S-HARP:
A Scalable Parallel Dynamic Partitioner

Andrew Sohn
Computer Science Department
New Jersey Institute of Technology

Horst Simon
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

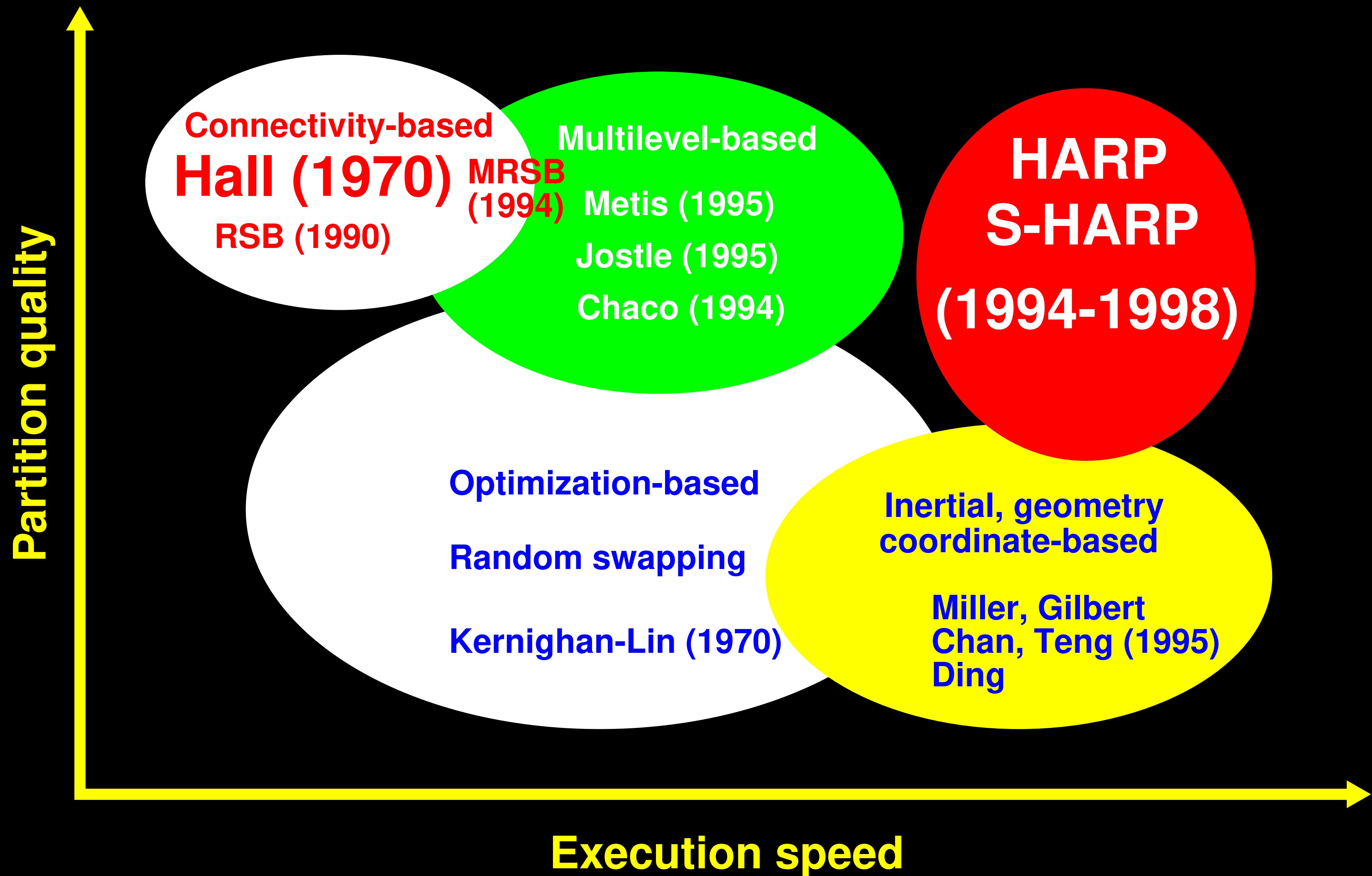
Supercomputing 98, Orlando, Florida, November 7 - 13, 1998

Global Load Balancer JOVE

(Sohn and Simon, 1994)

- Dual graph
- Fast partitioning
- Accurate metric

Classification of Partitioners



S-HARP

A universal scalable partitioner

- ✔ fast partitioning from scratch
- ✔ any # of partitions on any # of processors
- ✔ no initial layout of graph
- ✔ no information needed from previous iterations
- ✔ no imbalance factor issue
- ✔ no parameters to fine tune
- ✔ no local optimization
- ✔ much more scalable than other paritioners

Assumptions and Issues

- eigenvectors precomputed

  (talk on a scalable eigensolver in the afternoon by *Wu and Simon.* The eigensolver is currently being integrated into S-HARP)

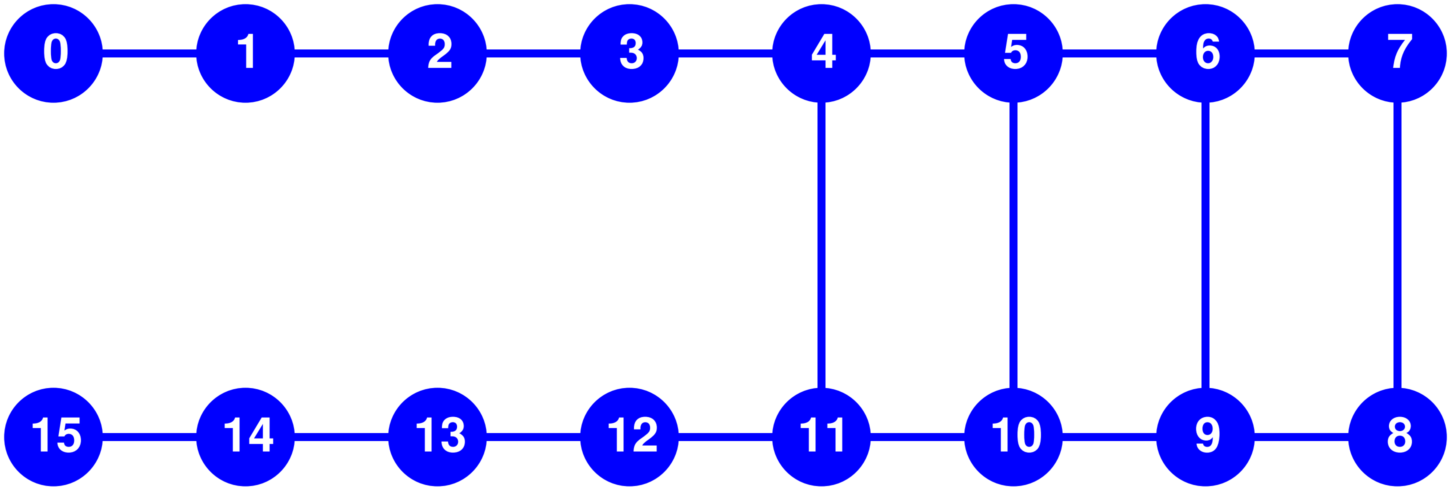
- no topological change during mesh adaption
- does not handle weighted edges and will be investigated.

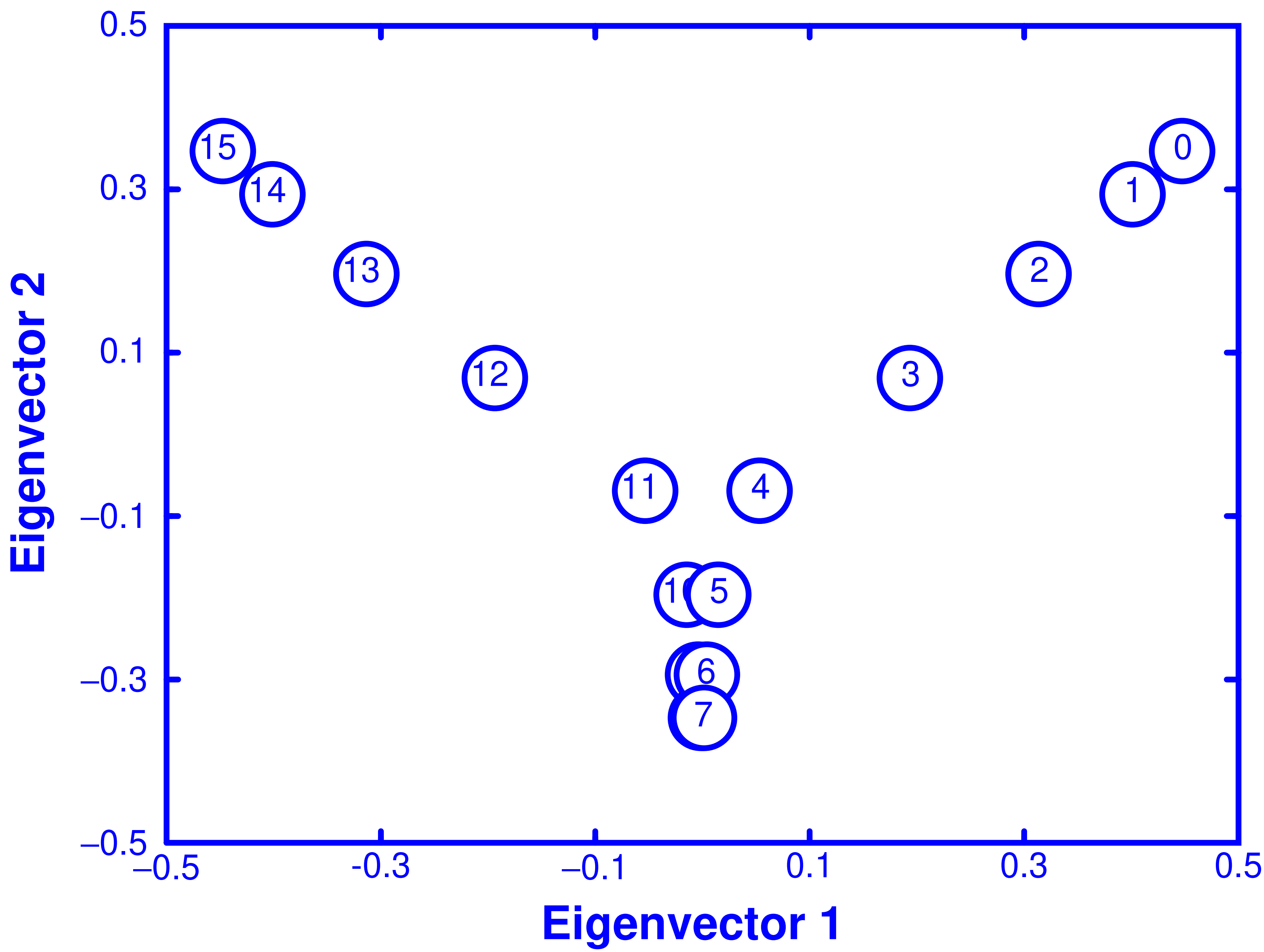
The Idea

- Eigenvectors from connectivity
- Vertices represented in eigen coordinate
- Inertial partitioning in eigen coordinate

The 'Roach' Mesh

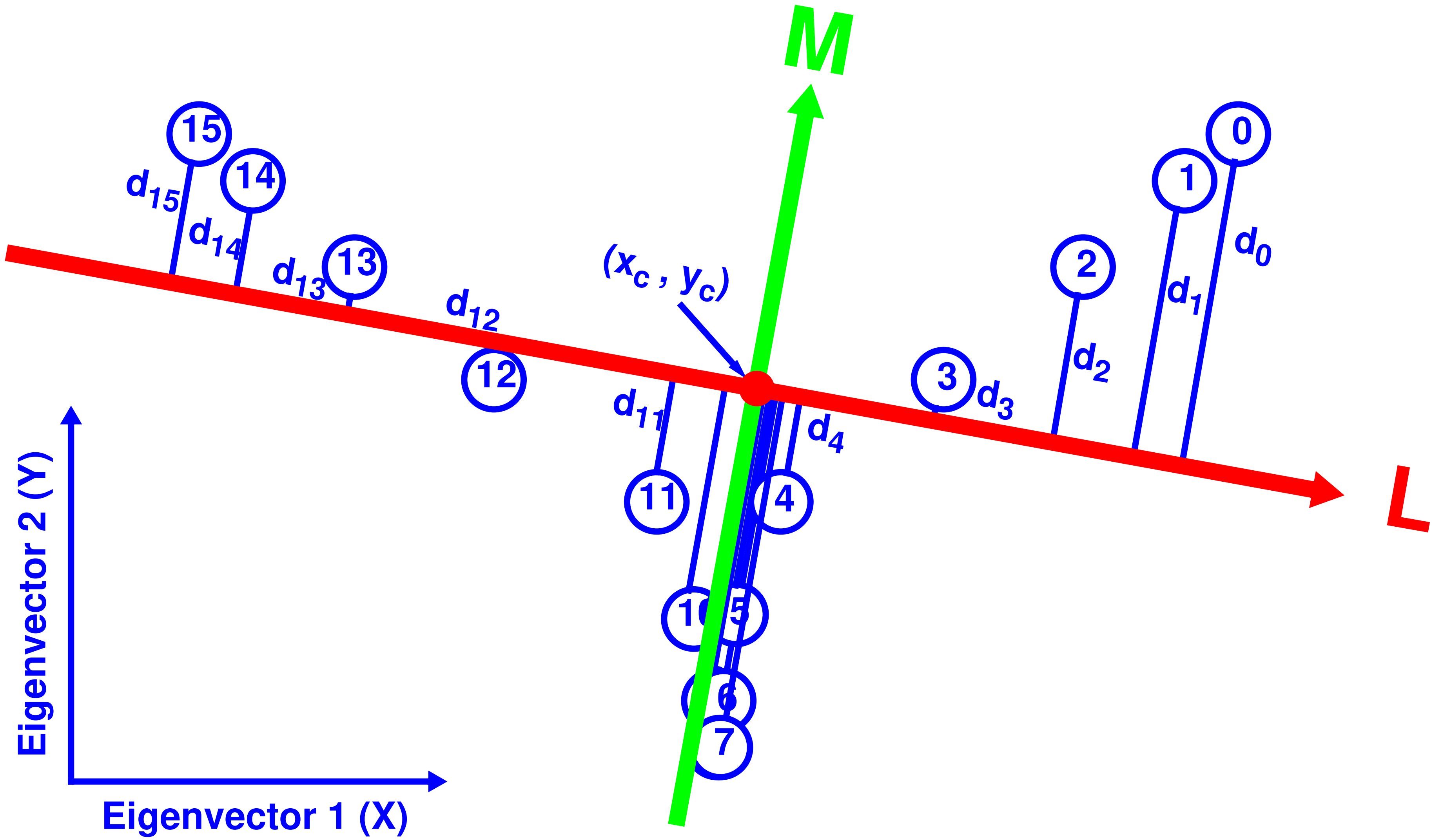
(Guattery and Miller 1995)





0.5
0.3
0.1
–0.1
–0.3
–0.5
–0.5
-0.3
–0.1
0.1
0.3
0.5
Eigenvector 1
Eigenvector 2
15
14
13
12
11
4
5
6
7
3
2
1
0

Inertial Partitioning

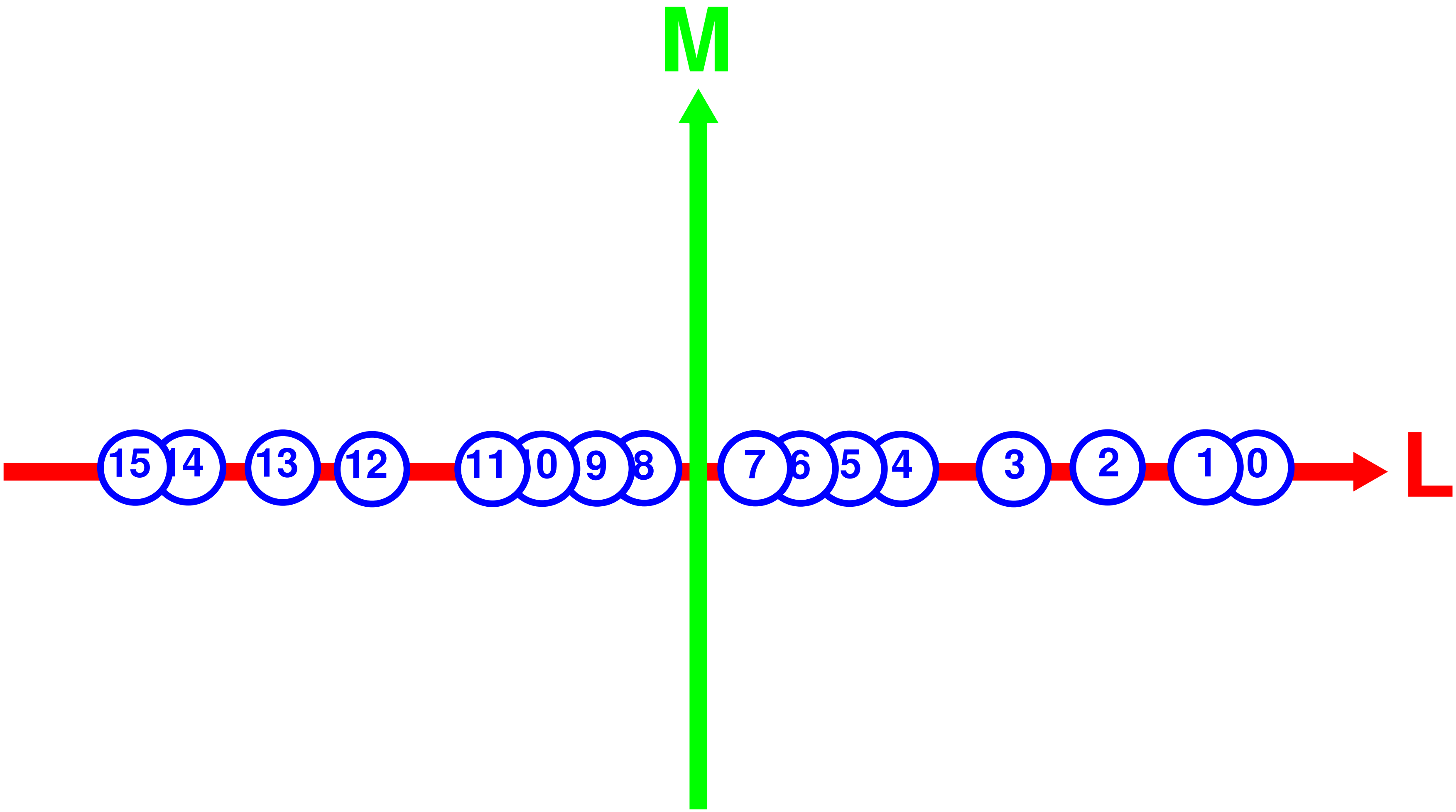


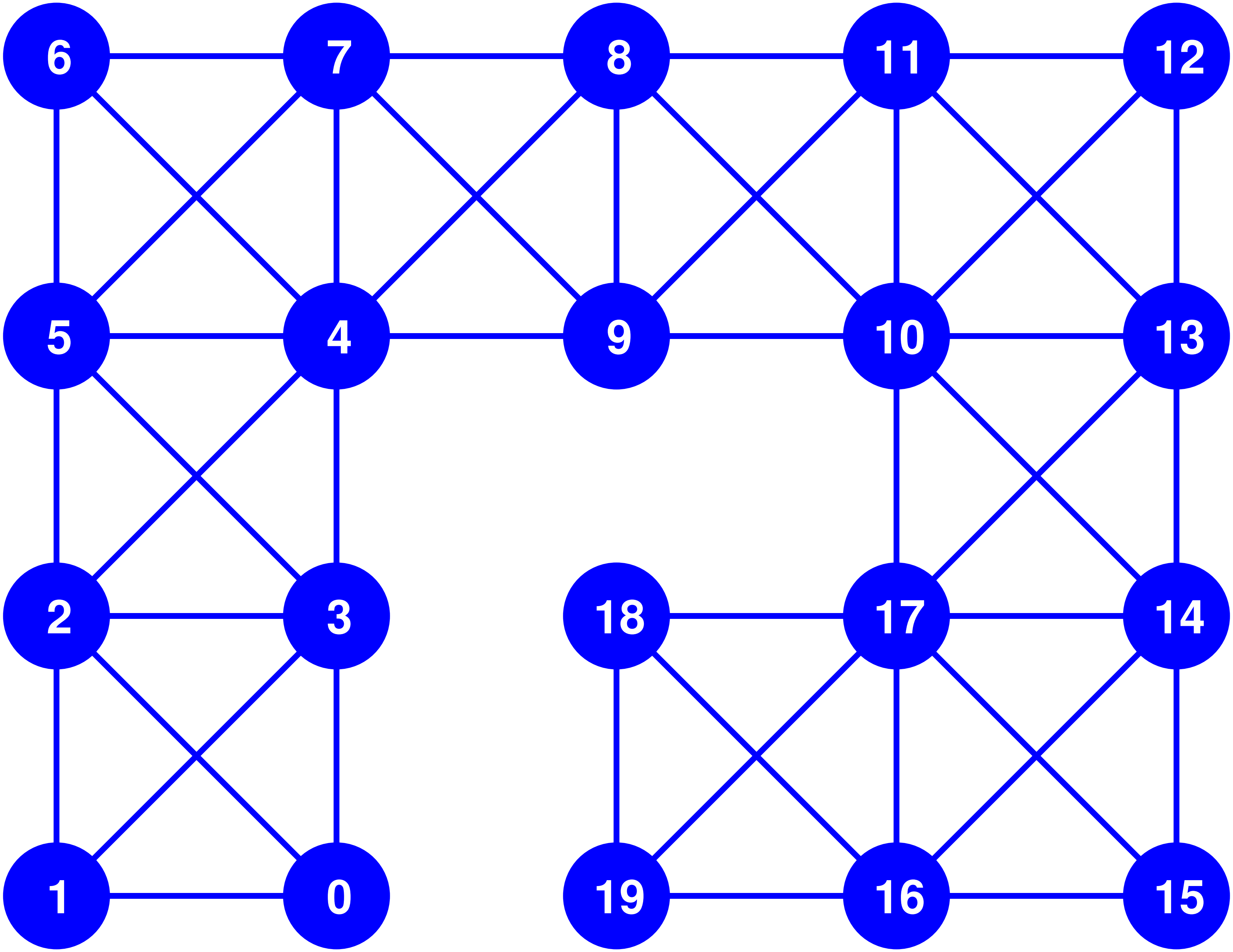
Inertial Partitioning

Parallel sorting

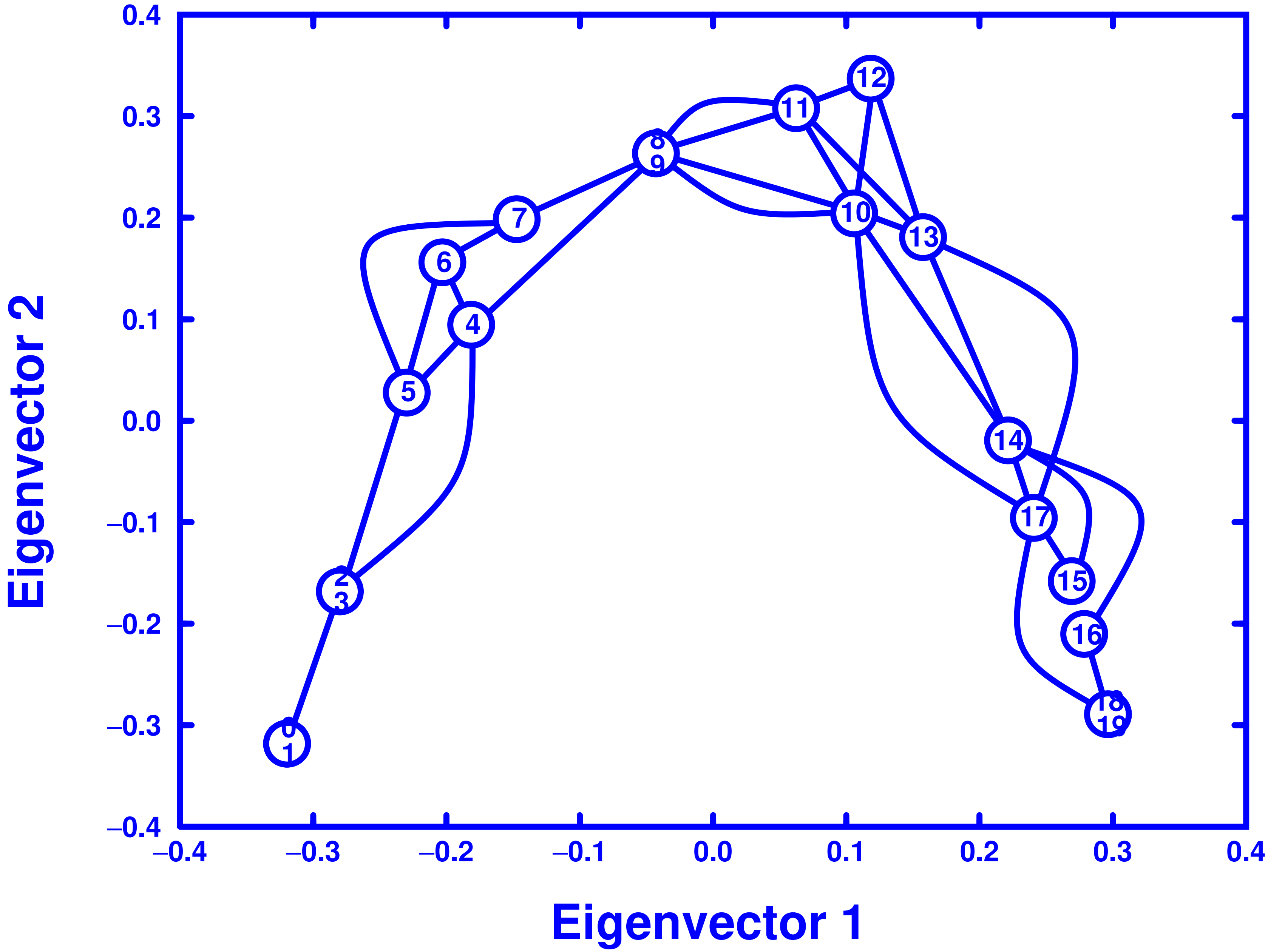
because of the weighted vertices

Inertial Partitioning





6
7
8
11
12
5
4
9
10
13
2
3
18
17
14
1
0
19
16
15


0.4
0.3
0.2
0.1
0.0
–0.1
–0.2
–0.3
–0.4
–0.4
–0.3
–0.2
–0.1
0.0
0.1
0.2
0.3
0.4
Eigenvector 1
Eigenvector 2
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

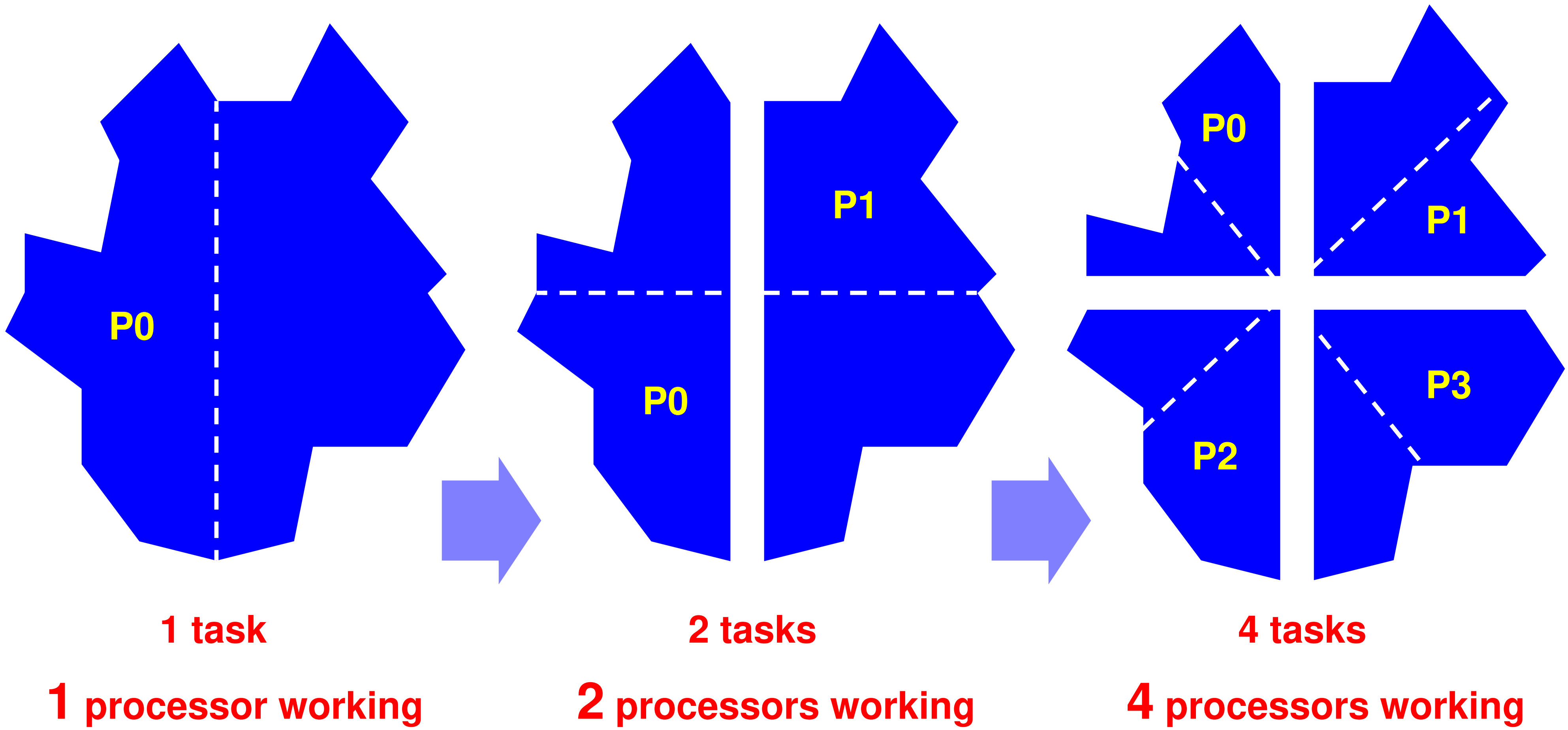
Parallelization

Recursion is not highly parallel!

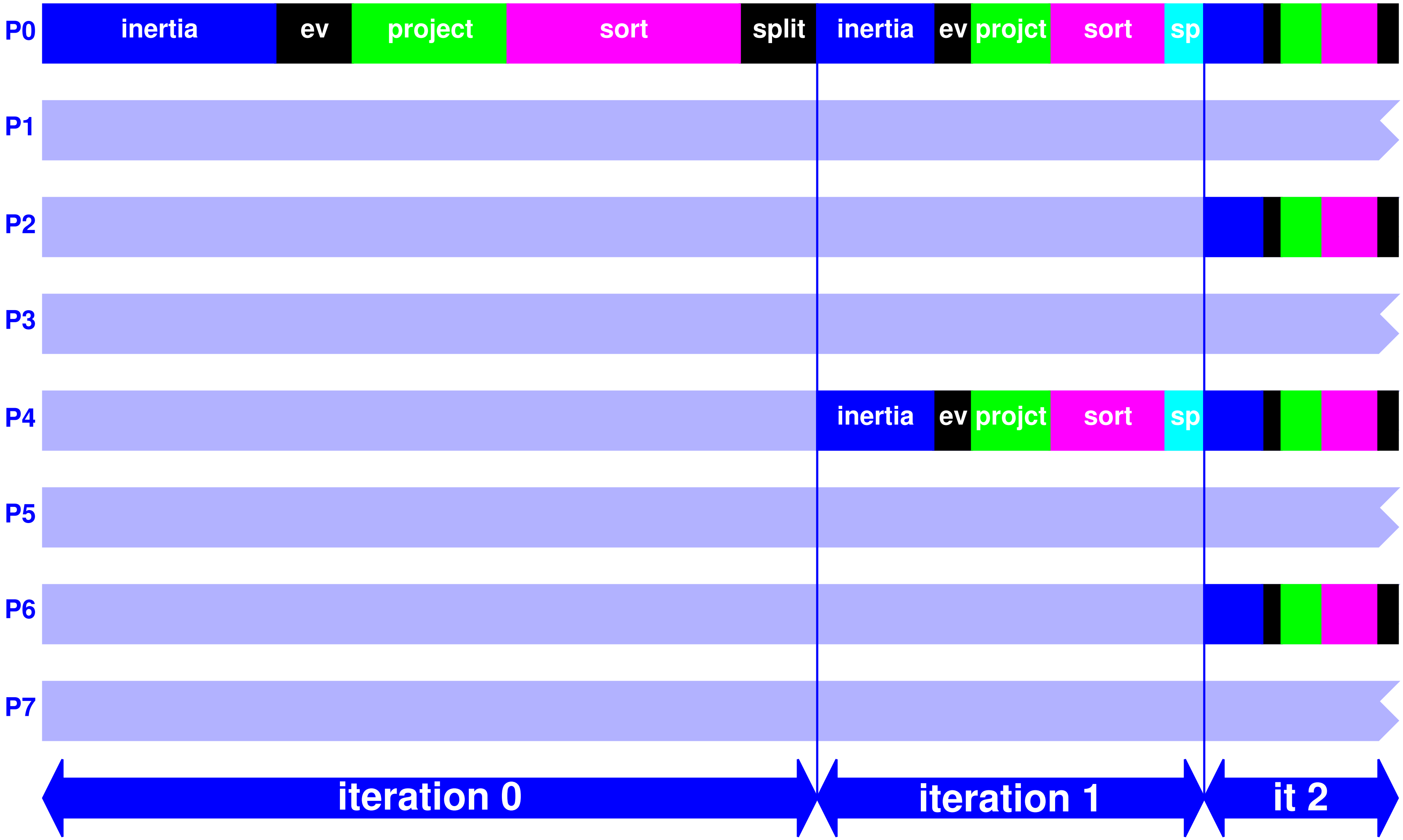
S-HARP uses

- Task parallelism
- Loop parallelism

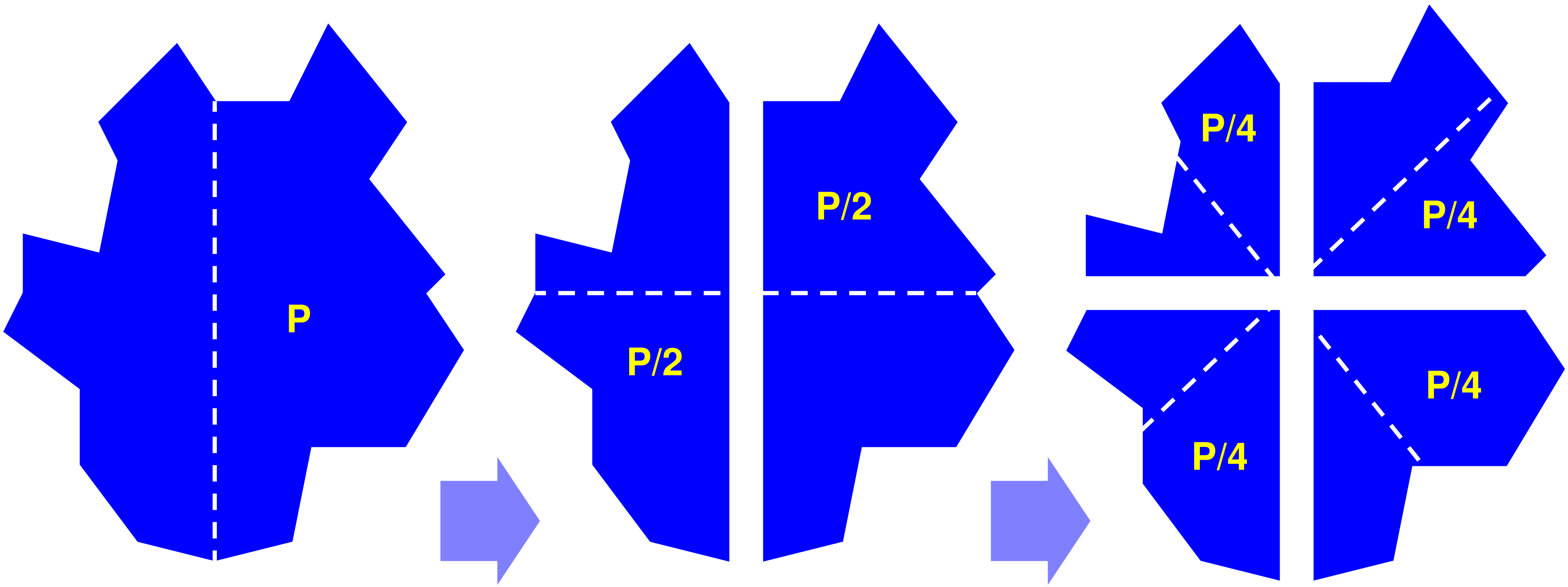
Task Parallelism



Task Parallelism

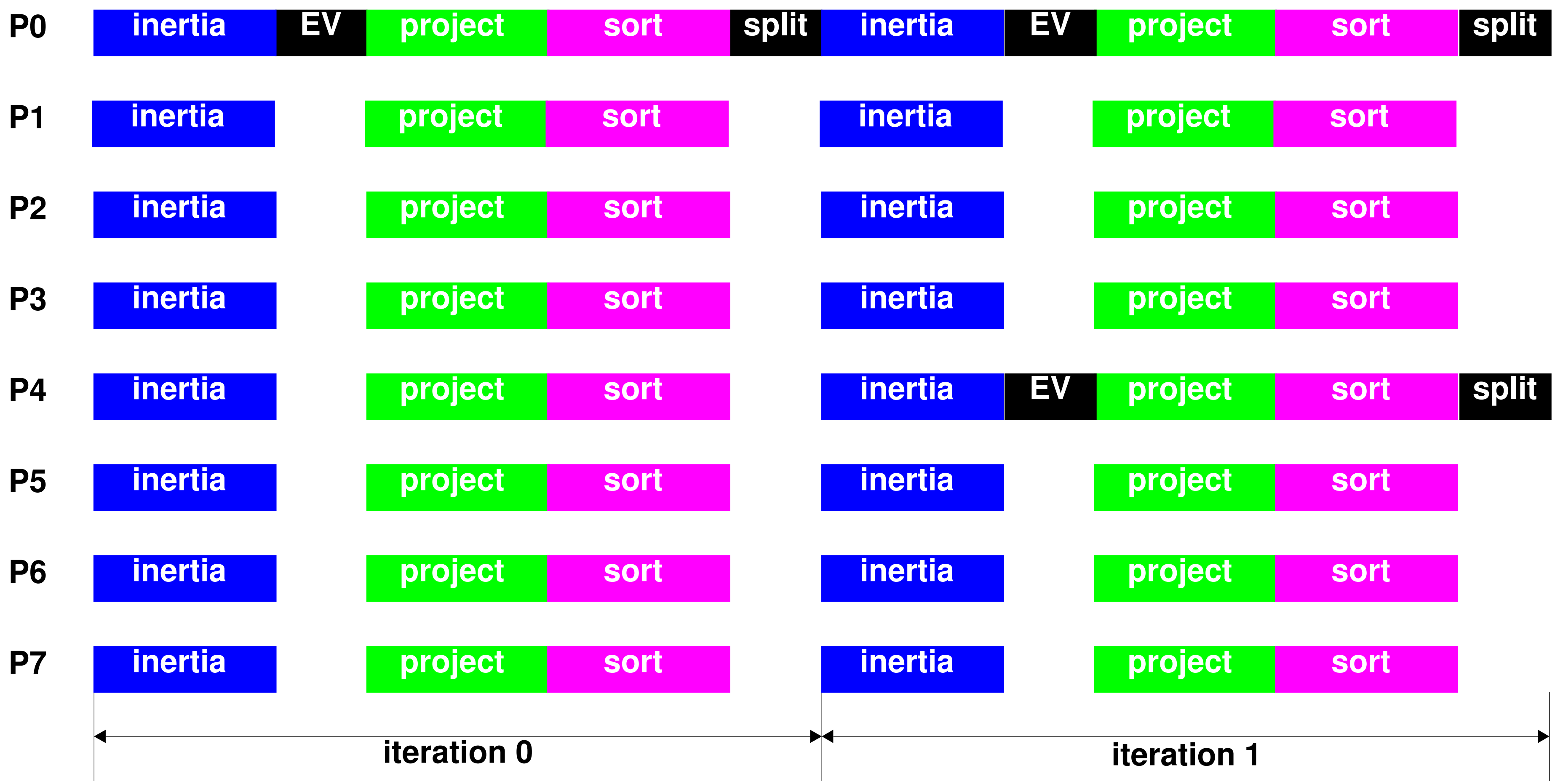


Task+Loop Parallelism



1 task	2 tasks	4 tasks
All processors working	**All** processors working	**All** processors working

Task+Loop Parallelism



The figure is not drawn to scale.

Experiment Settings

- S-HARP implemented in C with MPI.
- Parallel bitonic sorting used.
- Two distributed-memory multiprocessors IBM SP-2 and SGI/Cray T3E.
- Test meshes: over 100000 vertices and 200000 edges.
- Eigenvectors computed on Cray C90 once and for all.
- Compared with ParaMeTiS-1 and Jostle.
  (L. Oliker of RIACS, NASA Ames provided some of the data)

Test Meshes

Name	Type	# of vertices	# of edges	Nature
Spiral	2D	1200	3191	toy grid spiral
Labarre	2D	7959	22,936	aircraft engine
Strut	3D	14,504	57,387	civil engineering
Airfoil	2D	30,269	44,929	four-element air foil
Hsctl	3D	31,736	142,776	high speed civil trans.
Mach95	3D	60,968	118,527	helicopter blade
Ford2	3D	100,196	222,246	model Ford car

Partitioning Time

Ford2 on a 64-processor T3E

P, # of prcrs	Number of partitions *S*							
	2	4	8	16	32	64	128	256
1	0.360	0.829	1.288	1.751	2.193	2.557	2.870	3.172
2	0.221	0.453	0.682	0.913	1.134	1.308	1.464	1.612
4	0.144	0.273	0.388	0.503	0.614	0.702	0.783	0.858
8	0.107	0.186	0.252	0.307	0.362	0.404	0.442	0.478
16	0.090	0.142	0.181	0.217	0.241	0.263	0.286	0.305
32	0.082	0.116	0.145	0.160	0.179	0.185	0.200	0.211
64	0.091	0.121	0.143	0.154	0.169	0.168	0.174	0.184

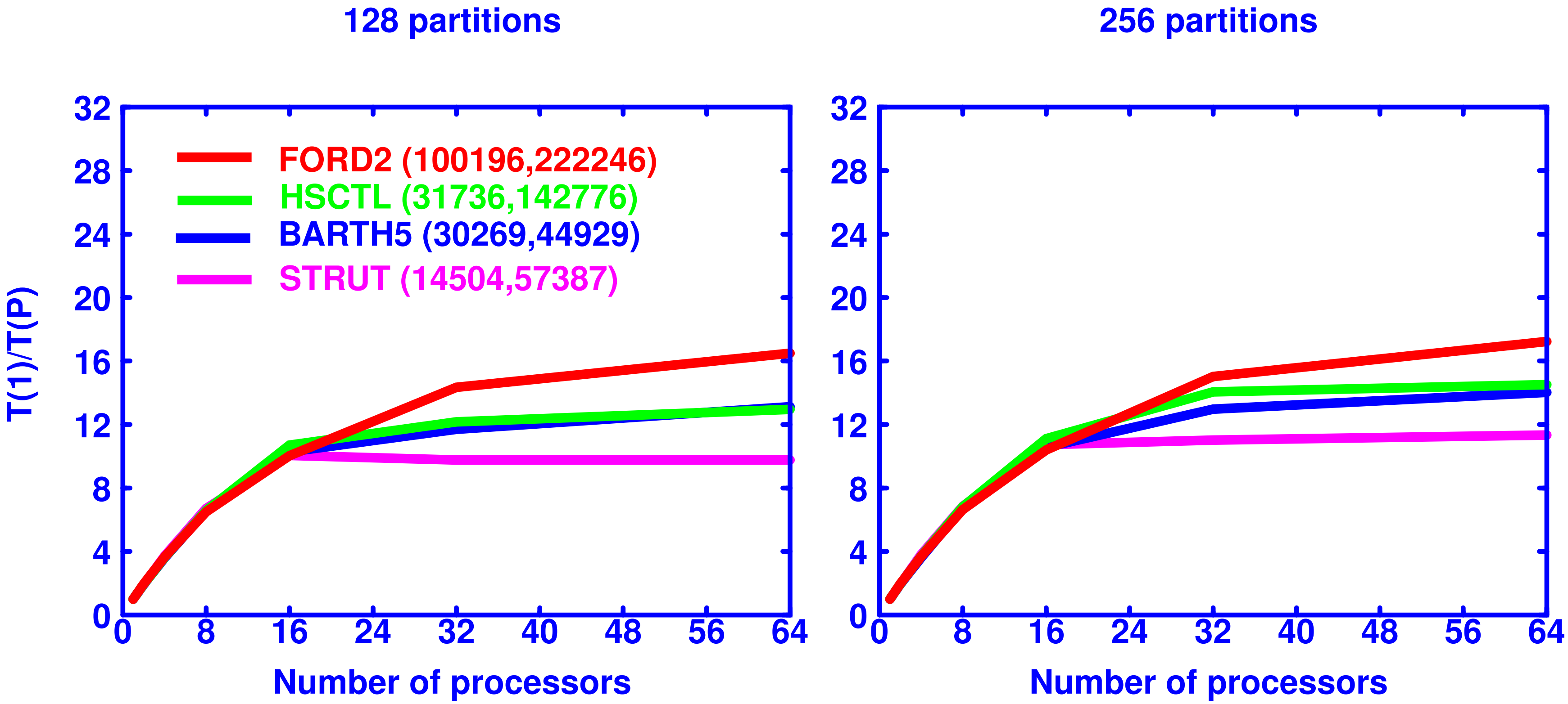
Partitioning Time

Ford2 on a 64-processor SP-2

P, # of prcrs	Number of partitions *S*							
	2	4	8	16	32	64	128	256
1	0.352	0.684	1.023	1.360	1.698	2.040	2.391	2.761
2	0.244	0.407	0.577	0.745	0.915	1.088	1.260	1.446
4	0.186	0.302	0.384	0.469	0.553	0.638	0.725	0.823
8	0.158	0.244	0.301	0.348	0.389	0.430	0.481	0.521
16	0.140	0.209	0.262	0.292	0.315	0.333	0.354	0.378
32	0.127	0.189	0.234	0.261	0.275	0.288	0.295	0.306
64	0.141	0.198	0.239	0.269	0.279	0.287	0.294	0.299

Speedup

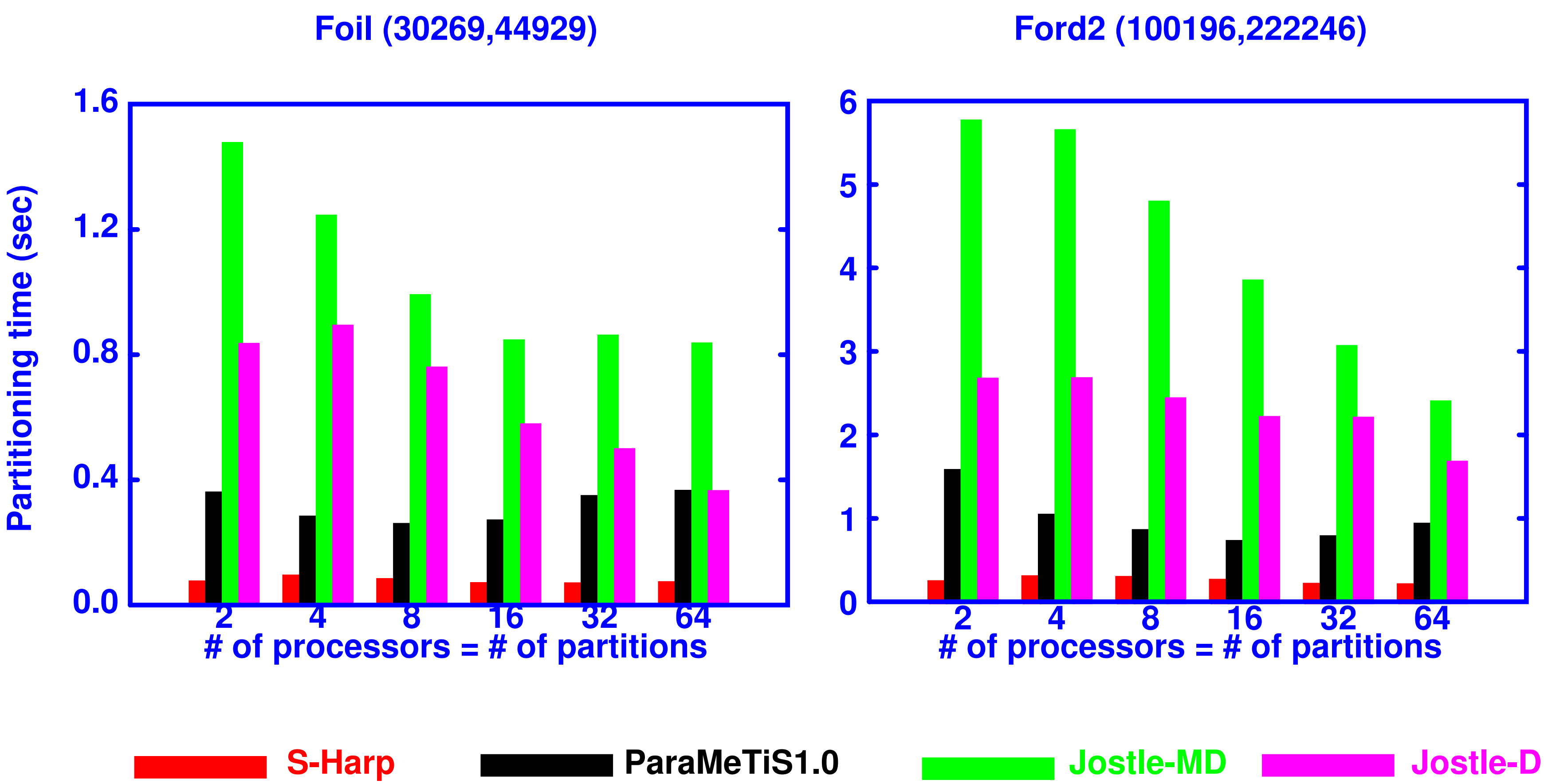
T3E



128 partitions
256 partitions
FORD2 (100196,222246)
HSCTL (31736,142776)
BARTH5 (30269,44929)
STRUT (14504,57387)
T(1)/T(P)
Number of processors
Number of processors
0
4
8
12
16
20
24
28
32
0
8
16
24
32
40
48
56
64

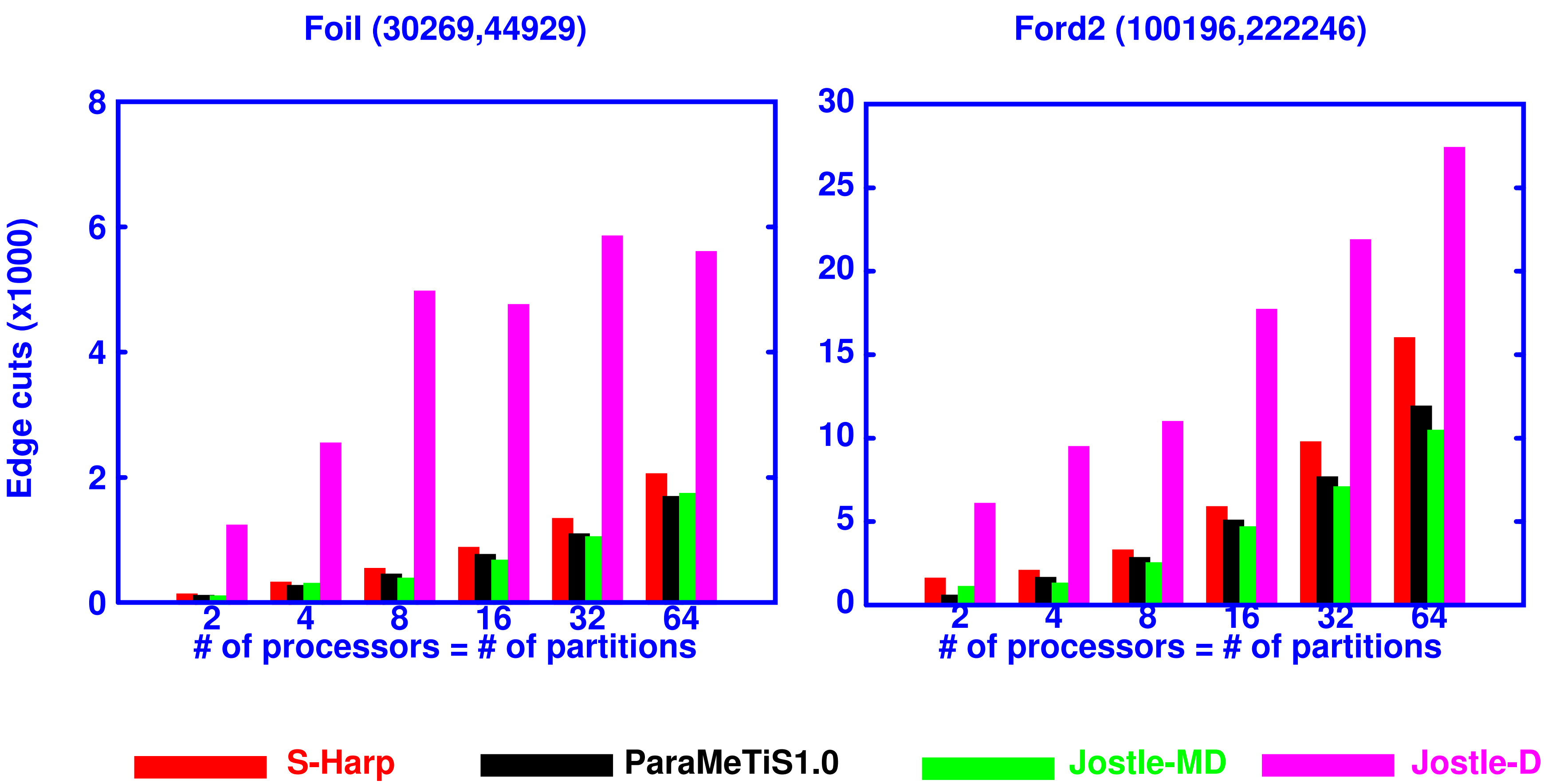
Comparison

Execution time on T3E



Comparison

Edge cuts



Conclusions

- a universal scalable partitioner
- over 100,000 Vs into 256 sets in 0.18 sec on 64 processors
- 17-fold speedup on 64-processors
- 3 to 15 times faster than other dynamic partitioners

S-HARP Home Page

www.cs.njit.edu/sohn/sharp

- ✔ **Papers, meshes, pointers to other partitioners.**
- ✔ **Source code, a user manual, sample meshes will be posted in the future.**

Dynamic Load Balancing Problem?