Parallel State Space Searching Algorithms

Part II

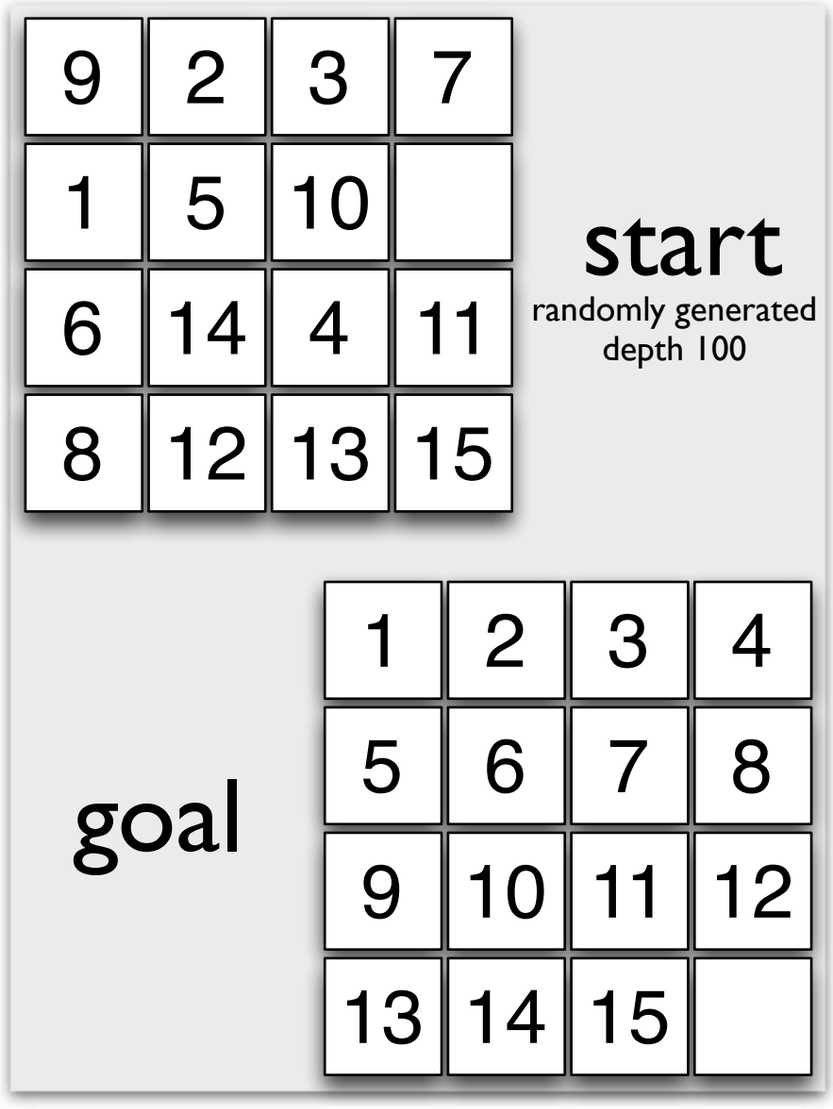
Professor Keyes – APAM 4990
Hart Lambur, Blake Shaw

Overview

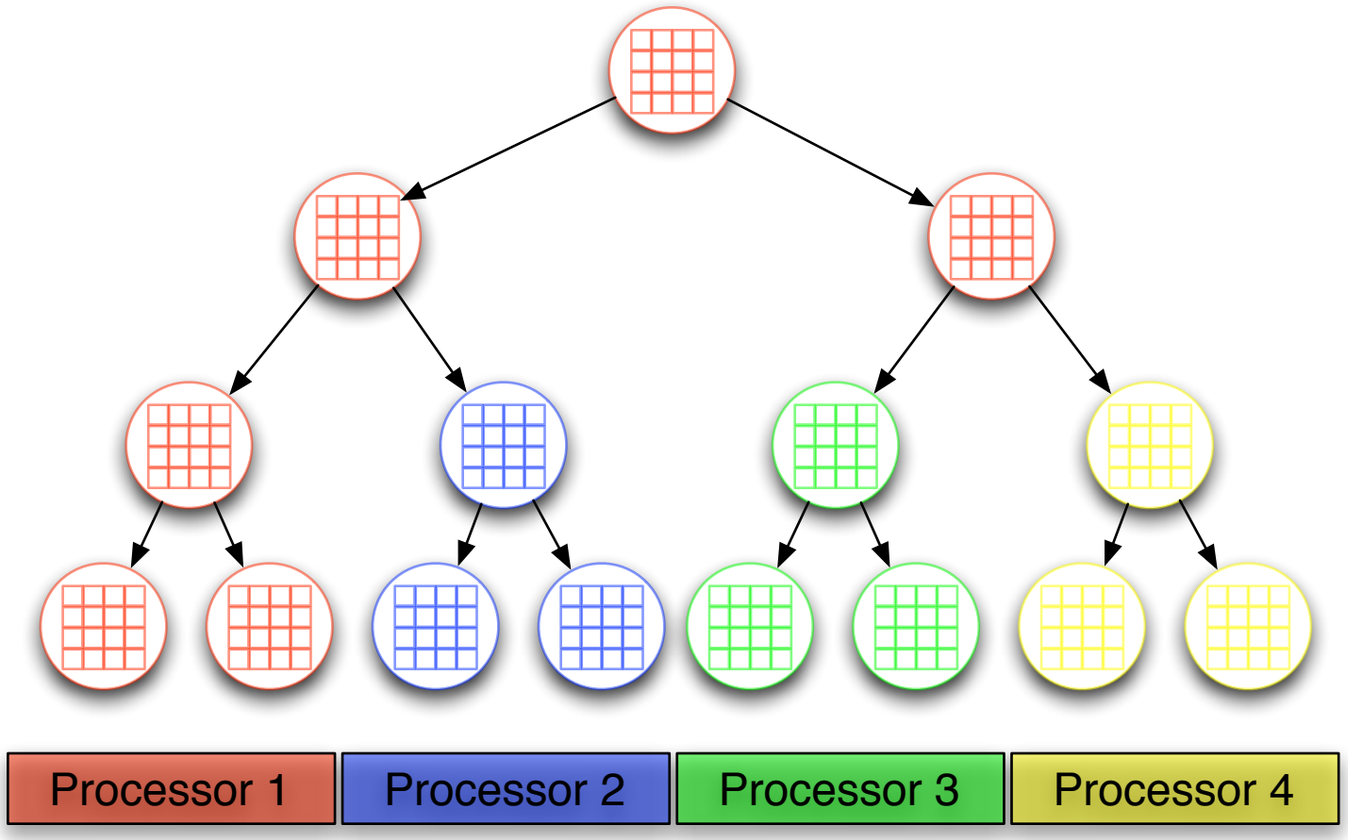
- State space searching – 15 puzzle review
- Two implementations
  - Parallel IDDFS
  - Parallel A* with neighbor synchronization
- Profiling with Jumpshot
- Performance and scaling
- Lessons learned & future directions

Review –15 puzzle

- State space searching
- 15 puzzle
  - Complexity
- Serial Algorithms
  - BFS / DFS / A*



Parallel ID-DFS



- Iterative Deepening
- Work splitting strategy
  - Head processor distributes start nodes
  - Each processor maintains own open / closed list

Parallel ID-DFS Output

```
Generating start node with random depth: 12
Puzzle State: w/ path: (SSENWNESSEEN) depth: 12 and h-value: 12

1    8    2    3
4    5    6    *
9    10   11   7
12   13   14   15

Distributing 4 Start Nodes to Processors
Done Distributing Nodes

3: Found Goal Node at Depth: 12 with path: SWWNNWSESWNN

0: Stopping. Goal found by: 3 while I was searching at depth bound 11
1: Stopping. Goal found by: 3 while I was searching at depth bound 14
2: Stopping. Goal found by: 3 while I was searching at depth bound 15

0: DONE after 36.20 sec. Current Node Depth: 12 Gen: 5956 Exp: 2867 Del: 275
1: DONE after 48.79 sec. Current Node Depth: 15 Gen: 9957 Exp: 4765 Del: 515
2: DONE after 66.48 sec. Current Node Depth: 16 Gen: 12274 Exp: 5873 Del: 582
3: DONE after 19.75 sec. Current Node Depth: 12 Gen: 5310 Exp: 2560 Del: 298
```

TimeLine : 05-May-08-1501prof.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 0	0	-0.001353	-0.001353	18.2658366505	66.4784585	66.4784585	0.1037126544

Cumulative...

Row

Row Count 5.0

TimeLines

SLOG-2

0

1

2

3

@ LineID

-4

-3

-2

-1

Fit All Row

0.00 5.00 10.00 15.00 20.00 25.00 30.00 35.00 40.00 45.00 50.00 55.00 60.00 65.0

Time (seconds)

TimeLine : 05-May-08-1501prof.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 0	14	-0.001353	-0.001353	-0.0003029729	0.0020765486	66.4784585	0.0000053503

Cumulative...

Row

Row Count

5.0

TimeLines

SLOG-2

0

1

2

3

@ LineID

-0.0012 -0.0008 -0.0004 0.00 0.0004 0.0008 0.0012 0.0016 0.00

Time (seconds)

−4 −3 −2 −1

Fit All Row

Parallel Best First (A*)

- Work splitting strategy:
  - Head processor distributes nodes
  - Each processor maintains own ‘best’ list (on open list) and closed list
  - Synchronization strategy to ensure ‘best’ part of tree is spread to each processor
- Ring communication synchronization strategy
  - Each processor passes it’s best node to its neighbor periodically

Parallel A* Output

```
Generating start node with random depth: 30
Puzzle State: w/ path: (SESEESWNNESSWWWNNEESWSWNNNEESW) depth: 30 and h-value: 20

1    2    6    3
4    *    7    15
9    12   10   14
5    8    13   11

Distributing 8 Start Nodes to Processors
Done Distributing Nodes

4: Found Goal Node at Depth: 30 with path: WSESWNNESSEENNWSSENWNNWWSESWNN

0: Stopping. Goal found by: 4 while I was at depth: 8
1: Stopping. Goal found by: 4 while I was at depth: 8
2: Stopping. Goal found by: 4 while I was at depth: 14
3: Stopping. Goal found by: 4 while I was at depth: 14
5: Stopping. Goal found by: 4 while I was at depth: 21
6: Stopping. Goal found by: 4 while I was at depth: 16
7: Stopping. Goal found by: 4 while I was at depth: 8

0: DONE. Current Node Depth: 8 Gen: 2973 Exp: 1700 Del: 359
1: DONE. Current Node Depth: 8 Gen: 3003 Exp: 1710 Del: 347
2: DONE. Current Node Depth: 14 Gen: 3031 Exp: 1712 Del: 405
3: DONE. Current Node Depth: 14 Gen: 3030 Exp: 1720 Del: 361
4: DONE. Current Node Depth: 30 Gen: 3046 Exp: 1716 Del: 376
5: DONE. Current Node Depth: 21 Gen: 3151 Exp: 1710 Del: 258
6: DONE. Current Node Depth: 16 Gen: 3266 Exp: 1708 Del: 126
7: DONE. Current Node Depth: 8 Gen: 3069 Exp: 1697 Del: 293
```

TimeLine : 05-May-08-1706prof.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
1 / 4	0	0.0002275	0.0002275	0.0014277875	15.1788655	15.1788655	0.0250472574

BaseAligne...

SLOG-2

0
1
2
3
4
5
6
7

@ LineID

TimeLines

Row

Row Count

9.0

8
7
6
5
4
3
2
1

.00 1.25 2.50 3.75 5.00 6.25 7.50 8.75 10.00 11.25 12.50 13.75 15

Time (seconds)

Fit All Row

TimeLine : 05-May-08-1706prof.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 4	11	0.0002275	0.0002275	0.0009122386	0.005775467	15.1788655	0.0000067004

BaseAligne...

SLOG-2

0
1
2
3
4
5
6
7

@ LineID

TimeLines

Row

Row Count

9.0

8
7
6
5
4
3
2
1

Fit All Row

0.0005 0.001 0.0015 0.002 0.0025 0.003 0.0035 0.004 0.0045 0.005 0.0055

Time (seconds)

TimeLine : 05-May-08-1706prof.slog2 <Identity Map>

Lowest / Max. Depth	Zoom Level	Global Min Time	View Init Time	Zoom Focus Time	View Final Time	Global Max Time	Time Per Pixel
0 / 4	8	0.0002275	0.4431708264	0.4749525504	0.5067342867	15.1788655	0.0000767675

BaseAligne...

SLOG-2

0
1
2
3
4
5
6
7

@ LineID

TimeLines

Row

Row Count

9.0

8
7
6
5
4
3
2
1

Fit All Row

0.445 0.45 0.455 0.46 0.465 0.47 0.475 0.48 0.485 0.49 0.495 0.50 0.505

Time (seconds)

TimeLine : 05-May-08-1706prof.slog2 <Identity Map>

Lowest / Max. Depth: 0 / 4
Zoom Level: 7
Collapse the Y-axis tree label by 1 level
0.0002275
15.021413218
Zoom Focus Time: 15.1295020906
View Final Time: 15.1788655
Global Max Time: 15.1788655
Time Per Pixel: 0.0001901598
Row
Row Count: 9.0
BaseAligne...
TimeLines
SLOG-2
0
1
2
3
4
5
6
7
@ LineID
15.025
15.0375
15.05
15.0625
15.075
15.0875
15.10
15.1125
15.125
15.1375
15.15
15.1625
Time (seconds)
8
7
6
5
4
3
2
1
Fit All Row

Performance

Depth 10

ID-DFS NP=1	ID-DFS NP=4	A* NP=1	A* NP=4
277.527	87.15	0.000575	0.005918

- Parallel ID-DFS shows 3.2x speedup factor
- A* algorithmically superior to ID-DFS
- Parallel overhead of A* dominates search time for simple problems

Scaling

Depth 30

A* NP=1	A* NP=4	A* NP=8	A* NP=12
26.8331	3.14834	4.9506	0.330026

- Data shows obvious inconsistencies
- Results from AFQ cluster vary for repeated trials
  - Unavailability of undisturbed CPU time
  - Varying communication overhead
- Need to average over a large number of unique puzzles
  - Given the nature of the cluster we didn't feel the need to do extensive data collection

Conclusion

- Lessons learned
  - Difficulty of asynchronous computing
  - MPI is much harder to debug and understand
  - Proper usage of LSF and cluster environment is important
- Future directions
  - Room for performance improvement
  - Many more synchronization strategies to implement and develop (and invent)