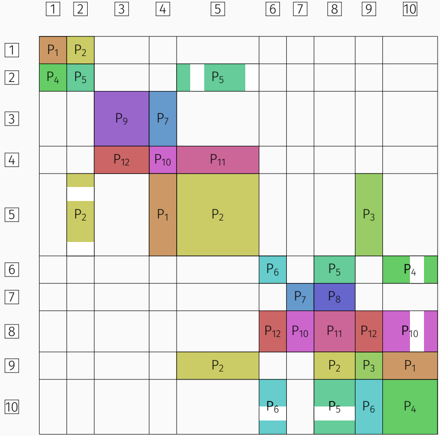
SPARSE MATRIX 2D BLOCK LAYOUT

```
for Supernode $\mathcal{K} = \mathcal{N}$ down to 1 do
    Compute selected elements of $A^{-}1$ within $\mathcal{K}$
end
```



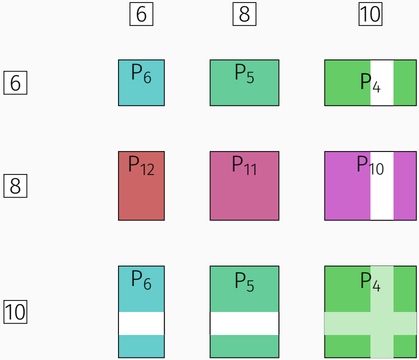
1
2
3
4
5
6
7
8
9
10
1 P_1 P_2
2 P_4 P_5 P_5
3 P_9 P_7
4 P_{12} P_{10} P_{11}
5 P_2 P_1 P_2 P_3
6 P_6 P_5 P_4
7 P_7 P_8
8 P_{12} P_{10} P_{11} P_{12} P_{10}
9 P_2 P_2 P_3 P_1
10 P_6 P_5 P_6 P_4

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9
P_{10}	P_{11}	P_{12}

- 2D Block Cyclic layout
- 4-by-3 processor grid
- No explicit load balancing
- Works well in practice [Gupta]

PARALLEL PROCESSING OF A SUPERNODE

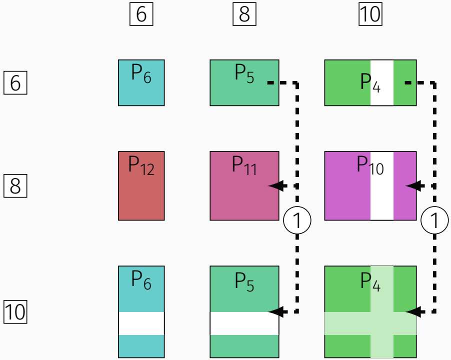
$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$



- D: diagonal block
- L: lower triangular block
- U: upper triangular block

PARALLEL PROCESSING OF A SUPERNODE

$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

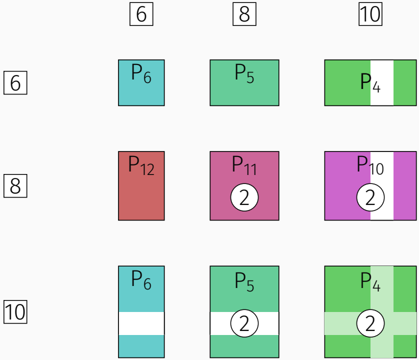


- D: diagonal block
- L: lower triangular block
- U: upper triangular block

- Broadcast L along columns

PARALLEL PROCESSING OF A SUPERNODE

$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1}\ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

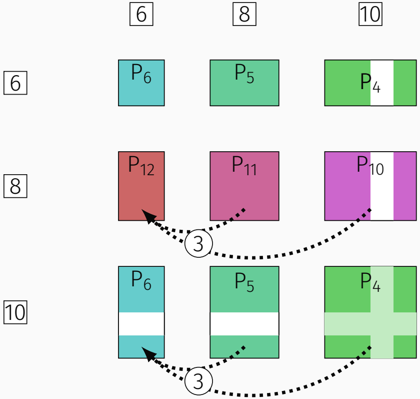


- D: diagonal block
- L: lower triangular block
- U: upper triangular block

- Broadcast L along columns

PARALLEL PROCESSING OF A SUPERNODE

$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

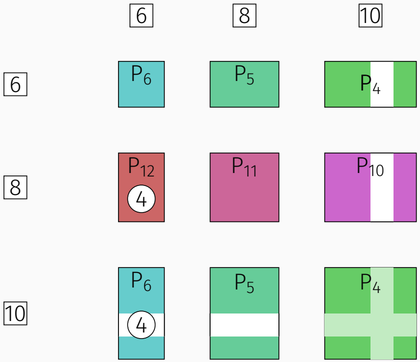


- D: diagonal block
- L: lower triangular block
- U: upper triangular block

- Broadcast L along columns
- Reduce contributions to L along rows

PARALLEL PROCESSING OF A SUPERNODE

$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

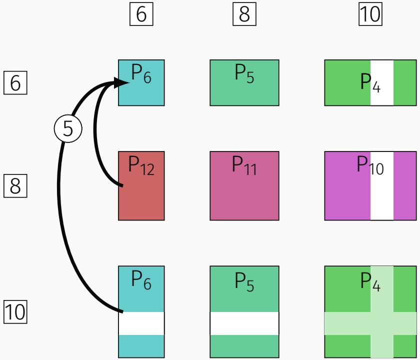


- D: diagonal block
- L: lower triangular block
- U: upper triangular block

- Broadcast L along columns
- Reduce contributions to L along rows

PARALLEL PROCESSING OF A SUPERNODE

$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

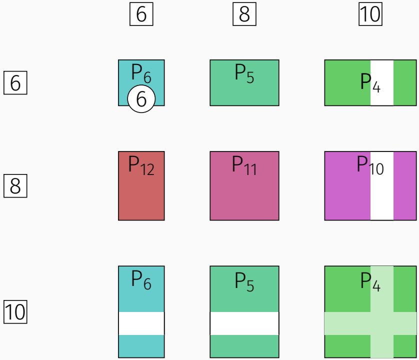


- D: diagonal block
- L: lower triangular block
- U: upper triangular block

- Broadcast L along columns
- Reduce contributions to L along rows
- Reduce contribution to D within supernode column

PARALLEL PROCESSING OF A SUPERNODE

$$A^{-1} = \begin{pmatrix} d^{-1} + \ell^T S_{-1} \ell & -\ell^T S^{-1} \\ -S^{-1}\ell & S^{-1} \end{pmatrix}$$

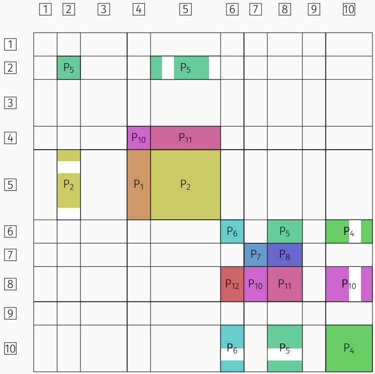


- D: diagonal block
- L: lower triangular block
- U: upper triangular block

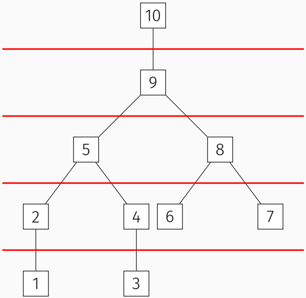
- Broadcast L along columns
- Reduce contributions to L along rows
- Reduce contribution to D within supernode column

CONCURRENCY BETWEEN SUPERNODES

```
for Supernode $\mathcal{K} = \mathcal{N}$ down to 1 do
    $\mathcal{R}_{\mathcal{K}} \leftarrow$ non-zero rows in supernode $\mathcal{K}$
    $Y = S^{-1}_{\mathcal{R}_{\mathcal{K}},\mathcal{R}_{\mathcal{K}}} \ell_{\mathcal{R}_{\mathcal{K}},\mathcal{K}}$
    $A_{\mathcal{K},\mathcal{K}} \leftarrow d^{-1} + Y^{T} \ell_{\mathcal{R}_{\mathcal{K}},\mathcal{K}}$
    $A_{\mathcal{R}_{\mathcal{K}},\mathcal{K}} \leftarrow -Y$
end
```



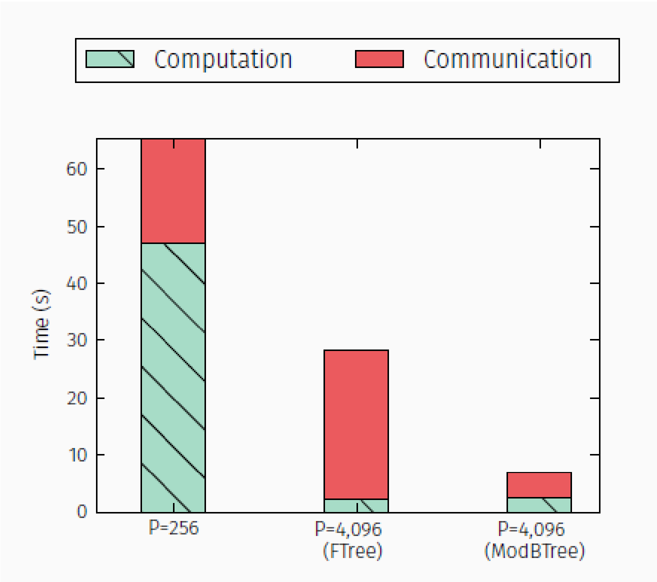
- Top-Down elimination tree traversal
- Exploit elimination tree to increase concurrency

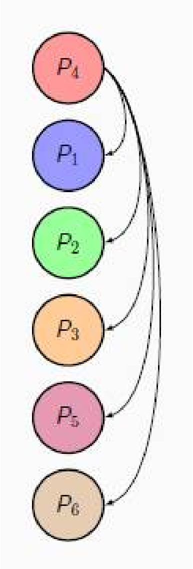


- Serializations from common ancestors
- Serializations from layout
- How to schedule supernodes ?
- Level-based heuristic as first step

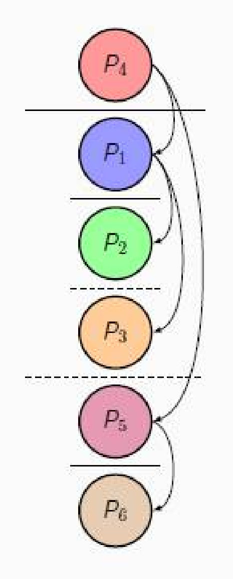
Progress: Tree-level parallelization

- “Flat-tree” communication pattern not efficient

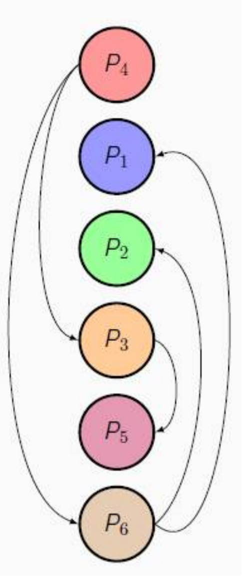




Flat tree



Binary tree



Mod Binary tree

5x Speedup!

[M. Jacquelin, LL, N. Wichmann and C. Yang, IPDPS 2016]

PEXSI MPI + X

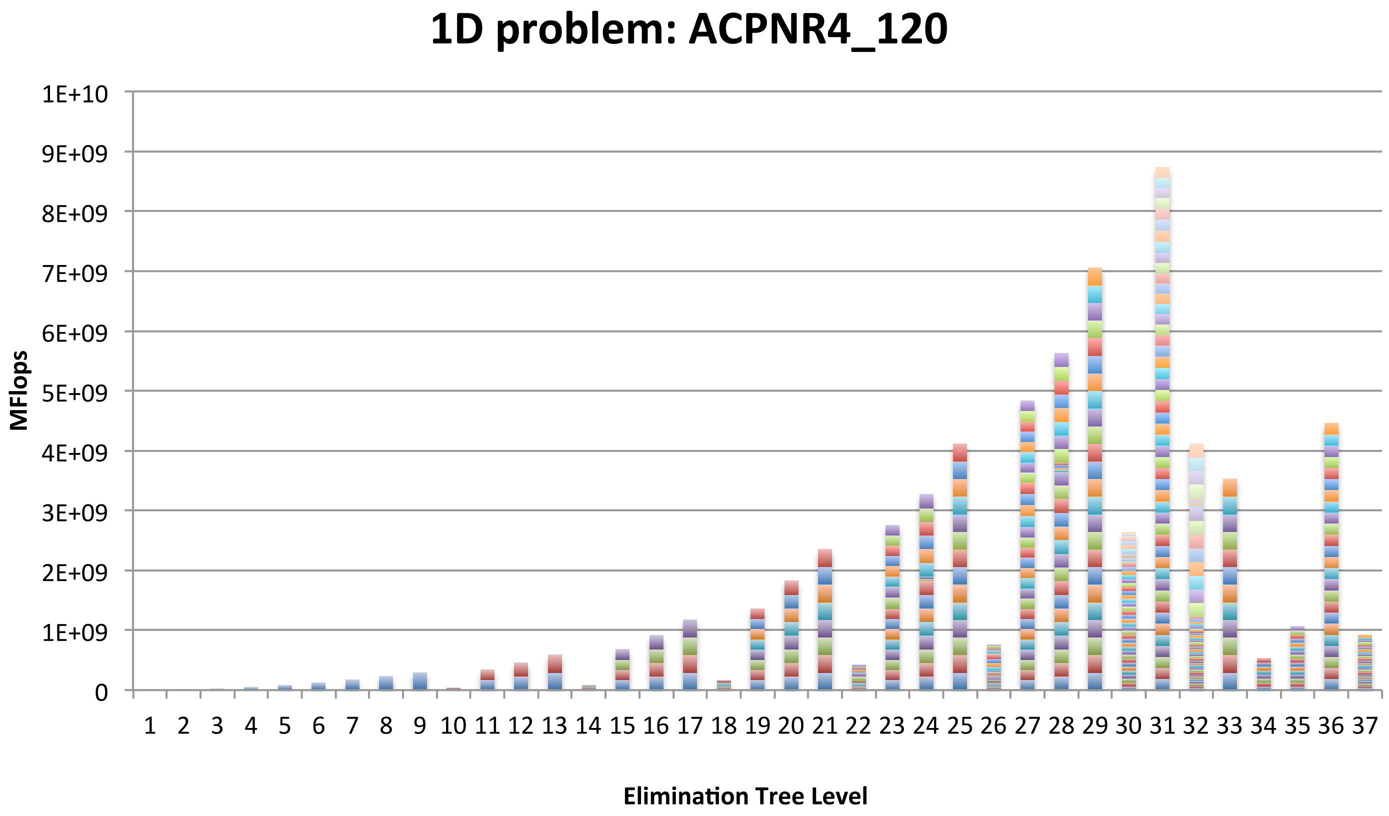
- Target architecture: Manycore CPUs & GPUs
- Enough parallelism within node ?
- Two different strategies:
  - Fine granularity tasks to keep all cores busy (CPUs)
  - Fork-join model with compute intensive phases (GPUs)
  - Focus on KNL first
- Need to overlap MPI communications with parallel intranode computations

Amount of parallelism in PEXSI

- MPI Messages are “aggregated”

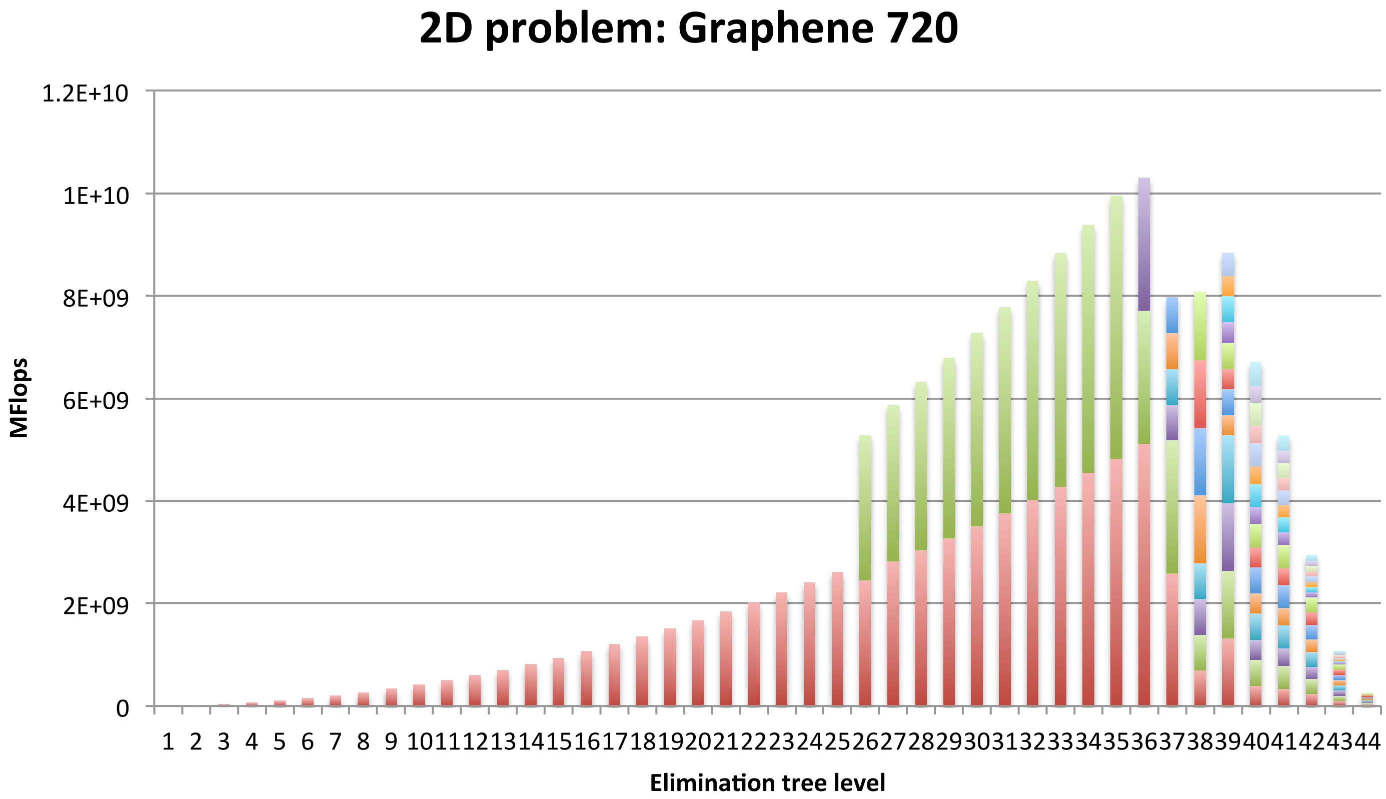
Amount of parallelism in PEXSI

- MPI Messages are “aggregated”



Amount of parallelism in PEXSI

- MPI Messages are "aggregated"



- Fewer GEMMs as matrix get denser ⇔ Need to break down GEMMs into smaller chunks

Parallelization strategy

- Use OpenMP Task to perform Indirect addressing and GEMMs
- Each GEMM is performed sequentially within a task
- One MPI message ⇔ Many fine granularity tasks

Parallelization strategy

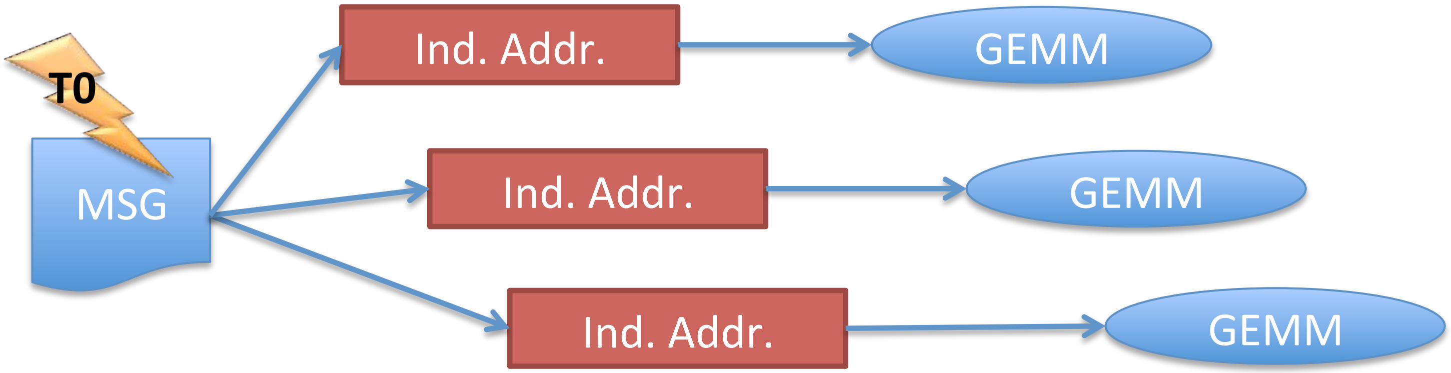
- Use OpenMP Task to perform Indirect addressing and GEMMs
- Each GEMM is performed sequentially within a task
- One MPI message ⇔ Many fine granularity tasks

Parallelization strategy

- Use OpenMP Task to perform Indirect addressing and GEMMs
- Each GEMM is performed sequentially within a task
- One MPI message ⬄ Many fine granularity tasks

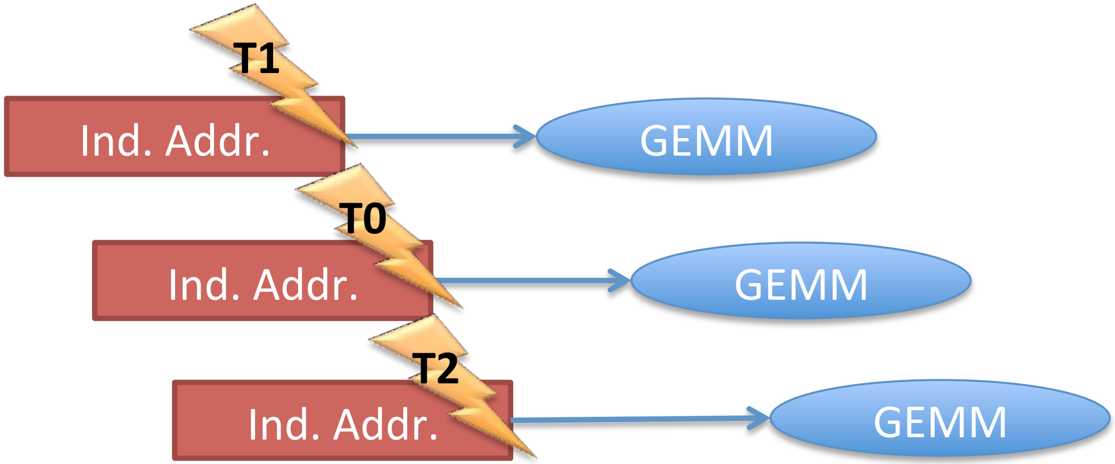
Parallelization strategy

- Use OpenMP Task to perform Indirect addressing and GEMMs
- Each GEMM is performed sequentially within a task
- One MPI message ⇔ Many fine granularity tasks



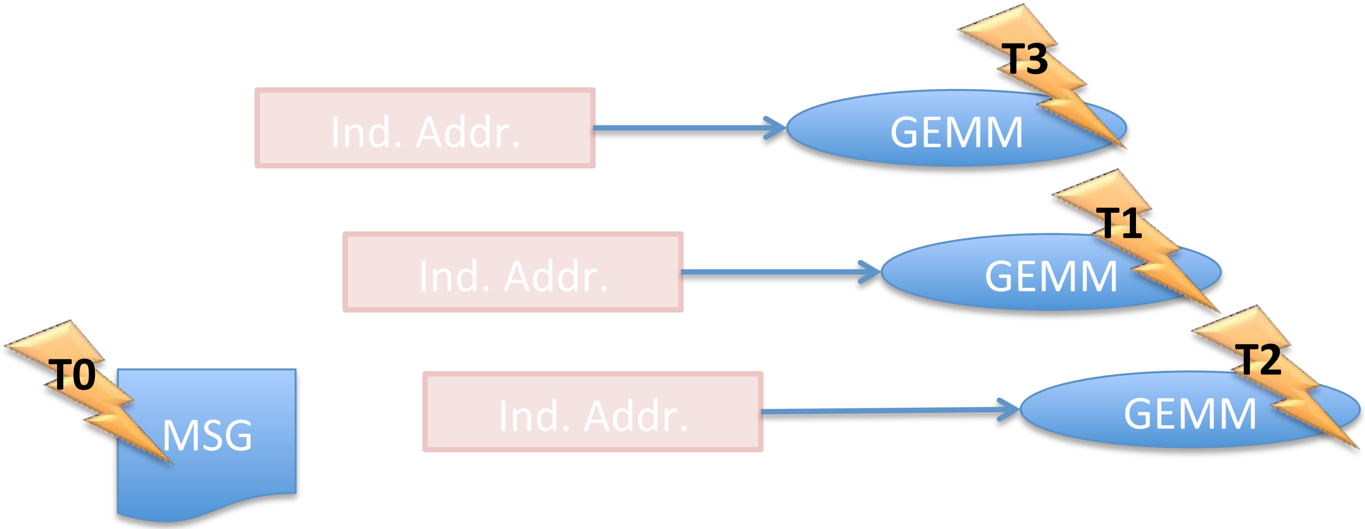
Parallelization strategy

- Use OpenMP Task to perform Indirect addressing and GEMMs
- Each GEMM is performed sequentially within a task
- One MPI message ⬄ Many fine granularity tasks



Parallelization strategy

- Use OpenMP Task to perform Indirect addressing and GEMMs
- Each GEMM is performed sequentially within a task
- One MPI message ⬄ Many fine granularity tasks


T3
Ind. Addr.
GEMM
T1
Ind. Addr.
GEMM
T0
T2
MSG
Ind. Addr.
GEMM