.txt
Production runs
In job script:
mapper
MPICH_RANK_ORDER
(job-specific
task placement file)
Application

Overview


Profiling run
Application+TopoProfiler
Use gnu or intel
Cray not supported
Comm graph files
(cg_*.txt)
auto_comm_graph.sh
best_commgraph.txt
Production runs
In job script:
mapper
MPICH_RANK_ORDER
(job-specific
task placement file)
Application
Profiling lib not
necessary
Can use any compiler

Topomapping module

- **module load topomapping**
- Instructions currently in /sw/bw/topomapping/README.txt

Profiling run

- Load GNU or Intel environment (e.g. module swap PrgEnv-cray PrgEnv-gnu)
- module load topomapping
- module unload darshan
- Build application and link with topoprofiler
  - Add -L$TOPOMAPPING_LIBRARY_PATH -ltopoprofiler -ltmgr (end of library list)
- Use auto_comm_graph.sh script to run profiling job (in $TOPOMAPPING_BINARY_PATH)
  - Copy and edit the script (instructions in script)
- Obtain best_comm_graph.txt at the end

Profiling run tips

- For profiling run, use same size (# ranks) as production runs, and similar application parameters
- Duration/number of iterations can be small
  - As long as representative communication patterns occur in that time

TopoProfiler_sumFile.txt

```
Total Execution Time: 370.4938
Total Messages: 11545739264.0
Total Bytes: 190421553643520.0
Average Messages (rank average):
1409392.0000
Min Messages: 1409392.0000 on Rank: 0
Max Messages: 1409392.0000 on Rank: 0
Average Bytes (rank average):
23244818560.0000
Min Bytes: 23244818560.0000 on Rank: 0
Max Bytes: 23244818560.0000 on Rank: 0
Average Hopbytes (rank average): 2.8672
Min Hopbytes: 1.6979 on Rank: 1672
Max Hopbytes: 6.3622 on Rank: 3040
Average Idle Time (rank average):
167.4324
Min Idle Time: 146.0450 on Rank: 4850
Max Idle Time: 185.6581 on Rank: 891
```

Total run time

Messages

Bytes

$$hopbytes_m = \frac{\sum_n hops(m,n) \times bytes(m,n)}{totalbytes_m}$$

Communication time

TopoProfiler_sumFile.txt

```
Average Point to Point Operation Time (rank average):
128.5341
Min Point to Point Operation Time: 92.2550 on Rank:
4177
Max Point to Point Operation Time: 157.8383 on Rank:
6783
Average Collective Operation Time (rank average):
38.8982
Min Collective Operation Time: 8.5124 on Rank: 2642
Max Collective Operation Time: 72.5520 on Rank: 3593
Average number of Send Calls (rank average):
704696.0000
Min number of Send Calls: 704696.0000 on Rank: 0
Max number of Send Calls: 704696.0000 on Rank: 0
Average number of Recv Calls (rank average):
704696.0000
Min number of Recv Calls: 704696.0000 on Rank: 0
Max number of Recv Calls: 704696.0000 on Rank: 0
Average number of Collective Operations (rank
average): 19317.0000
Min number of Collective Operations: 19317.0000 on
Rank: 0
Max number of Collective Operations: 19317.0000 on
Rank: 0
```

Time in point-to-point comm

Time in collectives

Production runs

- module load topomapping
- Add following lines in job script (before application):
  - aprun -n NUM_RANKS mapper time_limit commgraph.txt
  - export MPICH_RANK_REORDER_METHOD=3
- Mapper binary launched inside batch job, will read the cg file and automatically generate placement file for the given shape
- Topoprofiler not needed

Conclusion

- Good results in several applications tested
  - Improvement is highly dependent on application. For example:
    - Up to ~2.5x improvement with MILC
    - Up to ~70% improvement with Qbox
  - Profiler and mapper designed to be automatic (knobs hidden from user)
    - There is possibility of scenarios where they don't make best decision
- Welcome feedback to see how it performs and keep improving the tool