Evaluating New Communication Models in the Nek5000 Code for Exascale

The Cost of Synchronizing a Billion Processes

Ivy Bo Peng, Stefano Markidis, Erwin Laure
KTH Royal Institute of Technology

Gordon Moore: what do you think is the limit of Moore’s Law?
Stephen Hawking: the ultimate limit is the speed of light and the size of a single atom
Outline

• A Billion-Way Parallelism at Exascale.
• The Cost of Synchronizing Imbalanced Processes
• A LogP Monte Carlo Simulator
• Synchronizing a billion processes
  – Communication- and imbalance–dominated synchronization.
  – The impact of the number of cores per node
• Conclusions
A Billion-Way Parallelism at Exascale

- We are now million-way parallelism:
  - Thianhe-2 (33 PF) = 3,120,000 cores
  - Sequoia (17 PF) = 1,572,864 cores

- Nodes are becoming fatter but not faster
  - Exascale supercomputers are expected to have many cores per node\(^\text{[1]}\)
  - Performance growth mainly comes from the increase of core number not the performance of a single core

- Network latency lags behind bandwidth
  - Hide it if we cannot improve it
  - Hide more when problem scales up

Process Imbalance

• Process imbalance is statistically inevitable on billions of processes.

• Two major sources:
  – OS and architecture noises
  – Load imbalance

• A single slow process could impact the global performance:
  – Blocking collective operations
  – Non-blocking point-to-point operations

Synchronization in Message Passing Systems

• Synchronization is done through point-to-point Communications.
• Several algorithms for synchronization with different communication cost.
• Process imbalance causes processes to reach the synchronization point at different time

\[ \text{SYNC} = \text{COMM} + \text{IMB} \]

Can \( \text{SYNC} \leq \text{COMM} + \text{IMB} \)?
Imbalance Absorption with Latency Hiding

Process imbalance can be hidden (absorbed) by communication.

An example of full imbalance absorption using a linear barrier on three processes.

Different synchronization algorithms have different absorption property. Algorithms with higher absorption rates not necessarily have higher communication costs.
The LogP Model[2]

- We use the LogP model to evaluate communication cost:
  - \( L \): the largest latency between any two processes (approx. 1-10 us on modern network)
  - \( o \): CPU overhead in message transmission (on snd and recv ops)
  - \( P \): number of nodes
  - \( g \): gap between two messages
- We added:
  - \( N \): number of processes per node
- We assume instantaneous synchronization on the same node (\( \approx 100 \) ns vs \( \approx 5 \) us)

Cost for one message is \( L + 2o \)
Sync. algorithms need \( \log(P) \) communication cost or more

The Process Imbalance Model

- We use two random distribution functions for modeling imbalance:
  - Normal
  - Exponential
- The imbalance time scale is characterized by standard deviation $\sigma$
The LogP Monte Carlo Simulator

- These quantities are calculated as expected values:
  - $\text{Synch} = \text{time difference between first enter and last exit from barrier}$
  - $\text{Imbalance} = \text{max. time difference reaching the sync point}$
- Effective Imbalance = Synch – Comm
- We study representative barrier algorithms in each complexity category:
  - MCS Tree
  - Tournament 2, 4, 16-way
  - Dissemination

Ivy Bo Peng et al. "The cost of synchronizing imbalanced processes in message passing systems" submitted to CLUSTER
Synchronizing a Billion Processes

\[ \sigma = L + 2o \]

\[ 2^{20} \text{ nodes, } 1024 \text{ procs per node, normal imbalance distribution with } \sigma = L + 2o = 5.5 \text{ us} \]
Synchronizing a Billion Processes

$$\sigma = 4(L + 2\alpha)$$

$2^{20}$ nodes, 1024 procs per node, normal imbalance distribution with $\sigma = 3(L + 2\alpha) = 16.5$ us
Imbalance- v.s Communication-Dominated Synchronization

Asymptotic upper bound for impact of imbalance > impact of communication $\approx 1.7 \frac{\sigma}{(L+2o)}$

Dissemination barrier, 1024 procs per node, normal imbalance distribution with $\sigma = L + 2o = 5.5$ us
Impact of Number of Cores per Node

1 billion processes, normal imbalance distribution with $\sigma = L + 2\sigma = 5.5$ µs

**Synchronizing One Billion Processes**

- **MCS Tree**
  - 64 cores: 43.7 µs
  - 1024 cores: 48.5 µs
  - 16384 cores: 52.7 µs

- **Tour. 4-way**
  - 64 cores: 204.0 µs
  - 1024 cores: 170.0 µs
  - 16384 cores: 136.0 µs

- **Tour. 16-way**
  - 64 cores: 318.0 µs
  - 1024 cores: 265.0 µs
  - 16384 cores: 212.0 µs

- **Dissemination**
  - 64 cores: 67.1 µs
  - 1024 cores: 67.1 µs
  - 16384 cores: 67.1 µs
Conclusions

• Process Imbalance with time scales greater than the time for sending one message will impact synchronization at exascale.

• Certain synchronization algorithms (with not optimal communication cost) allow to hide imbalance with communication.

• Larger number of cores per node increases the impact of process imbalance.

• Selection of the optimal synchronization algorithms should not only consider communication cost but also imbalance. absorption.
Thanks!