Hiding Communication in Scientific Applications with Graph-based Execution

Pietro Cicotti[1], Xiaoye S. Li[2], Scott B. Baden[1]

1 University of California, San Diego

2 Lawrence Berkeley National Lab

Motivation

Architectural Trend

✓ Large number of processing elements

✓ Heterogeneous computing resources with accelerators

Challenges

X Expose high degree of parallelism

X Scale to thousands of cores

X Hide the cost of communication

Proposed Solution

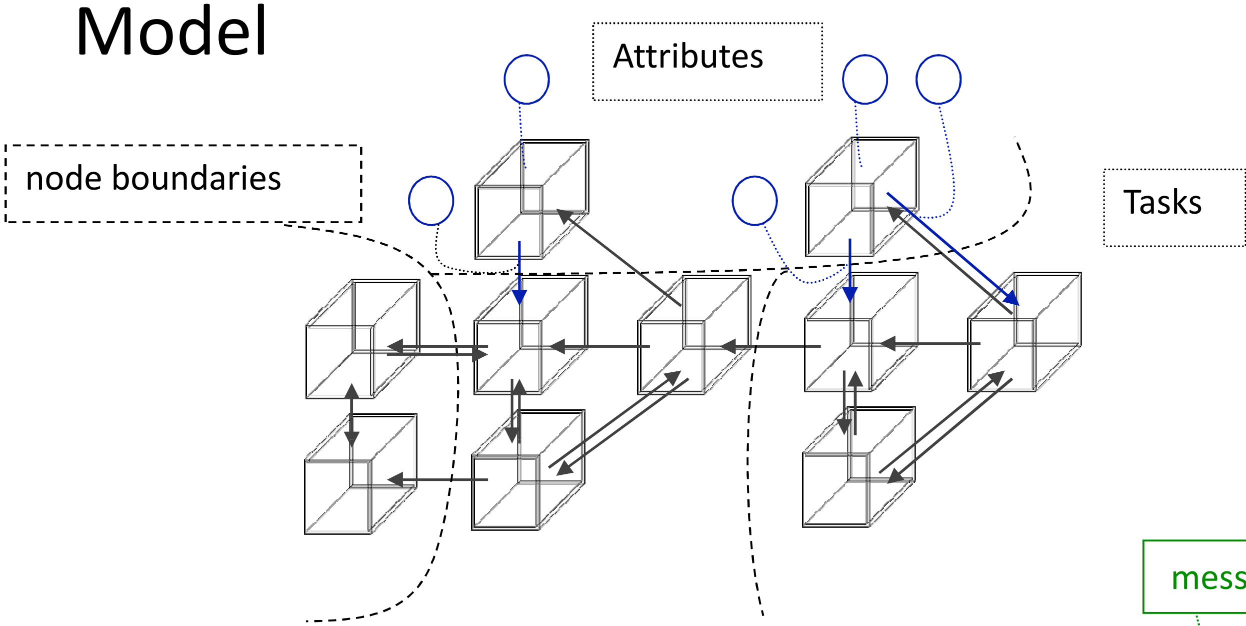
- Decompose work into tasks
- Express dependencies explicitly
- Schedule tasks dynamically
- Tarragon
  - C++ library
  - API →Create graph
  - Runtime system →Execute graph

Tarragon model

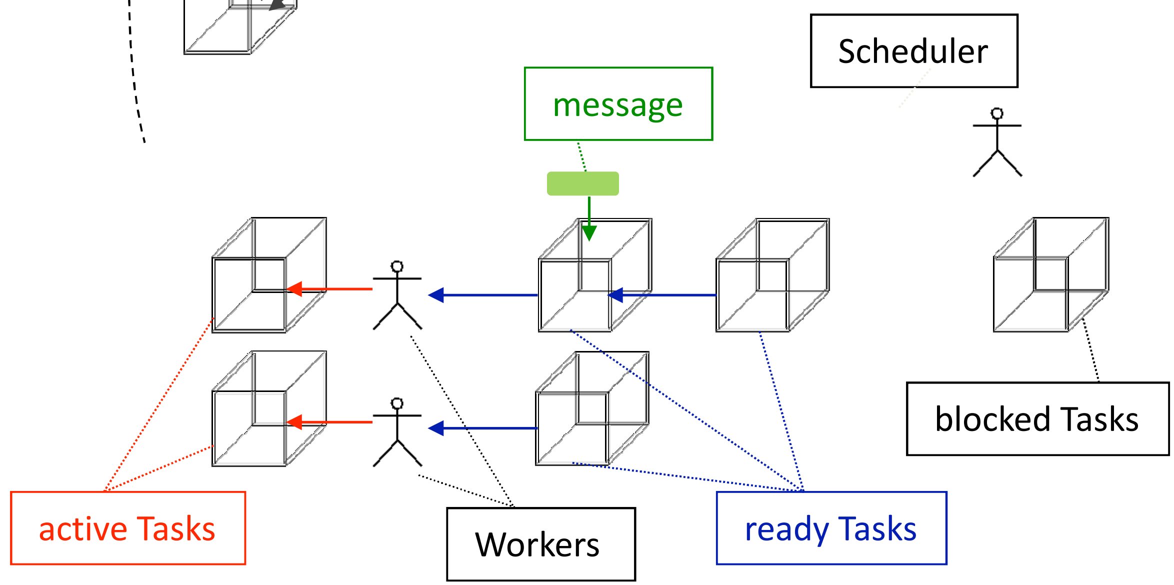
- Graph-based programming model
  - User-defined data-flow semantics
    - Tasks can execute concurrently
    - Communication transfers data and control
  - Tasks are virtual processes
  - One-sided communication
    - Asynchronous communication

Tarragon

Model



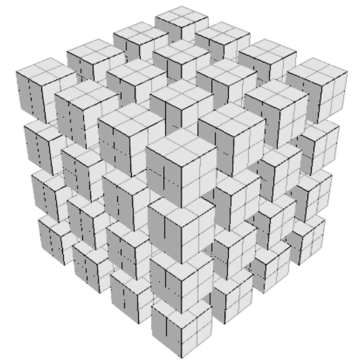
SW architecture



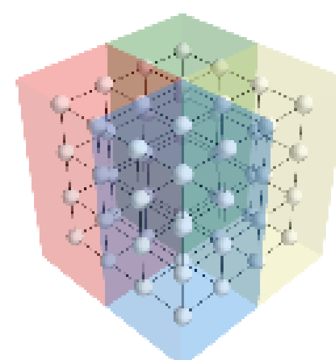
Applications

- **Stencil computation**
- *Matrix multiply*
- Smith-Waterman
- ***Parallel Sparse LU Factorization***

Jacobi



- Jacobi 3D
  - 7-point stencil operation
    - e.g. Poisson equation in 3D
  - execute: exchange ghosts cells and relaxation
  - 3D tasks grid
    - Over-decomposition
    - Nearest neighbor connectivity

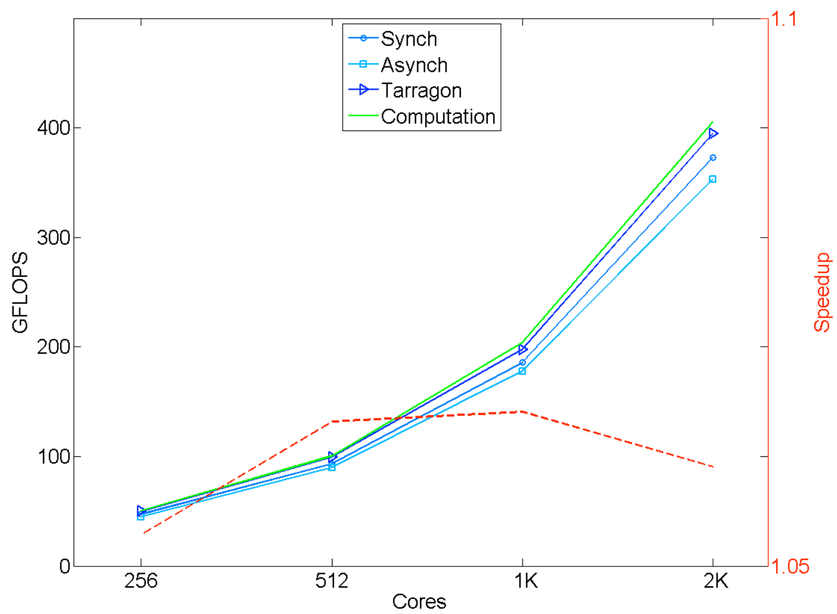


Performance results

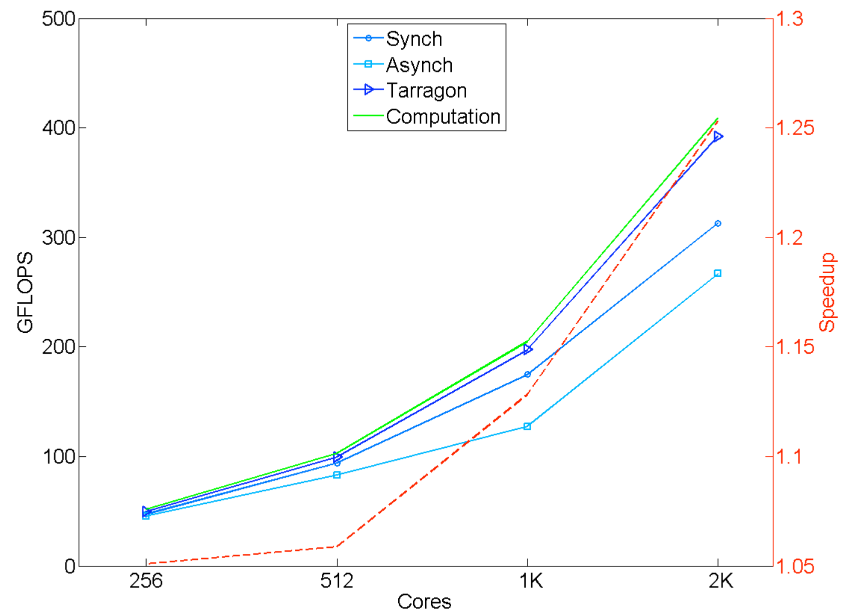
- Abe@NCSA
  - Dell Intel 64 Cluster
  - Dual Clovertown (2x4) + Infiniband

http://www.ncsa.illinois.edu/UserInfo/Resources/Hardware/Intel64Cluster/TechSummary/

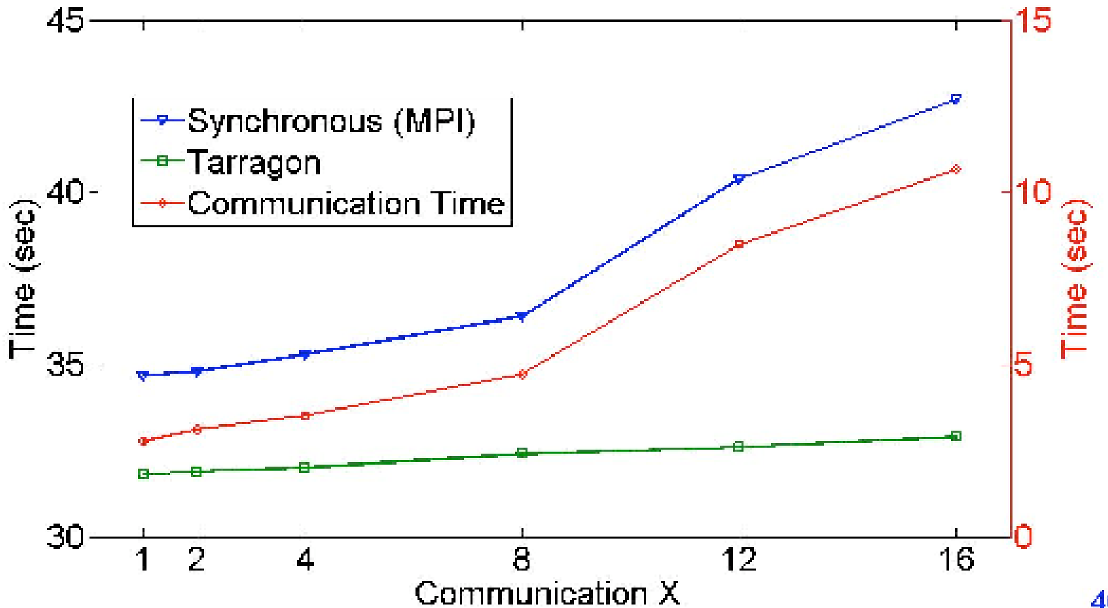
Weak scaling (16M)



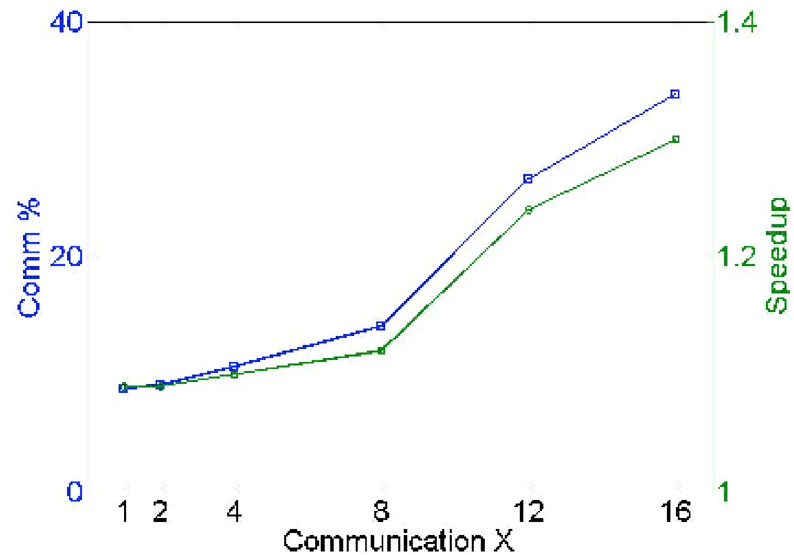
Strong scaling (1600^3)



Hiding communication

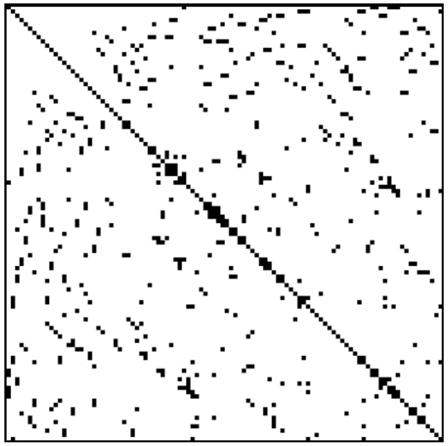


- 1000^3 on 64 cores



Parallel Sparse LU Factorization

- Solve general sparse linear system
  - Ax = b
  - Gaussian elimination
    - A=LU
  - Lower/upper triangular systems
    - Ly=b Ux=y
  - A is sparse
    - Stored in compressed format
    - Indirection
  - A is blocked

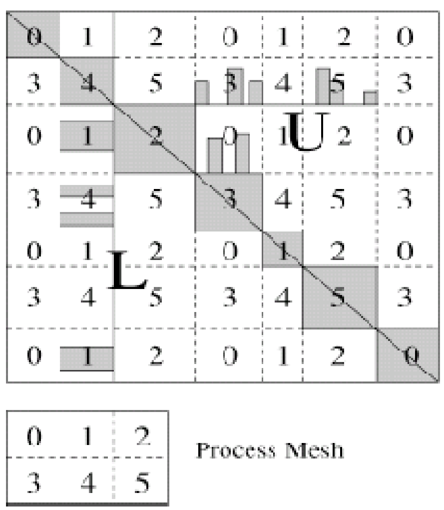


SuperLU

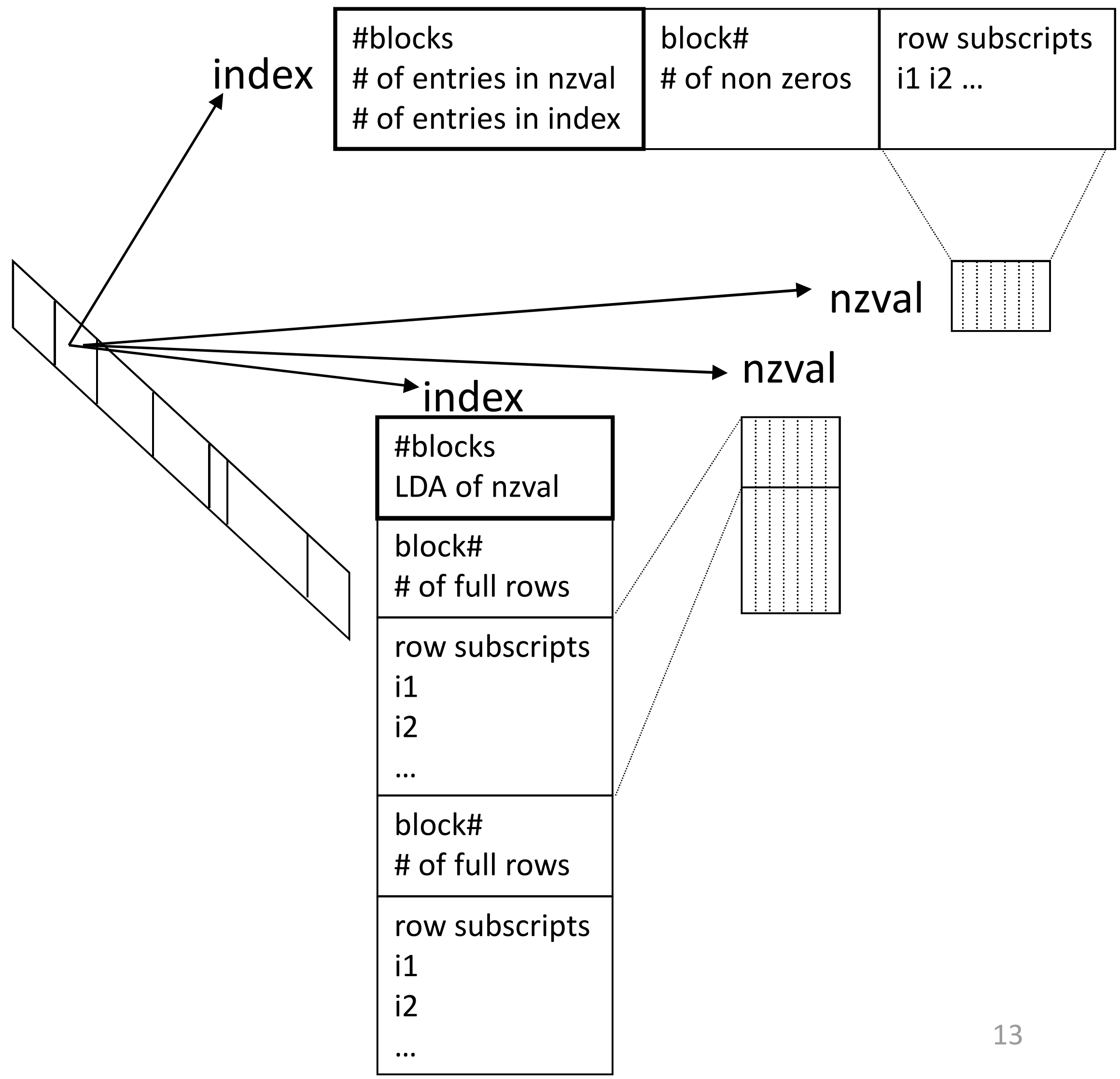
- SuperLU (Li and Demmel)
  - Direct solver using LU factorization
  - Serial, MT, and DIST
- Permutations
  - Increase numerical stability and minimize fill-in
- Symbolic factorization
  - Identify blocks and define data layout
- Numerical factorization (dominates time)
- Triangular solves (5% time)

LU Data

- LU
  - 2D block cyclic mapping

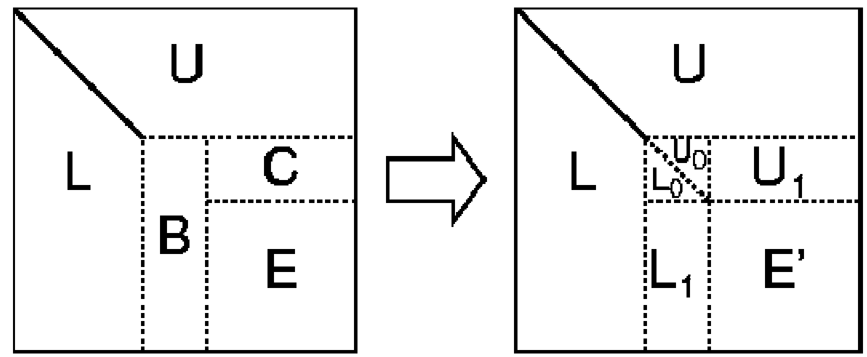


(Figure by Xiaoye S. Li)



SuperLU Factorization

- Panel factorization
  - U_0 , L_0 and L_1= Factor B
  - L_0,$L_{1,}$ U_0 → isend/irecv
- Row update
  - Solve L_0 U_1= C
  - U_1 →send/recv
- Trailing sub-matrix update
  - $E' = E - L_1 \cdot U_1$

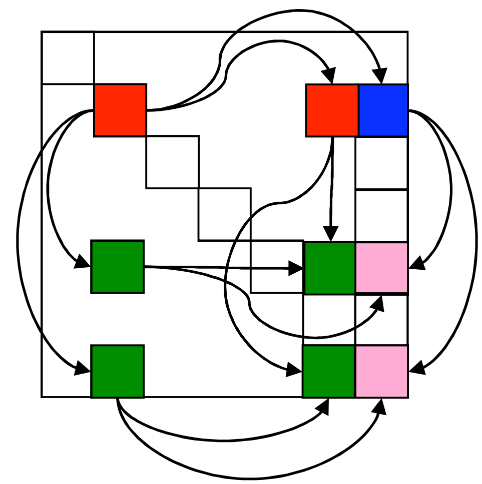


SuperLU+Tarragon

- Challenges (*performance bottlenecks*)
  - Preserve locality?
  - Searching blocks?
  - Graph size?
- Implementations
  - block = task
  - blocks aggregation = task
  - *panels + update = task*

LU with Tarragon (I)

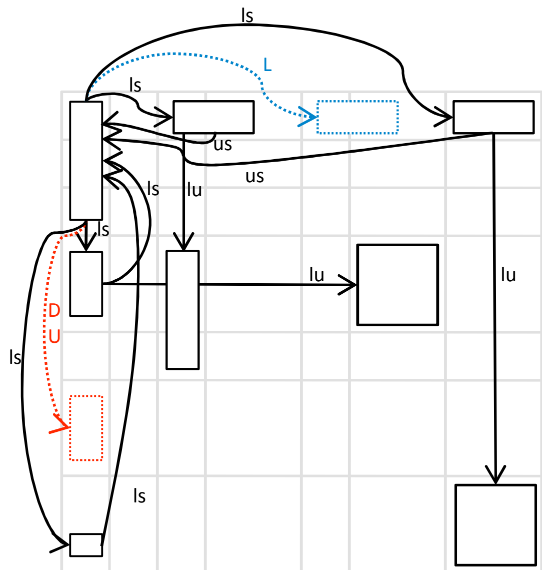
- Task ←→ Block
- Pros
  - High parallelism
- Cons
  - Fine grain computation
  - Fine grain communication
  - Size of the graph



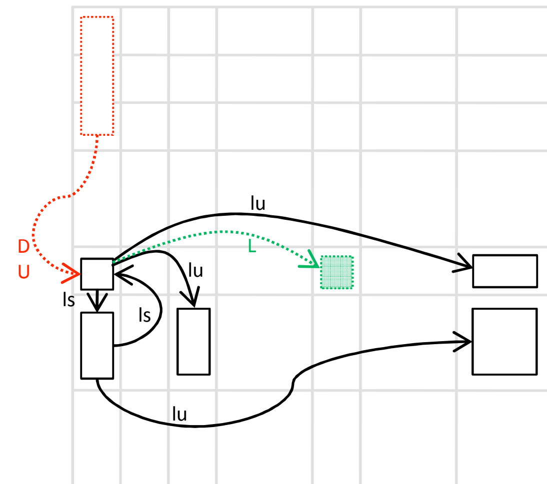
LU with Tarragon (II)

- Blocks aggregation ←→ Task
- Pros
  - Smaller graph
  - Coarser communication
  - Coarser computation
- Cons
  - Reduced parallelism

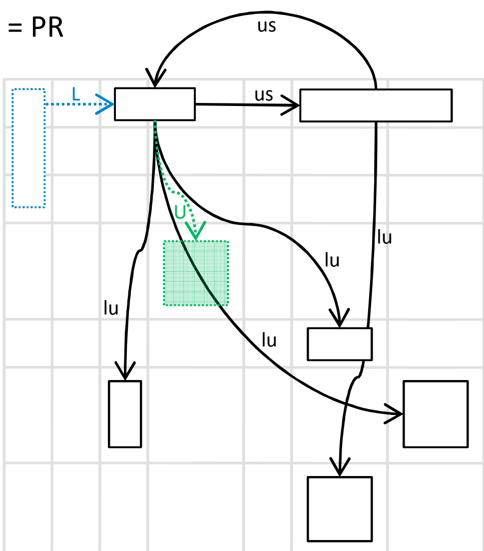
Tasks



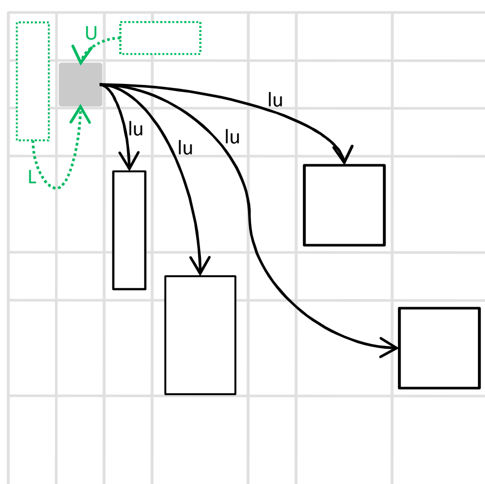
= PC



= PR



≠ PC ≠ PR



Timings

memory access pattern

serialization

	1	1	1	8	8	8	16	16	16
Matrix	SLU	T	T+	SLU	T	T+	SLU	T	T+
bbmat	38.16	15.72	15.00	5.05	3.60	3.41	3.36	32.61	31.60
g7jac200	55.27	29.59	27.32	9.36	6.69	5.99	5.60		
inv-extrusion-1	9.03	16.77	15.81	1.54	3.52	3.05	1.25	8.34	7.57
matrix31	1.94	1.59	1.58	0.51	0.38	0.33	0.50	0.49	0.47
mixing-tank	7.10	24.57	24.03	1.40	4.57	4.57	1.19	4.25	3.82
nasasrb	2.86	3.55	3.43	0.81	0.84	0.80	1.03	7.64	7.31
rajat24	1.46	4.18	3.97	1.26	4.03	4.55	3.32		
stomach	28.32	44.73	43.93	5.51	9.49	9.55	6.11		
torso1	31.03	7.35	7.23	5.65	1.93	1.83	3.86		
twotone	86.57	20.41	19.77	10.66	4.61	4.77	6.96		

memory requirements

Timings

memory access pattern

	1	1	1	8	8	8	16	16	16
Matrix	SLU	T	T+	SLU	T	T+	SLU	T	T+
bbmat	38.16	15.72	15.00	5.05	3.60	3.41	3.36	6.17	6.16
g7jac200	55.27	29.59	27.32	9.36	6.69	5.99	5.60		
inv-extrusion-1	9.03	16.77	15.81	1.54	3.52	3.05	1.25	4.52	4.60
matrix31	1.94	1.59	1.58	0.51	0.38	0.33	0.50	0.49	0.49
mixing-tank	7.10	24.57	24.03	1.40	4.57	4.57	1.19	5.48	5.71
nasasrb	2.86	3.55	3.43	0.81	0.84	0.80	1.03	1.25	1.28
rajat24	1.46	4.18	3.97	1.26	4.03	4.55	3.32		
stomach	28.32	44.73	43.93	5.51	9.49	9.55	6.11		
torso1	31.03	7.35	7.23	5.65	1.93	1.83	3.86		
twotone	86.57	20.41	19.77	10.66	4.61	4.77	6.96		

memory requirements

Conclusions

- Jacobi 3D
  - Effectively hides communication cost
- SuperLU
  - Constrains
    - Difficult to preserve locality when implementing fine grain parallelism
    - Data mapping doesn't match well one-process-per-node configuration
  - Problems
    - Poor task locality
    - Redundant memory operations
    - Memory foot print inflated by communication buffers
  - Work in progress
    - panels+update decomposition
- Thank you!