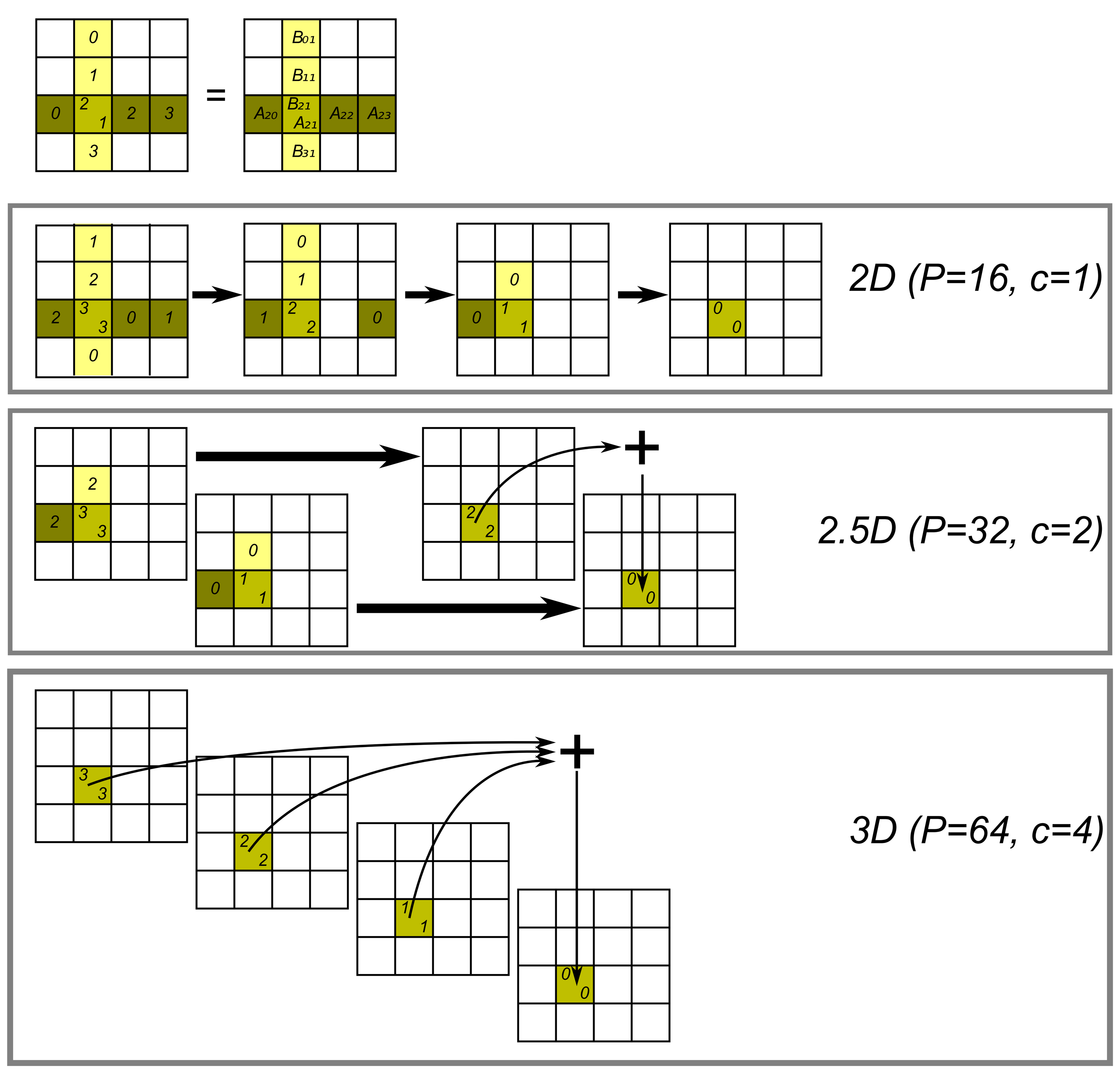
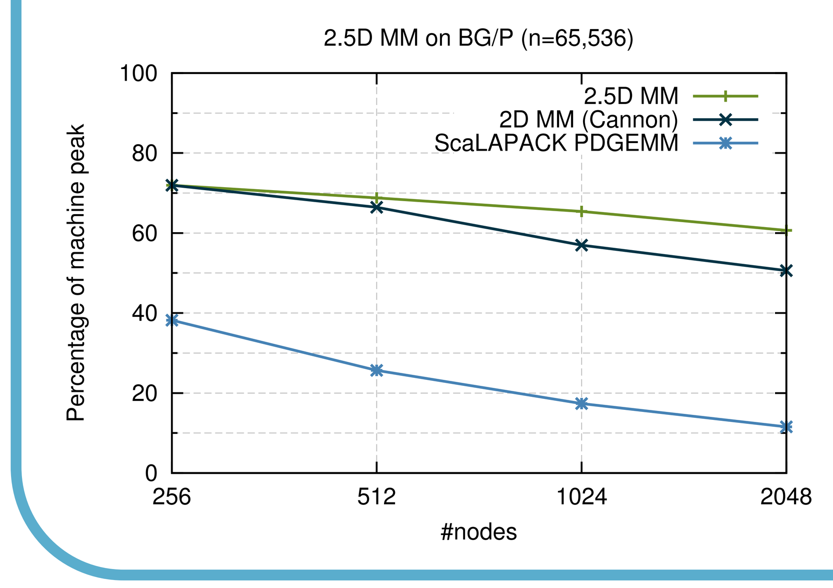
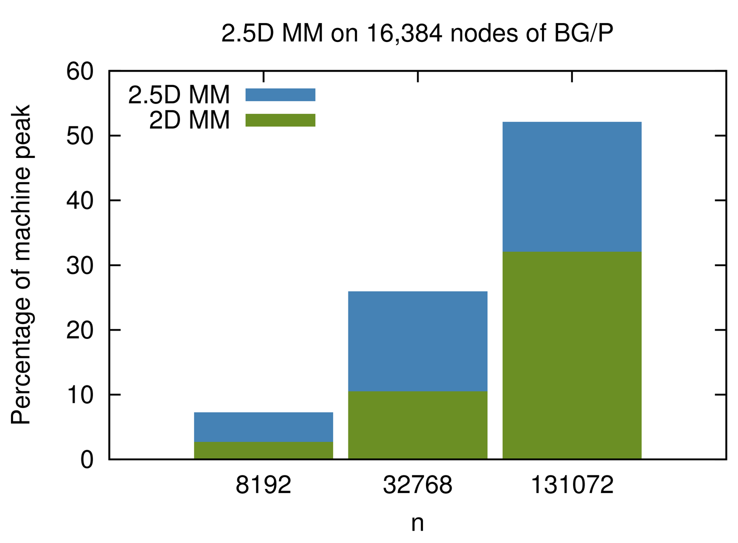
2.5D parallel algorithms for dense linear algebra

Edgar Solomonik and James Demmel
solomon@cs.berkeley.edu

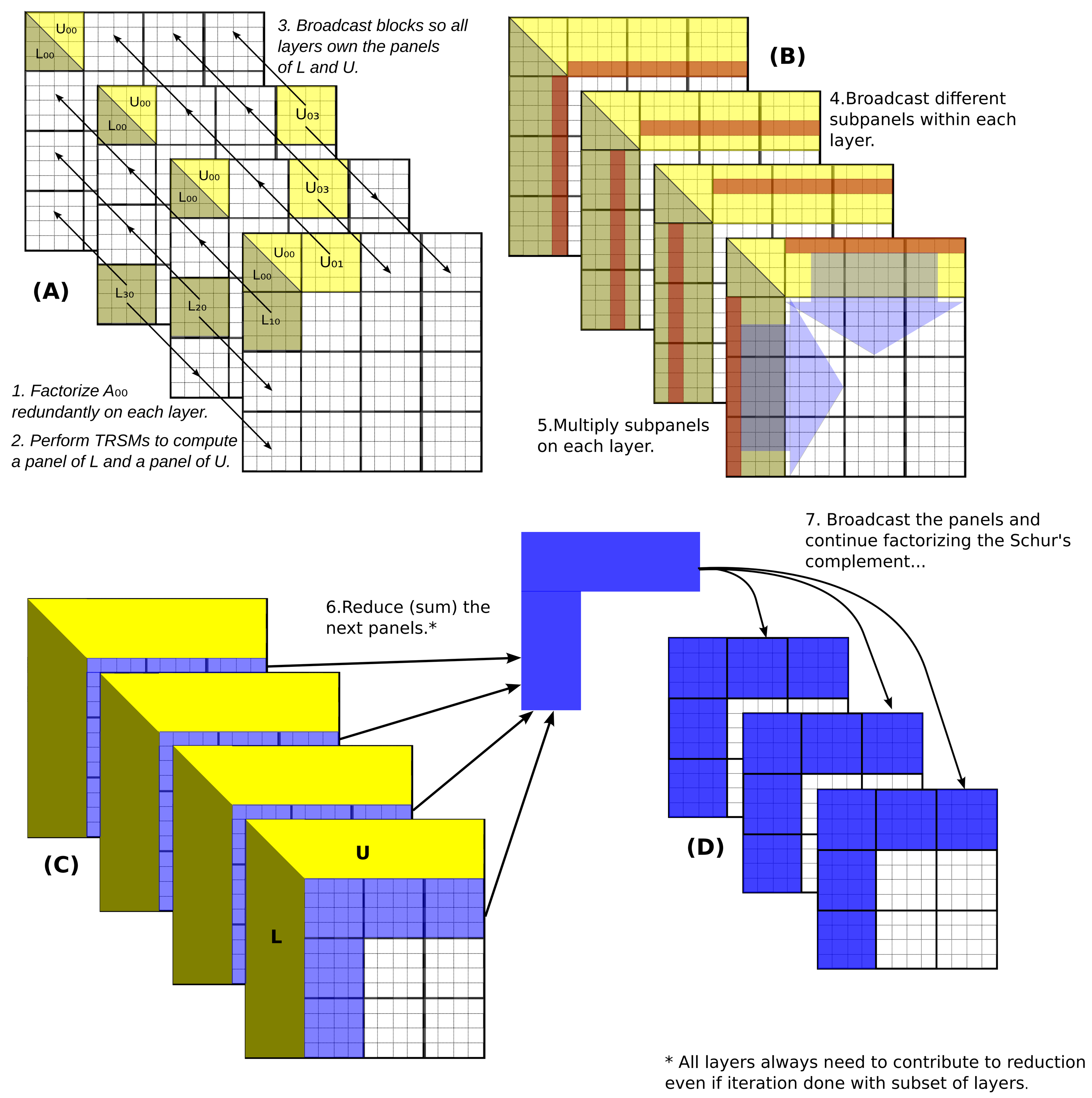
2.5D matrix multiplication

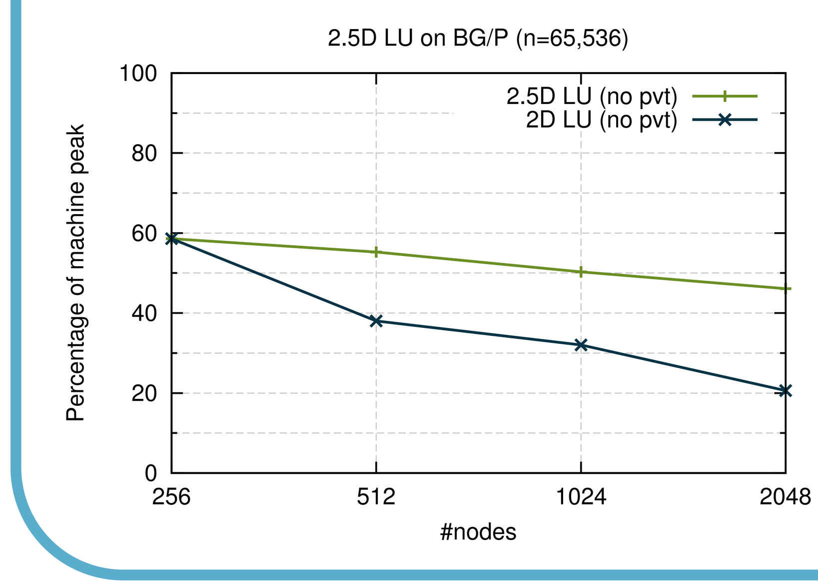


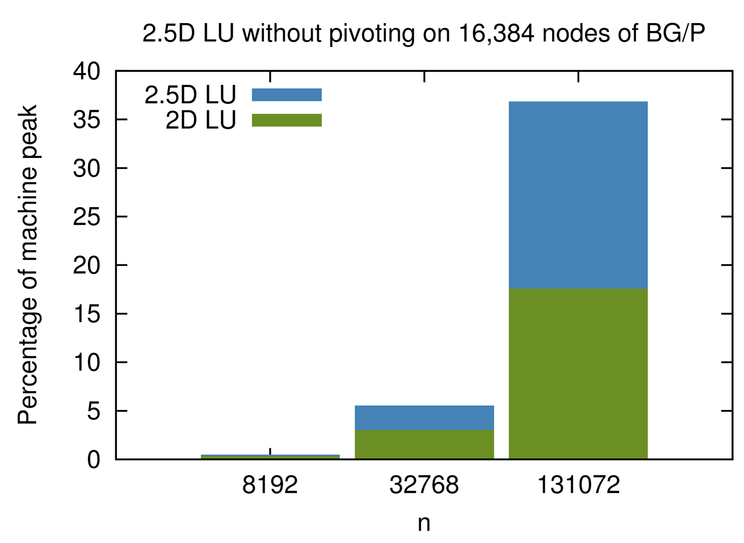




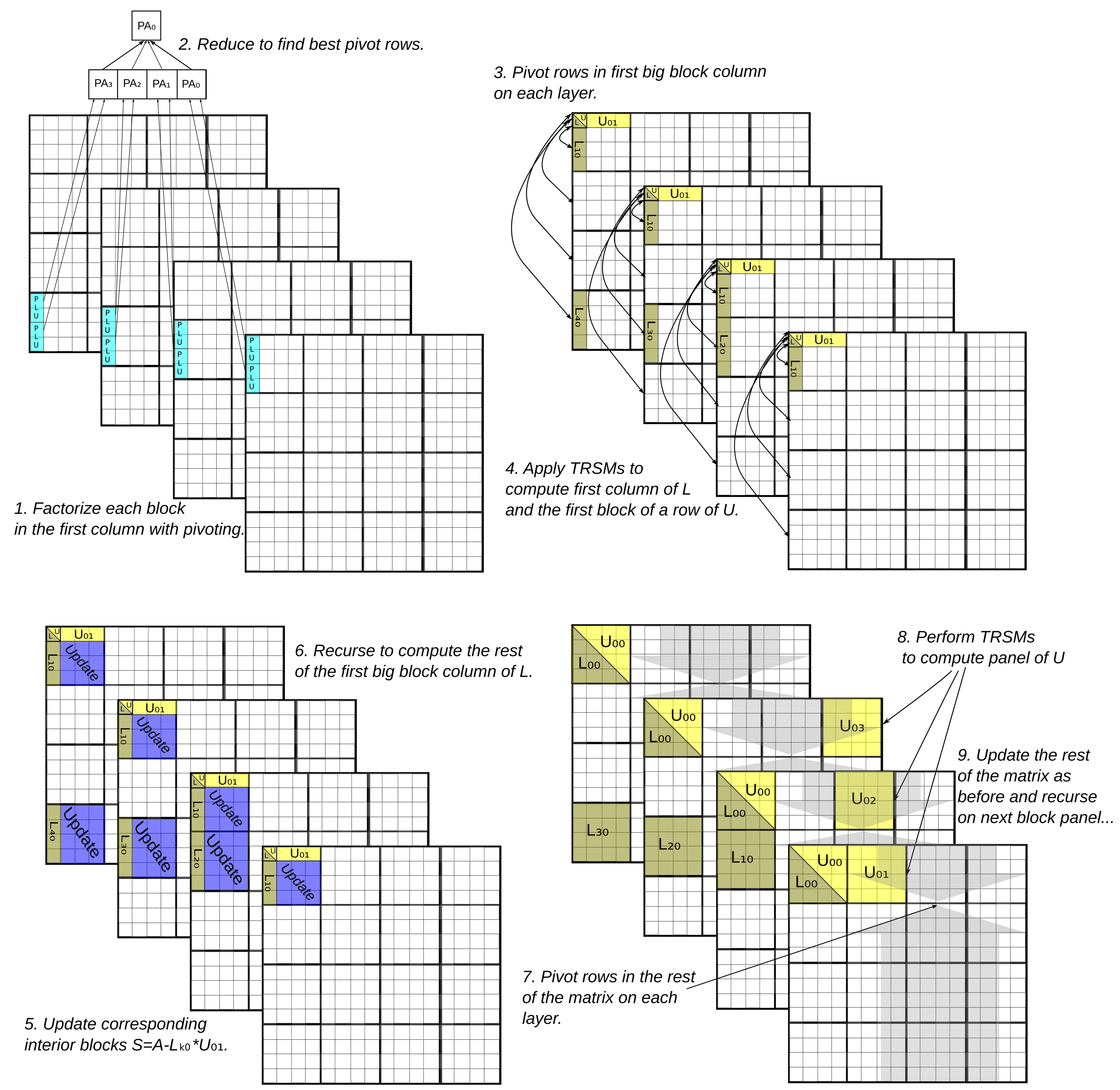
2.5D LU factorization

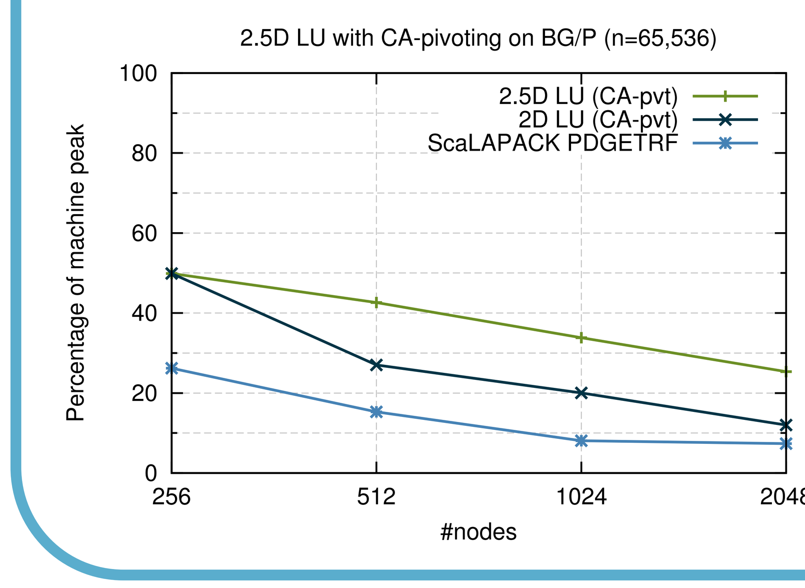


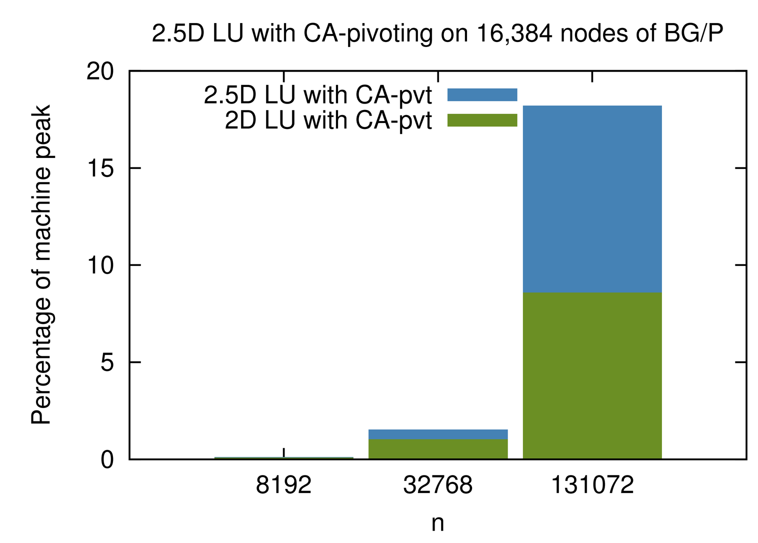




2.5D LU with pivoting







Reduced communication complexity

2.5D algorithms replicate matrices to reduce communication.

- Memory usage: $\uparrow O(c)$
- Words communicated: $\downarrow O(\sqrt{c})$
- Messages sent: $\downarrow O(c^{3/2})$ for MM and $\uparrow O(\sqrt{c})$ for LU

All communication costs are theoretically optimal according to lower bounds. 2D algorithms (standard in ScaLAPACK) are generalized by 2.5D algorithms ($c = 1$).

Network topology awareness

Replication of matrices adds a third dimension to the logical processor grid. 2.5D algorithms parameterize this layout according to physical network topology.

- Network contention reduced or eliminated
- Multicasts and reductions performed along torus edges
- More network bandwidth saturated

These mappings are suitable for torus network topologies. Recursive layouts are probably better for tree or switched network topologies.

Future directions

Algorithmic challenges:

- Nested 2D/2.5D MM tensor contraction algorithms
- Resolution of dependency challenges in 2.5D Householder QR
- Better theoretical models for network topologies

Technical challenges:

- Efficient abstraction of distributed matrix layouts
- Automation of topology mapping and algorithm synthesis