

Extending Distributed Functionality in Phylanx

Phylanx Year 3 Meeting

Maxwell Reeser

November 7, 2019

Division of Computer Science and Engineering
School of Electrical Engineering and Computer Science
Louisiana State University

- Map operations
 - No Data Dependencies
- Distributed Data Structures
 - distributed_object
 - UPC++
 - distributed_vector/matrix
 - Annotations
- Distributed Primitives
- Tiling testing
- Tiling Optimizer

- Uses same type of algorithm as other dot_d primitives
- Iterates through all tiles of RHS
 - Performs multiplication if intersection detected
 - Column width of tiles important
- Result matrices may be large

Cannon's Algorithm

- Matrix Matrix multiplication algorithm
- Different Communication Pattern from dot_d

Cannon's Algorithm

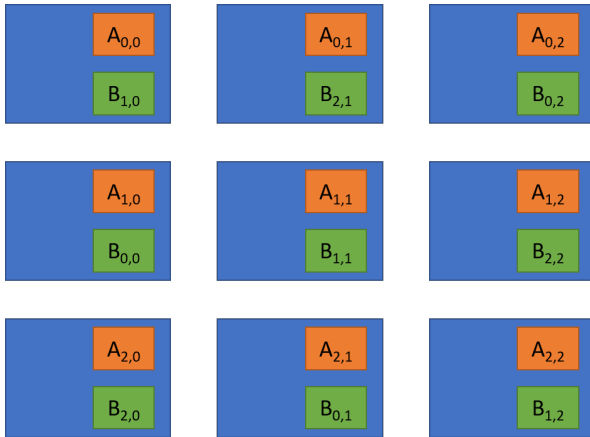


Figure 1: Starting Point

Cannon's Algorithm

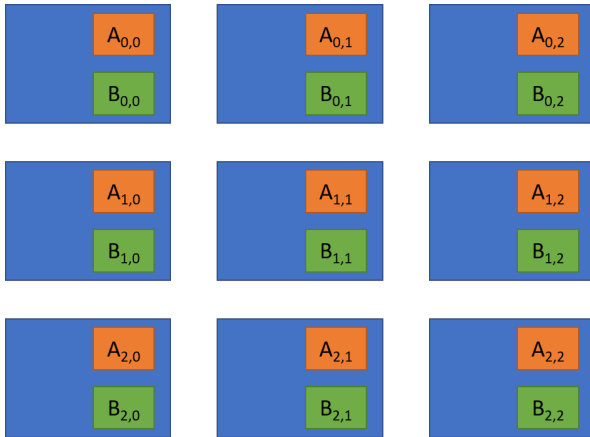


Figure 2: Alignment

Cannon's Algorithm

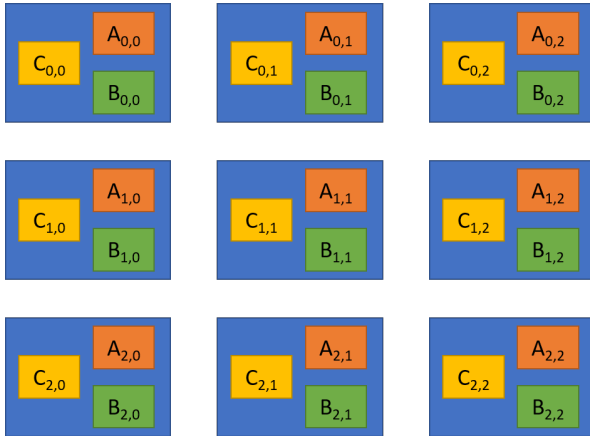


Figure 3: Multiply Local Values

Cannon's Algorithm

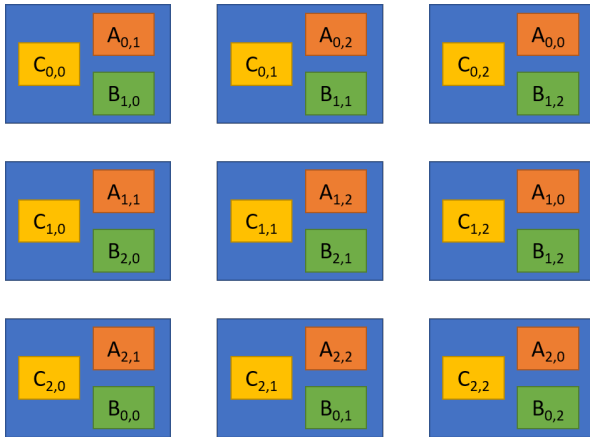


Figure 5: Multiply Local Values

Cannon's Algorithm

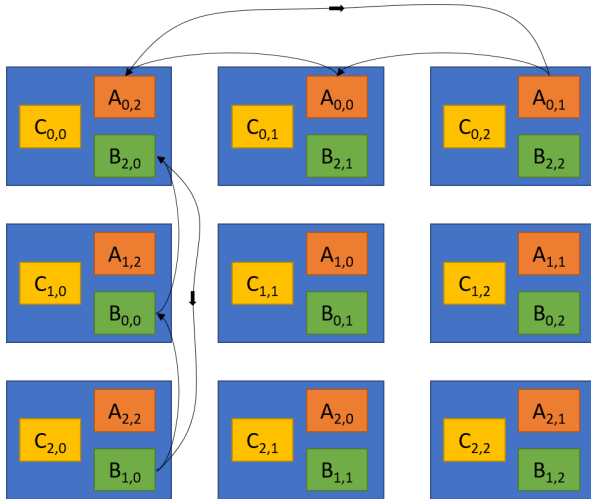


Figure 6: Shift Data

Cannon's Algorithm

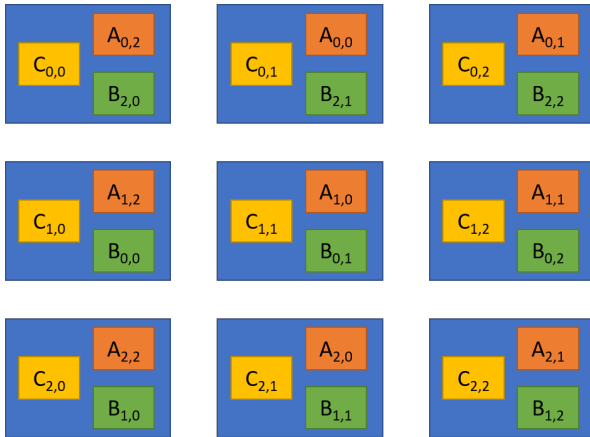


Figure 7: Multiply Local Values

- Currently no Distributed Data Loading
 - Manual specification of data in PhySL strings
- Bottleneck to serious testing
- Simple Fix
 - Base off existing CSV read primitive
 - Use filename as custom basename for finding participating localities

Questions?