Improving Performance and Energy Efficiency of Matrix Multiplication via Pipeline Broadcast

UNIVERSITY OF CALIFORNIA, RIVERSIDE

Li Tan[1], Longxiang Chen[1], Zizhong Chen[1], Ziliang Zong[2], Rong Ge[3], Dong Li[4]

Summary

Goal: Improving communication performance of distributed matrix multiplication to achieve energy efficiency

- Devise a high performance communication scheme
  - Fully exploiting network bandwidth of distributed matrix multiplication via non-blocking pipeline broadcast with tuned chunk size
- Model and quantify the communication time complexity of binomial tree broadcast and pipeline broadcast
  - Analyzing communication slack in two types of pipeline broadcast
- Evaluate all three types of communication schemes
  - On a 64-core power-aware cluster
  - Non-blocking pipeline broadcast with tuned chunk size is able to gain performance and energy savings

Background

- Matrix multiplication is a core operation of most numerical linear algebra algorithms
  - LU, Cholesky, and QR factorizations
  - Provided by ScaLAPACK and DPLASMA, etc.
- Distributed matrix multiplication algorithm (DIMMA)
  - Distribute matrix elements into a process grid
  - Broadcast local matrices in row-/column-wise
  - Perform local matrix multiplication

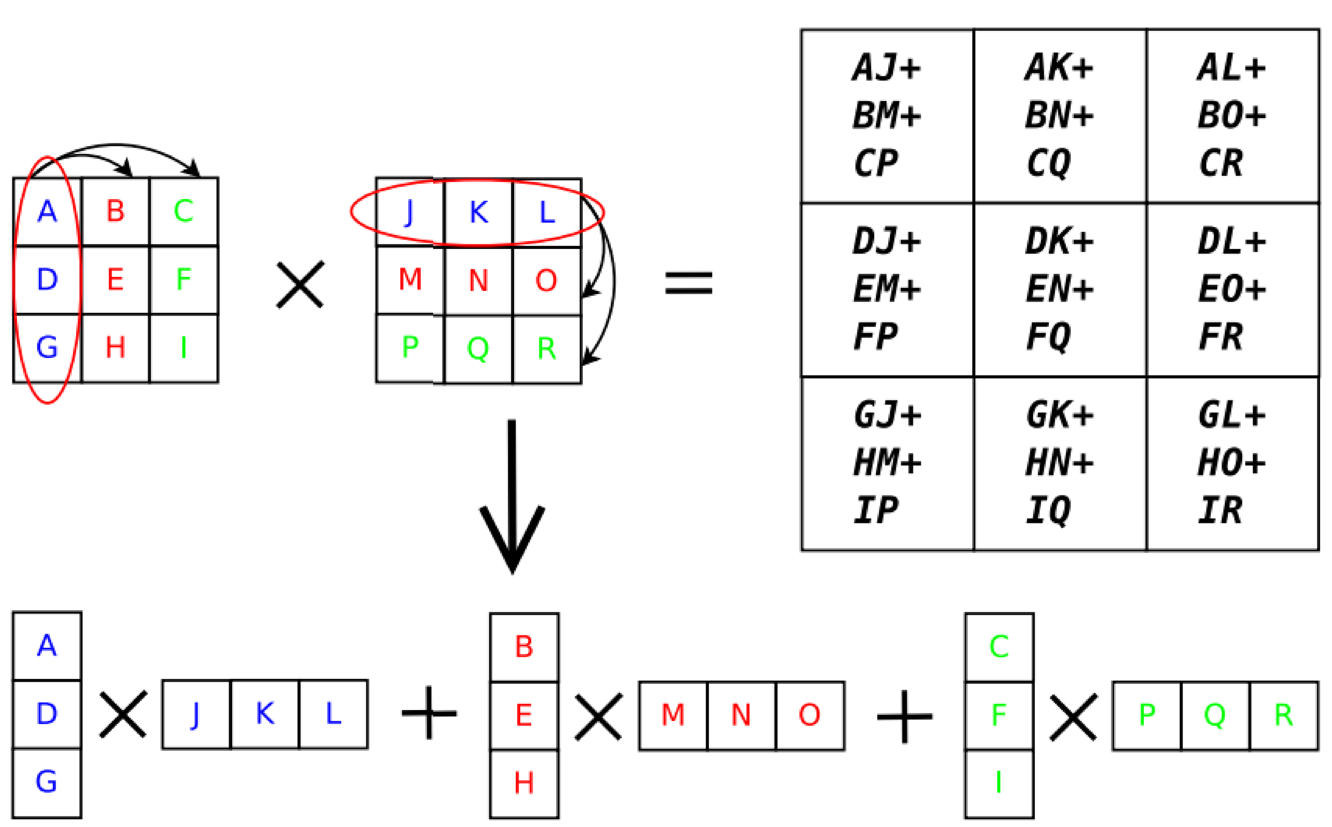
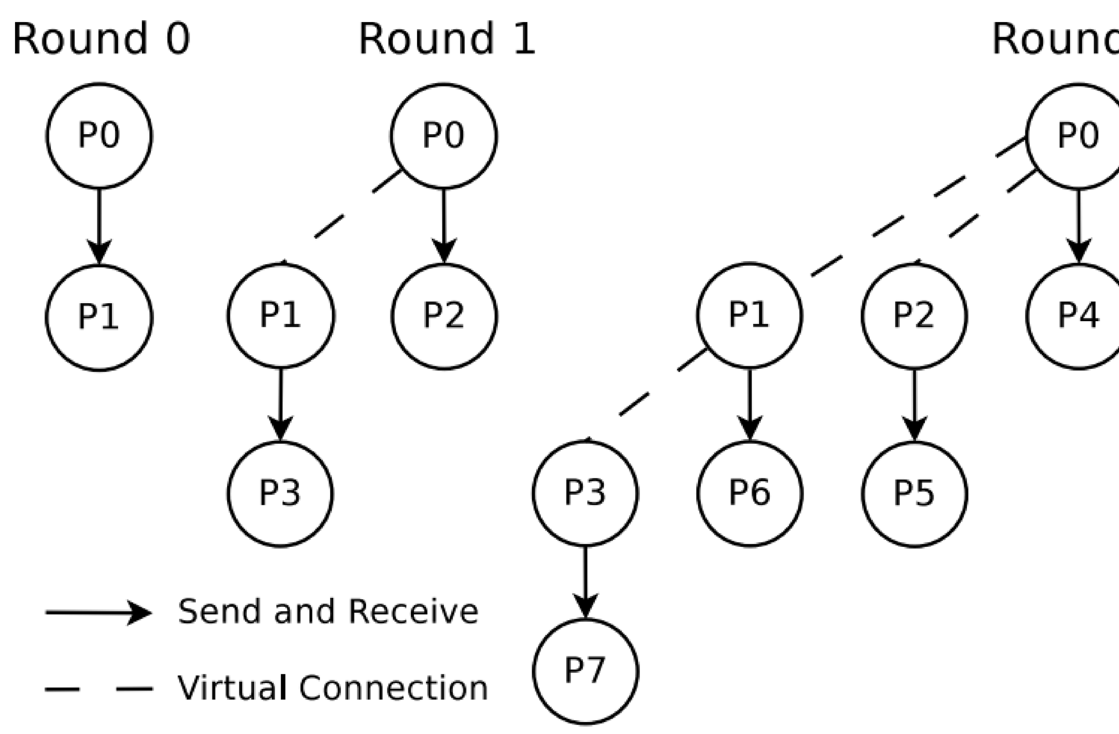


Fig. 1. A Distributed Matrix Multiplication Algorithm with a Global View.

Binomial Tree and Pipeline Broadcast Algorithms



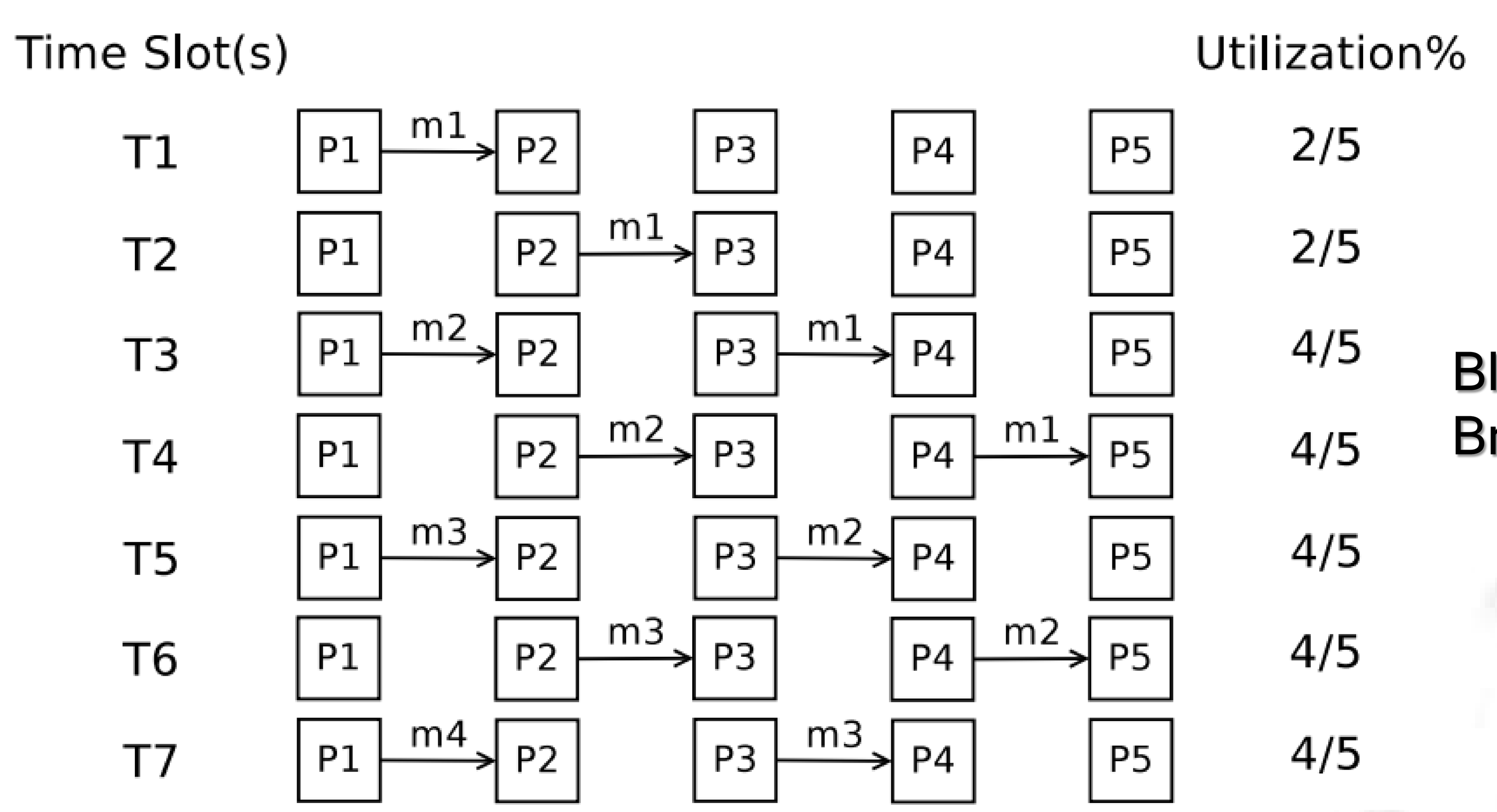
Binomial Tree Broadcast

- Binomial Tree Broadcast
  - Round 0, P0 sends to P1
  - Round j (j > 0), Pi (i <= j) sends to the subsequent one
- Communication time complexity
  - Binomial Tree Broadcast

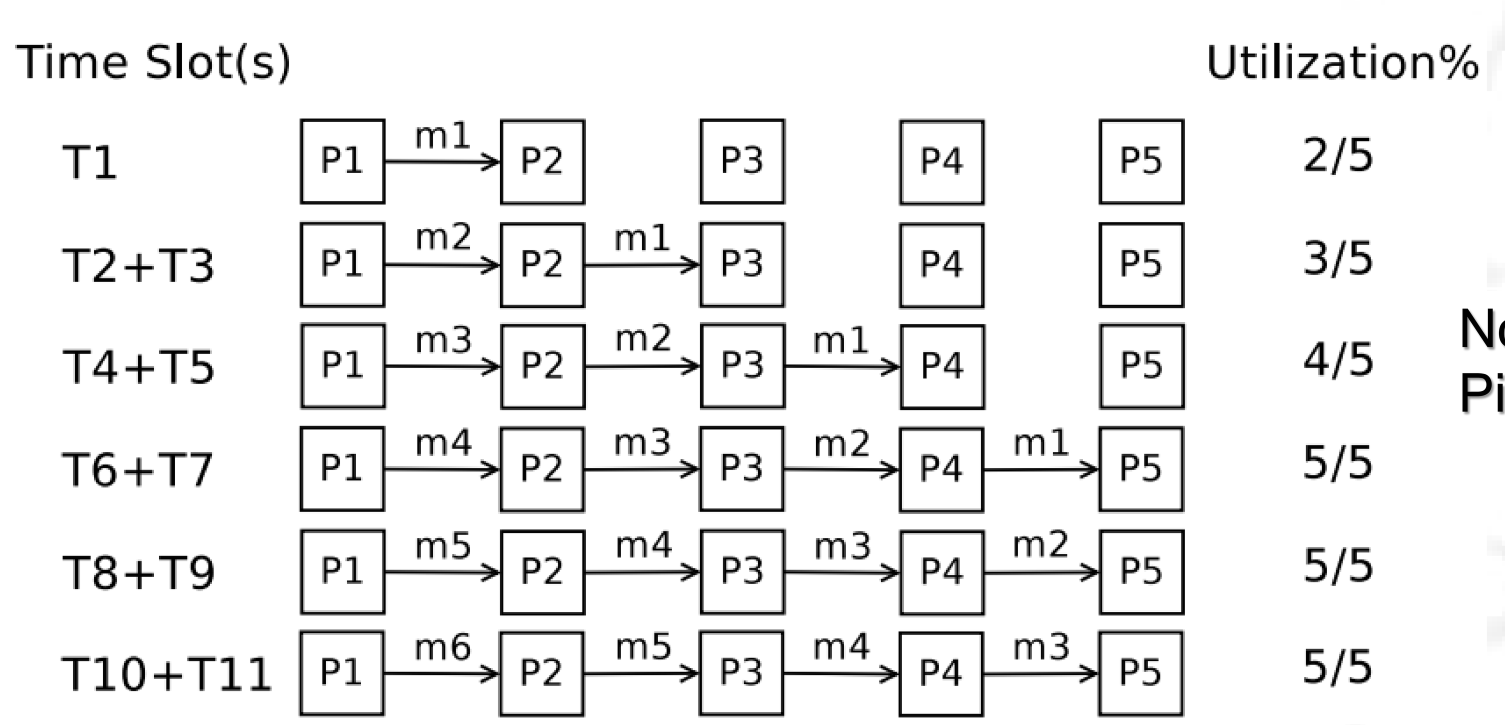
$$T_B = (T_s + \frac{S_{msg}}{BD}) \times logP$$

  - Pipeline Broadcast

$$T_P = (T_s + \frac{S_{msg}/n}{BD}) \times (n + P - 1)$$



Blocking Pipeline Broadcast



Non-blocking Pipeline Broadcast

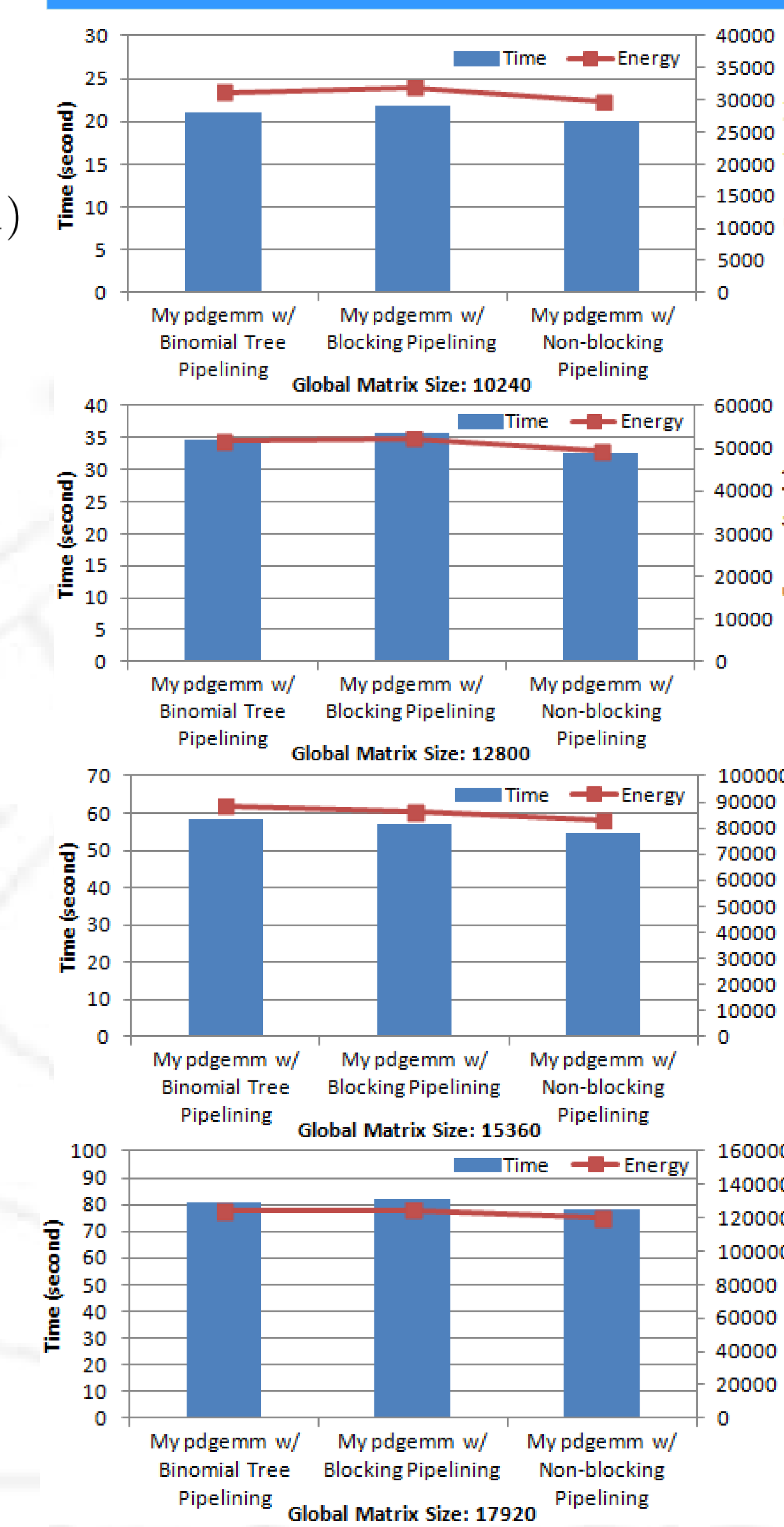
Type of Pipeline Broadcast	Pipeline Is Full?	Max Pipeline Time Latency[1]	Average Utilization[2]
Blocking, Odd PL	No	$2(\lfloor \frac{PL}{2} \rfloor - 1)$	$\sim 1 - \frac{1}{PL}$
Blocking, Even PL	Yes[3]	$PL - 2$	$\sim 1 - \frac{1}{PL}$
Non-blocking, Odd PL	Yes	$2PL - 5$	$\sim 100\%$
Non-blocking, Even PL	Yes	$2PL - 5$	$\sim 100\%$

TABLE III. PIPELINE BROADCAST EFFICIENCY COMPARISON

Implementation

- Rewriting the pdgemm() routine provided by ScaLAPACK
  - Employing non-blocking pipeline broadcast for comm.
  - Tuning the chunk size of pipeline broadcast
  - Call the dgemm() routine provided by ATLAS for comp.
  - The same interface & results as ScaLAPACK pdgemm

Performance Gain and Energy Savings



- **Performance Gain**
  - Non-blocking vs. Binomial: 6.5%
  - Non-blocking vs. Blocking: 8.4%
- **Energy Savings**
  - Non-blocking vs. Binomial: 6.1%
  - Non-blocking vs. Blocking: 6.9%

Acknowledgements

- Special thanks to:
  - The HPCL Lab at the Marquette University
  - National Science Foundation Grant #CNS-1118043, #CNS-1116691, #CNS-1304969

1. University of California, Riverside, CA, USA
2. Texas State University-San Marcos, TX, USA
3. Marquette University, WI, USA
4. Oak Ridge National Laboratory, TN, USA