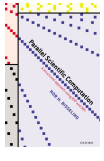
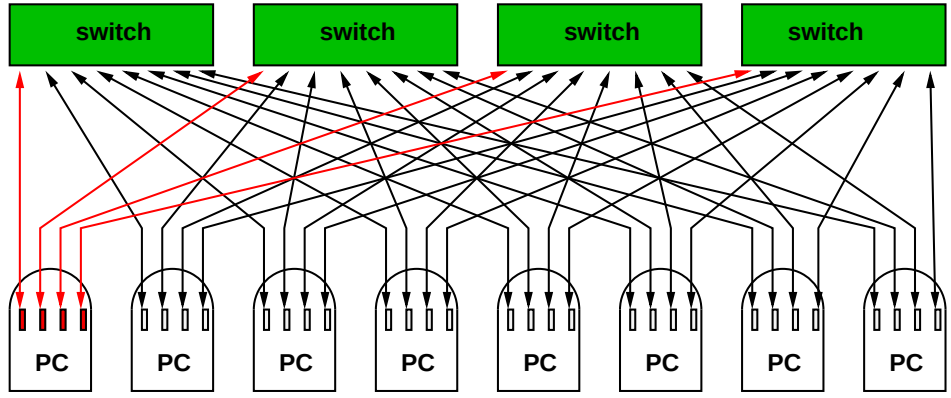
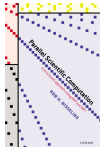
Experimental results on a Beowulf cluster
(PSC §4.10)



Beowulf cluster



- A Beowulf cluster consists of several PCs connected by communication switches.
- We use a cluster of 32 IBM x330 nodes, located at the Physics Department of Utrecht University, part of DAS-2, the 200-node Distributed ASCI Supercomputer built by 5 collaborating Dutch universities.
- Each node contains 2 Pentium-III processors with 1 GHz clock speed, 1 Gbyte of memory, and a local disk.
- Nodes are connected by a Myrinet-2000 network.

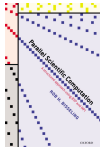


Panda BSP library

p	g	l	$T_{\mathrm{comm}}(0)$
1	1337	7 188	6 767
2	1400	100 743	102 932
4	1401	226 131	255 307
8	1190	440 742	462 828
16	1106	835 196	833 095
32	1711	1 350 775	1 463 009
64	2485	2 410 096	2 730 173

Benchmarked BSP parameters p, g, l and time of a 0-relation ($r = 323$ Mflop/s)

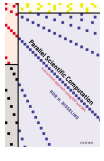
- Experimental BSP library on top of Panda portability layer on top of Myrinet.
- 2 processors per node; for $p \leq 32$, only 1 is used.



Test set of sparse matrices

Matrix	n	nz	Origin
`random20k`	20 000	99 601	random sparse matrix
`amorph20k`	20 000	100 000	amorphous silicon
`prime20k`	20 000	382 354	prime number matrix
`bcsstk32`	44 609	2 014 701	automobile chassis
`cage12`	130 228	2 032 536	DNA electrophoresis

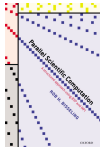
(Subset of the original test set from Section 4.10)



BSP cost for smaller matrices

p	random20k	amorph20k	prime20k
1	199 202	200 000	764 708
2	$102\,586 + 5073g$	$100\,940 + 847g$	$393\,520 + 4275g$
4	$51\,292 + 4663g$	$51\,490 + 862g$	$196\,908 + 5534g$
8	$25\,642 + 3452g$	$25\,742 + 1059g$	$98\,454 + 4030g$
16	$12\,820 + 2152g$	$12\,872 + 530g$	$49\,226 + 3148g$
32	$6\,408 + 1478g$	$6\,434 + 371g$	$24\,612 + 2620g$
64	$3\,202 + 1007g$	$3\,216 + 267g$	$12\,304 + 2235g$

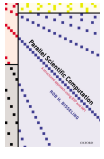
- Matrix and vectors partitioned by Mondriaan.
- Fixed synchronisation cost $4l$ not shown. Since $l \approx 100\,000$ for $p = 2$, matrices must have at least 100 000 nonzeros to make parallelism worthwhile.



BSP cost for larger matrices

p	bcsstk32	cage12
1	4 029 402	4 065 072
2	2 070 816 + 630g	2 093 480 + 10 389g
4	1 036 678 + 786g	1 046 748 + 15 923g
8	518 676 + 842g	523 376 + 16 543g
16	259 390 + 1163g	261 684 + 9 984g
32	129 692 + 917g	130 842 + 6 658g
64	64 836 + 724g	65 420 + 5 385g

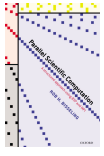
- Same number of nonzeros, but much more communication for cage12.
- This may be due to the high-dimensional underlying structure of cage12, in contrast to bcsstk32, which is only 3D.



Measured execution time (in ms)

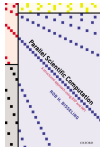
p		random 20k	amorph 20k	prime 20k	bcsstk32	cage12
1	(seq)	9	7	18	71	92
1	(par)	10	8	19	72	96
2		73	13	56	59	205
4		57	16	77	39	228
8		48	15	50	25	226
16		32	11	46	24	128
32		28	17	37	23	87
64		36	29	45	34	73

- Compare `random20k` and `amorph20k`: same size and number of nonzeros, but `amorph20k` much faster.
- Only modest speedups obtained for larger problems.



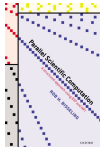
Did we gain something?

- We have shown that for large problem sizes the algorithm scales well.



Did we gain something?

- We have shown that for large problem sizes the algorithm scales well.
- Many research papers on parallel computing end like this, even if the statement is not true.
- The sad truth: we haven't reached the problem size yet where parallel computing becomes worthwhile. The high value $g \approx 1000$ hinders obtaining decent speedups.
- Our main goal should be to understand the results, whatever they may be.



Summary

- Clusters of PCs are cheap supercomputers with tremendous potential.
- BSPlib implementations for such clusters can have great impact. The Panda BSP library is one such library. But networks evolve fast and form a moving target.
- Top of my wish list: new implementations for clusters based on Infiniband, Fast Ethernet, etc., and for dual-core and multi-core PCs. Preferably open-source.
- 20 years from now:
  - Blue Gene, Earth Simulator, DAS-2, and their friends will all be dead
  - I shall be older and perhaps wiser
  - BSP costs like 65 420 $+$ 5 385g + $4l$ for matrix `cage12` will still be meaningful, as predictors for Exaflop/s (10^{18} flop/s) machines.