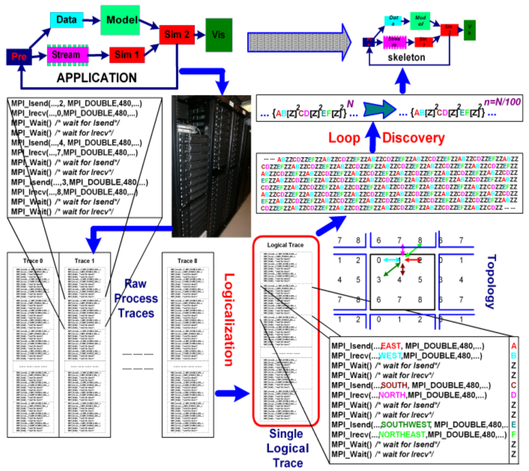
"Logicalization" of MPI Communication Traces

Qiang Xu **Jaspal Subhlok** **Rong Zheng**
University of Houston

Sara Voss
Coe College, Iowa

Context is Performance Skeleton construction

Performance skeleton: short running program whose execution time reflects the execution time of corresponding application



Logicalization

Combines all processor traces into a single logical program trace.

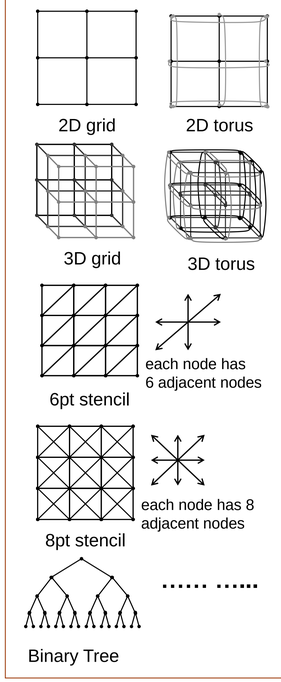
- Identify communication pattern
- Convert physical communication (process ids) to logical communication (e.g. to EAST neighbor in a grid)

*Orthogonal to **Trace Compression** based on **Loop Discovery** in a single trace*

Contact: www2.cs.uh.edu/~jaspal

Logicalization

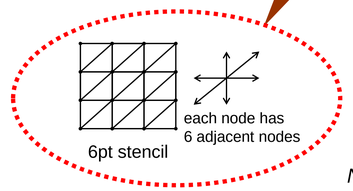
Topology Templates:



Communication Matrix:

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
P0	0	1	0	1	1	0	0	1	0	0	0	0	1	1	0	0
P1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	1	0
P2	0	1	0	1	0	1	1	0	0	0	0	0	0	0	1	1
P3	1	0	1	0	0	0	1	1	0	0	0	0	1	0	0	1
P4	1	1	0	0	0	1	0	1	1	0	0	1	0	0	0	0
P5	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0	0
P6	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	0
P7	1	0	0	1	1	0	1	0	0	0	1	1	0	0	0	0
P8	0	0	0	0	1	1	0	0	0	1	0	1	1	0	0	1
P9	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	0
P10	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0
P11	0	0	0	0	1	0	0	1	1	0	1	0	0	0	1	1
P12	1	0	0	1	0	0	0	0	1	1	0	0	0	1	0	1
P13	1	1	0	0	0	0	0	0	0	1	1	0	1	0	1	0
P14	0	1	1	0	0	0	0	0	0	0	1	1	0	1	0	1
P15	0	0	1	1	0	0	0	0	1	0	0	1	1	0	1	0

Graph Matching



Physical Execution Traces:

Trace 0 … … …Trace 15

```
..... .....
MPI_Isend(... 1, MPI_DOUBLE, 480, ...)
MPI_Irecv(... 3, MPI_DOUBLE, 480, ...)
MPI_Wait() /* wait for Isend */
MPI_Wait() /* wait for Irecv */
..... .....
MPI_Isend(... 4, MPI_DOUBLE, 480, ...)
MPI_Irecv(...12, MPI_DOUBLE, 480, ...)
MPI_Wait() /* wait for Isend */
MPI_Wait() /* wait for Irecv */
..... .....
MPI_Isend(... 7, MPI_DOUBLE, 480, ...)
MPI_Irecv(...13, MPI_DOUBLE, 480, ...)
MPI_Wait() /* wait for Isend */
MPI_Wait() /* wait for Irecv */
..... .....
```

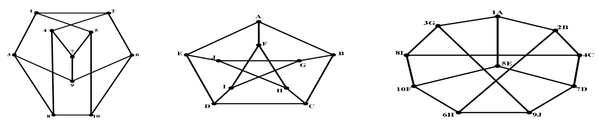
Single Logical Trace:

```
..... .....
MPI_Isend(...EAST, MPI_DOUBLE, 480, ...)
MPI_Irecv(...WEST, MPI_DOUBLE, 480, ...)
MPI_Wait() /* wait for Isend */
MPI_Wait() /* wait for Irecv */
..... .....
MPI_Isend(...SOUTH, MPI_DOUBLE, 480, ...)
MPI_Irecv(...NORTH, MPI_DOUBLE, 480, ...)
MPI_Wait() /* wait for Isend */
MPI_Wait() /* wait for Irecv */
..... .....
MPI_Isend(...SOUTHWEST, MPI_DOUBLE, 480, ...)
MPI_Irecv(...NORTHEAST, MPI_DOUBLE, 480, ...)
MPI_Wait() /* wait for Isend */
MPI_Wait() /* wait for Irecv */
..... .....
```

NAS BT code

Challenge: *Graph isomorphism*

Graph matching must solve graph isomorphism problem that has no known polynomial algorithms



Solution approach

First, eliminate almost all topologies by simple pre- tests: counting nodes, edges, degrees, graph spectrum

Then utilize VF2 graph matching algorithm and software

Conclusions

- Results show that the solution is effective in practice
- Applicable to static patterns. Can be extended to multiple patterns and multiple communication phases
- Logicalization combined with single trace compression for full solution
- Local (non global) communication causes inaccuracy but not failure

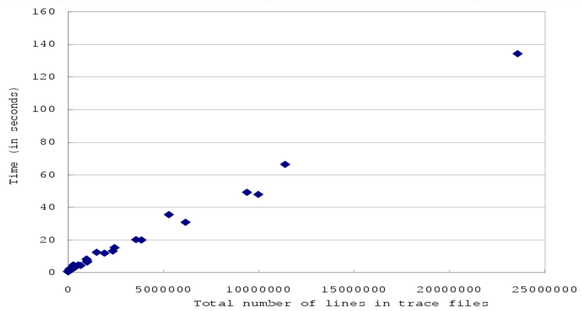
Results

Topologies Discovered in NPB:

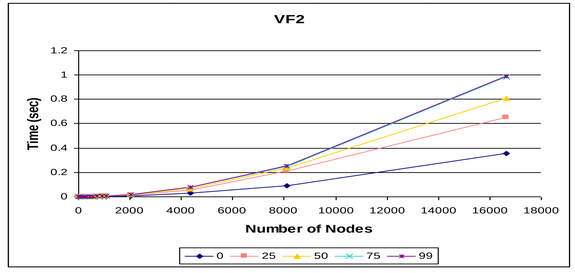
NPB	Processes	Graph Matching	NPB	Processes	Graph Matching
BT/SP	9	**3×3 6-p stencil**	MG	8	**2×2×2 grid**
	16	**4×4 6-p stencil**			4×2 torus
	36	**6×6 6-p stencil**		16	**2×2×2×2 grid**
	64	**8×8 6-p stencil**			4×2×2 torus
	121	**11×11 6-p stencil**			4×4 torus
LU	8	**4×2 grid**		32	**2×2×2×2×2 grid**
	16	**4×4 grid**			4×2×2×2 torus
	32	**8×4 grid**			4×4×2 torus
	64	**8×8 grid**		64	**2×2×2×2×2×2 grid**
	128	**16×8 grid**			4×2×2×2×2 torus
CG	8	**CG stencil**			4×4×2×2 torus
	16	**CG stencil**			4×4×4 torus
	32	**CG stencil**		128	**8×2×2×2×2 torus**
	64	**CG stencil**			8×4×2×2 torus
	128	**CG stencil**			8×4×4 torus

Performance of Logicalization

Name	16 processes		32/36 processes		64 processes		121/128 processes	
	Trace Length Records (Size)	Time (secs)	Trace Length Records (Size)	Time (secs)	Trace Length Records (Size)	Time (secs)	Trace Length Records (Size)	Time (secs)
BT	17106 (731 KB)	2.64	26754 (1081 KB)	8.35	36402 (1459 KB)	13.19	50874 (2106 KB)	30.76
SP	26888 (1147 KB)	4.14	41324 (17543 KB)	12.55	55760 (2365 KB)	20.34	77414 (3365 KB)	49.16
CG	41954 (1667 KB)	4.52	59964 (2376 KB)	11.94	59964 (2376 KB)	19.89	77978 (3224 KB)	47.89
LU	203048 (9185 KB)	15.39	203048 (9186 KB)	35.46	203048 (9088 KB)	66.28	203048 (9433 KB)	134.30
MG	8909 (373 KB)	2.48	8951 (374 KB)	4.56	8953 (373 KB)	4.75	9035 (386 KB)	7.33



Scalability of VF2 Graph Isomorphism



Varying degree of randomness in node numbering