Evaluating Sparse Matrix-Vector Product on the FinisTerrae Supercomputer

Juan C. Pichel
Galicia Supercomputing Center (CESGA)

Juan A. Lorenzo, Dora B. Heras and José C. Cabaleiro
Univ. of Santiago de Compostela

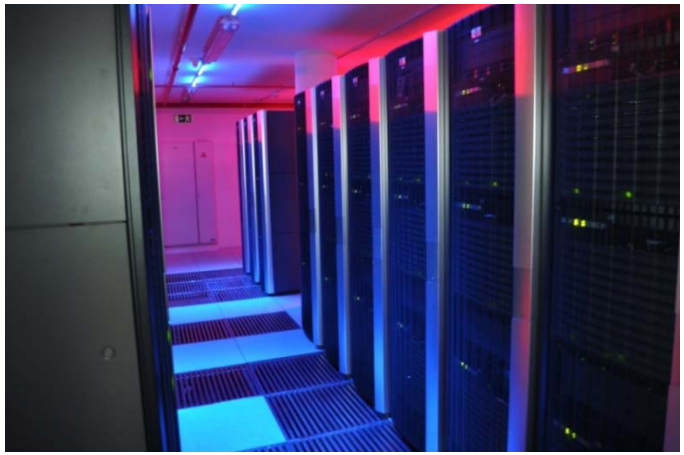
Summary

- Motivation
- FinisTerrae Architecture
- Experimental Conditions
    - Sparse Matrix-Vector Product (SpMV) characteristics
- Performance Evaluation
    - Loop Distribution
    - Memory Affinity
    - Processor Affinity
    - Data Reordering
    - Compiler Options and Power Efficiency
- Conclusions

CESGA

FinisTerrae Architecture

- SMP-NUMA system with more than 2500 processors (rank 427 in TOP500 - November 2008 list)
- 142 HP Integrity rx7640 nodes:
  - 8 1.6GHz Dual-Core Itanium2 (Montvale)
  - 128 GB main memory per node
- HP Integrity SuperDome (*not considered here*):
  - 64 1.6GHz Dual-Core Itanium2 (Montvale)
  - 1 TB main memory
- Interconnect network: Infiniband 4x DDR
- High Performance Storage System: HP SFS
- SuSE Linux Enterprise Server 10

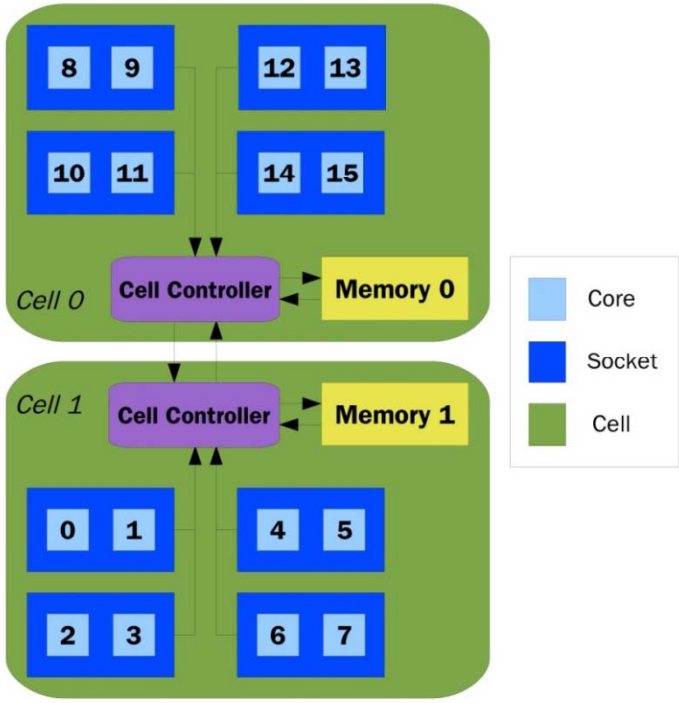


CESGA

FinisTerrae Architecture

- Processors arranged in a SMP-cell NUMA configuration
- Two buses at 8.5 GB/s, each connecting two sockets (four cores) to a 64 GB memory module through a Cell Controller (sx2000 chipset)
- Cell Controller maintains a cache-coherent memory system (directory-based protocol)
- Cell Controller connects both cells through a 34.6 GB/s crossbar
- Theoretical peak bandwidth:
  - Processor-Cell Controller: 17 GB/s
  - Cell Controller-Memory: 17.2 GB/s (four buses)

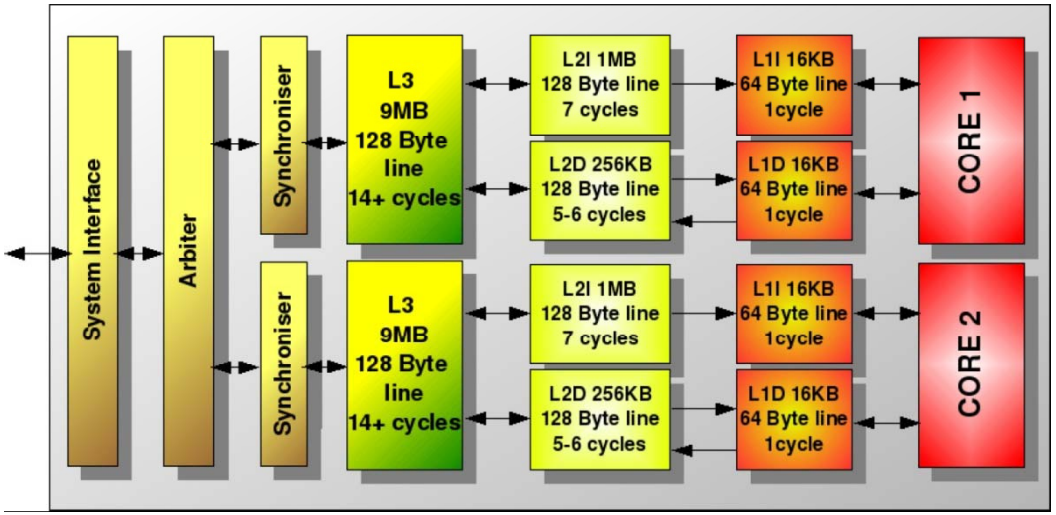
Node rx7640 scheme



CESGA

FinisTerrae Architecture

Itanium2 Montvale scheme



- Two 64 bit cores with three levels of cache per core:
  - L1I and L1D (write-through) are both 16KB and 4-way associative
  - L2I (1MB) and L2D (256 KB) (write-back) are 8-way associative
  - Unified 12-way L3 (9 MB) per core
- 128 General purpose registers and 128 FP registers
- 4 FP operations per cycle. Peak performance 6.4 GFlops/s.
- FP operations bypass the L1. HyperThreading disabled.

CESGA

Experimental Conditions

SpMV Characteristics

```
for (i=0; i < N; i++) {
    reg=0;
    for (j=PTR[i]; j < PTR[i+1]; j++) {
        reg = reg + DA[j]*X[INDEX[j]];
    }
    Y[i]=reg;
}
```

- SpMV is one of the most important kernels in scientific computing
- Notorious for sustaining low fractions of peak processor performance (indirect and irregular memory access patterns)

- Consider the operation: Y = A × X
  - X and Y are dense vectors, and A is a N×M sparse matrix
- A is stored using CSR format: DA (data), INDEX (column indices) and PTR (row pointers)
- Y, DA, PTR and INDEX accesses: high spatial locality
- X is accessed indirectly (through INDEX array): locality properties depend on the sparsity pattern

CESGA

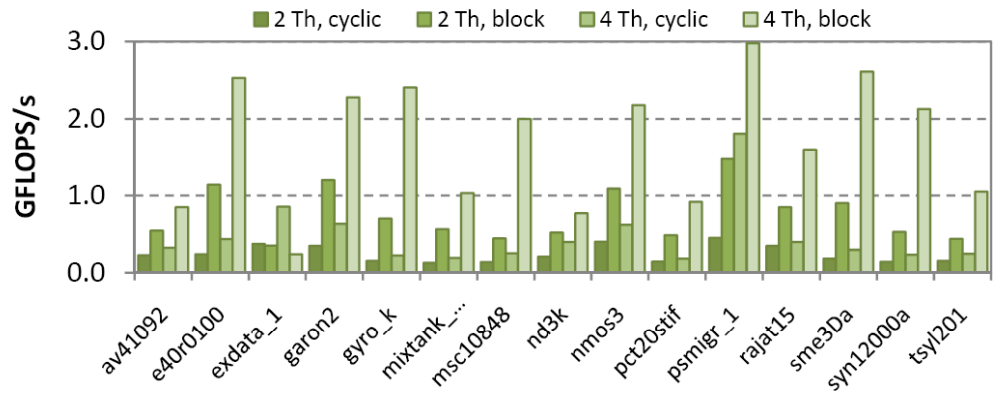
Experimental Conditions

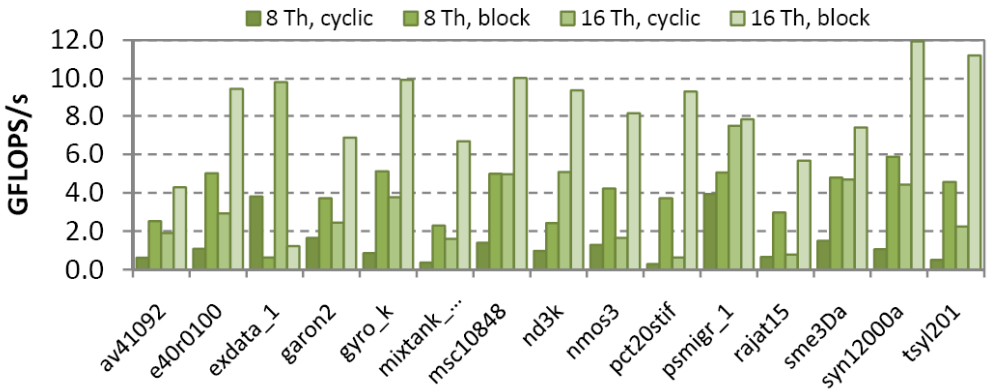
- All the codes were written in C (Intel's 10.0 Linux C compiler)
- OpenMP directives
- Parallel code uses a block distribution. Compiled using –O2 flag.
- Sparse matrices testbed:
  - Fifteen square sparse matrices from different real problems
  - Matrices from the University of Florida (UFL) Sparse Matrix Collection
  - N is the number of rows/columns, and NZ the number of nonzero elements

	N	N_Z		N	N_Z
av41092	41092	1683902	nmos3	18588	386594
e40r0100	17281	553562	pct20stif	52329	2698463
exdata_1	6001	2269501	psmigr_1	3140	543162
garon2	13535	390607	rajat15	32761	443573
gyro_k	17361	1021159	sme3Da	12504	874887
mixtank_new	29957	1995041	syn12000a	12000	1436806
msc10848	10848	1229778	tsyl201	20685	2454957
nd3k	9000	3279690			

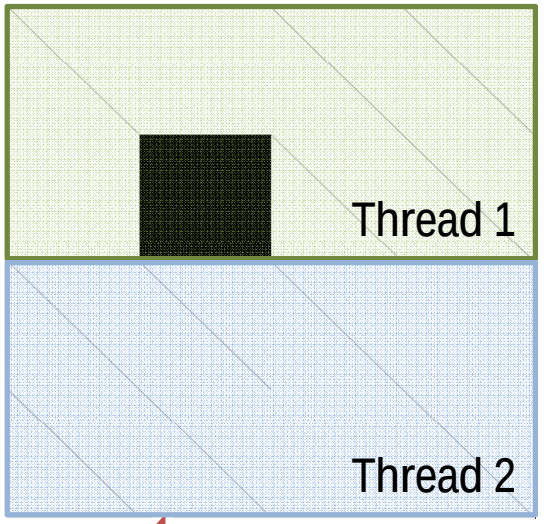
CESGA

Performance Evaluation





Loop Distribution

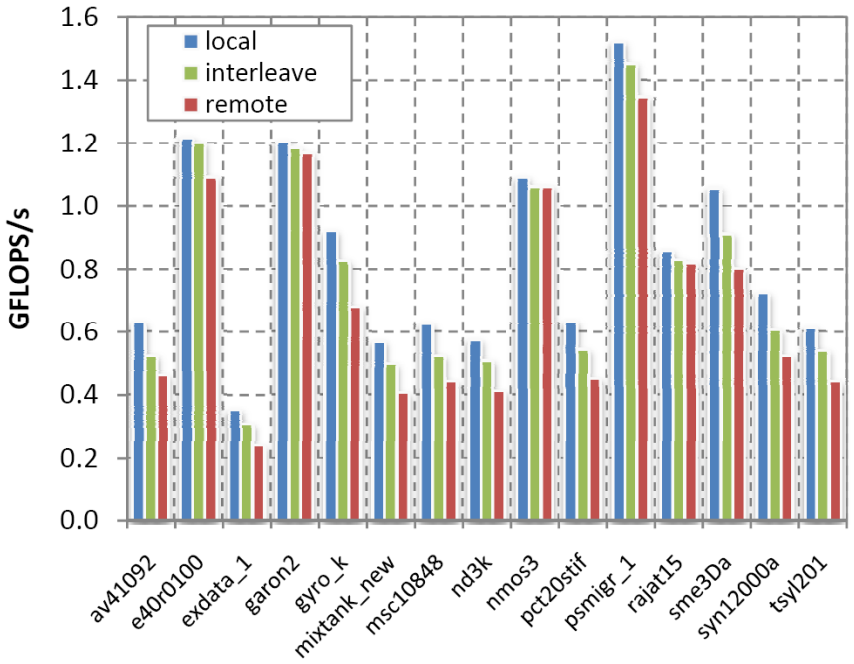


Load balancing Problems

- Block distribution outperforms cyclic one (except for *exdata_1*)
- In most of the cases block distribution is better using less threads
- Block distribution: threads work with disjoint parts of the matrix (fits better in the cache hierarchy of the Itanium2 processor)

CESGA

Performance Evaluation



- Example using two threads mapped to cores 8 and 12 (*libnuma* library)
- Remote memory: important degradation (average decrease 20.7%)
- Interleave policy: degradation about 10% (strong dependence with the pattern: e40r0100 – local, nmos3 –remote)

Memory Affinity

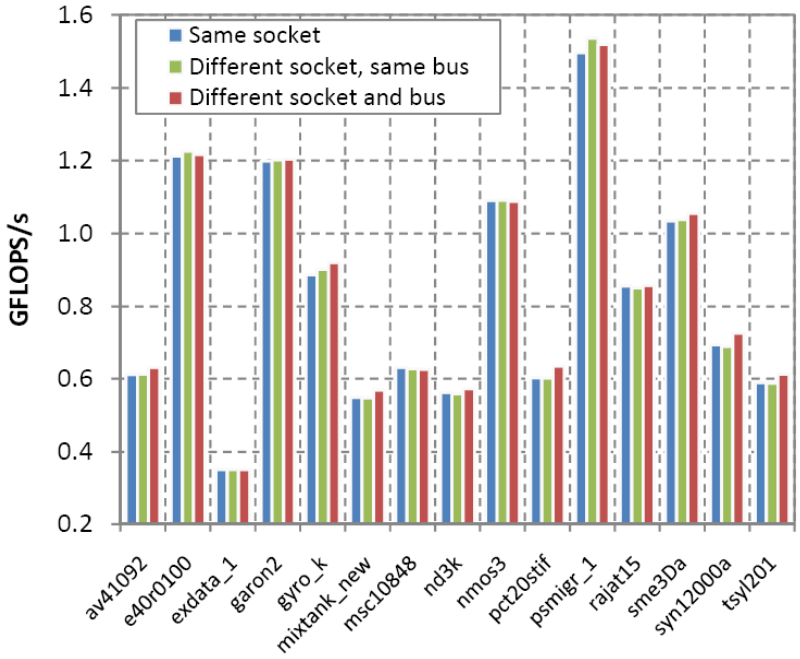
NUMA configuration:

- Local memory: threads and data in the same cell
- Remote memory: threads and data in different cells
- Interleave policy: 50% addresses on the same cell of the requesting processor, 50% on the other cell (round-robin)

CESGA

Performance Evaluation

Processor Affinity



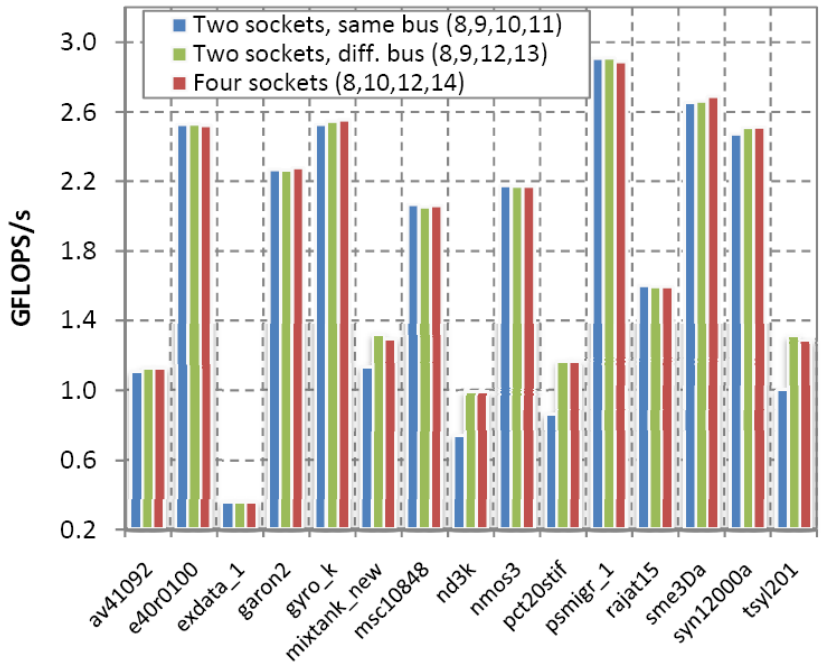
Mapping threads to the same socket (2 threads):

- Low influence of mapping threads to the same socket and different socket-same bus (0.2% in average)
- Some improvements when mapping to cores that do not share bus (up to 5%, specially for big matrices – tsyl201)
- Therefore, bus contention has some impact on the performance

CESGA

Performance Evaluation

Processor Affinity

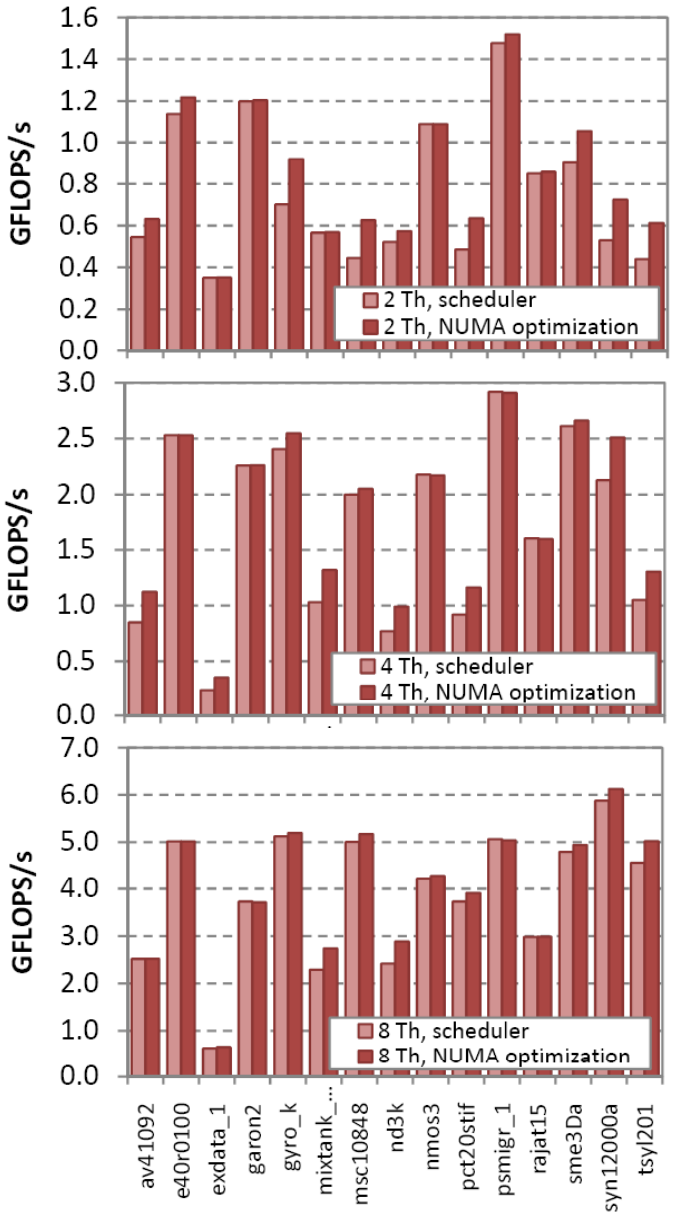


Mapping threads to different sockets (4 threads):

- Better performance mapping threads to sockets that do not share the bus (8% in average, big matrices up to 30%)
- No significant differences between using four sockets or two sockets in different bus

CESGA

Performance Evaluation



Processor Affinity

OS scheduler Vs. Explicit Mapping:

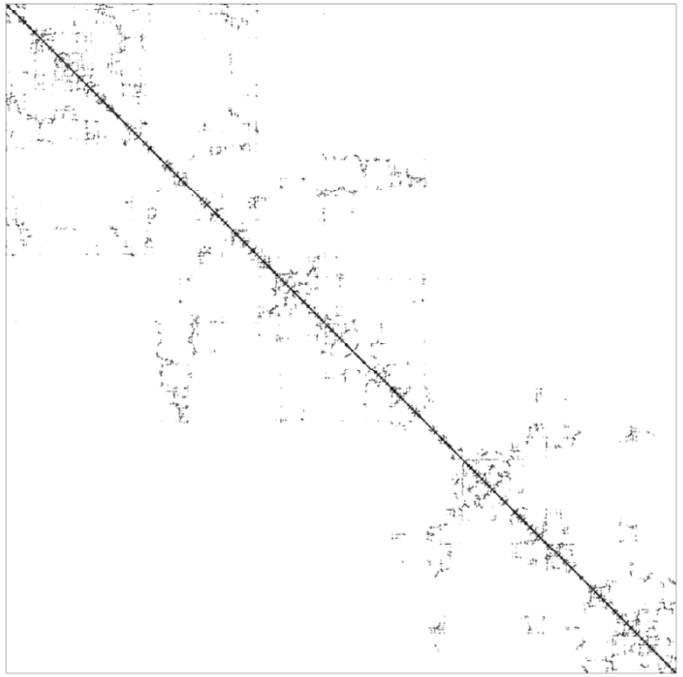
- Only best configurations are displayed (8 threads are mapped to the same cell)
- NUMA optimization has a great impact in most of the cases:
  - Up to 40% for matrix *msc10848*
  - Effects more visible for big matrices (mixtank, nd3k and tsyl201)
  - Average improvement:
    - 2 Threads: 15.6%
    - 4 Threads: 14.1%
    - 8 Threads: 4.7%

CESGA

Performance Evaluation

Data Reordering - Locality

- SpMV is characterized by irregular accesses. It tends to have low spatial and temporal localities
- Study the influence of the memory hierarchy
- Evaluation of a locality optimization technique for multicore processors:
  - Technique consists of reorganizing data guided by a locality model
  - The goal is to increase the grouping of elements nonzero elements in the matrix pattern:
    - Elements closer in the same row: X spatial locality
    - Elements closer in the same column: X temporal locality



CESGA

Performance Evaluation

Data Reordering - Locality

	Performance (GFlops/s)							
	2 Th		4 Th		8 Th		16 Th	
	Orig.	Reord.	Orig.	Reord.	Orig.	Reord.	Orig.	Reord.
av41092	**0.54**	0.51	**0.85**	0.78	**2.52**	0.83	**4.29**	2.06
e40r0100	1.14	**1.16**	2.53	**2.57**	5.02	**5.09**	9.46	**9.52**
exdata_1	0.35	**0.46**	0.24	**1.24**	0.63	**5.06**	1.22	**11.40**
garon2	1.20	**1.21**	2.28	**2.39**	3.73	**4.68**	6.88	**8.98**
gyro_k	0.70	**0.82**	2.41	**2.43**	5.12	**5.32**	9.92	**9.96**
mixtank_new	0.57	**0.58**	1.03	**1.49**	2.29	**4.84**	6.68	**10.38**
msc10848	0.45	**0.63**	2.00	**2.52**	5.00	**6.18**	10.03	**11.73**
nd3k	0.52	0.52	0.77	**0.88**	2.42	**3.12**	9.39	**10.25**
nmos3	1.09	**1.11**	2.18	**2.19**	4.22	**4.30**	8.14	**8.15**
pct20stif	0.49	**0.51**	0.92	**1.01**	3.73	3.73	9.33	**9.60**
psmigr_1	**1.48**	1.25	**2.98**	2.66	**5.06**	4.16	**7.83**	7.51
rajat15	0.85	0.85	1.60	**1.70**	2.98	**3.31**	5.67	**6.21**
sme3Da	0.90	**0.95**	2.61	**2.76**	4.79	**5.61**	7.40	**10.53**
syn12000a	0.53	**0.56**	2.13	**2.20**	5.88	5.88	11.93	**11.95**
tsyl201	0.44	**0.48**	1.05	**1.06**	4.56	**4.75**	11.20	**11.84**

- As the number of threads increases, the performance improvement become more important:
  - 2 Threads: 1.1%
  - 4 Threads: 9.0%
  - 8 Threads: 15.1%
  - 16 Threads: 17.2%
- With the exception of *av41092* and *psmigr_1*

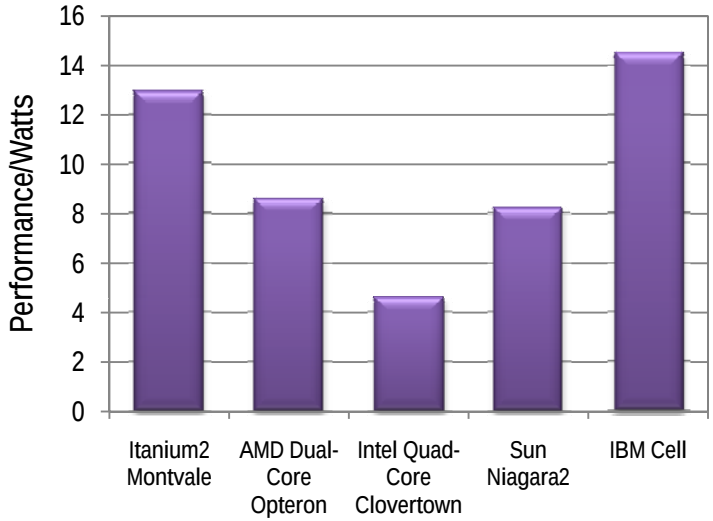
CESGA

Performance Evaluation

Compiler Options & Power Efficiency

- Evaluation of the code using:
  - More aggressive optimizations (-O3): -O2 plus prefetching, scalar replacement, and loop and memory access transformations
  - Optimizations for the specific CPU (-mtune=itanium2-p9000)
  - Best option for half of the matrices is –O2
  - Noticeable improvements in the rest of cases (3x, using two threads for *msc10848* and *tsyl201*)
  - Low influence of the specific CPU flag, about 1%

- Power efficiency:
  - Calculation of the MFlops-Watts ratio
  - Itanium2 Montvale: 12.9
  - Only is beaten by IBM Cell



CESGA

Performance Evaluation

Summary

	Performance (GFlops/s)							
	2 Th		4 Th		8 Th		16 Th	
	Orig.	Opt.	Orig.	Opt.	Orig.	Opt.	Orig.	Opt.
av41092	0.54	**0.72**	0.85	**1.12**	2.52	**2.53**	**4.29**	-
e40r0100	1.14	**1.23**	2.53	**2.55**	5.02	**5.04**	9.46	**9.52**
exdata_1	0.35	**1.15**	0.24	**2.09**	0.63	**8.61**	1.22	**17.98**
garon2	1.20	**1.22**	2.28	**2.40**	3.73	**4.69**	6.88	**8.98**
gyro_k	0.70	**0.92**	2.41	**2.57**	5.12	**5.40**	9.92	**9.95**
mixtank_new	0.57	**0.64**	1.03	**1.58**	2.29	**5.17**	6.68	**10.62**
msc10848	0.45	**1.19**	2.00	**3.35**	5.00	**7.11**	10.03	**13.38**
nd3k	0.52	**1.26**	0.77	**1.51**	2.42	**4.66**	9.39	**14.92**
nmos3	1.09	**1.10**	2.18	**2.21**	4.22	**4.30**	8.14	**8.15**
pct20stif	0.49	**0.65**	0.92	**1.20**	3.73	**4.18**	9.33	**9.60**
psmigr_1	1.48	**1.80**	2.98	**3.42**	5.06	**5.68**	7.83	**10.07**
rajat15	0.85	**0.86**	1.60	**1.70**	2.98	**3.33**	5.67	**6.21**
sme3Da	0.90	**1.06**	2.61	**2.80**	4.79	**5.62**	7.40	**10.67**
syn12000a	0.53	**1.24**	2.13	**3.30**	5.88	**7.06**	11.93	**13.56**
tsyl201	0.44	**1.17**	1.05	**1.85**	4.56	**6.98**	11.20	**14.01**

- SpMV performance:
  - Average: 1.1GFlops/s, 2.2 GFlops/s, 5.4 GFlops/s and 10.7 GFlops/s respectively
  - Improvement: 44.2%, 31.5%, 38.6% and 35.1% with respect to the non-optimization case
  - About 12% of peak performance for Itanium2 Montvale cores (that is, 6.4 GFlops/s)
  - Speedup (w.r.t 2 threads case): 3.4x, 9x and 18.4x (4, 8 and 16 threads)

CESGA

Conclusions

- SpMV has been evaluated on the FinisTerrae supercomputer.
- Influence of data and thread allocation:
  - For threads mapped to cores in the same cell, data allocated in local memory as expected
  - If cores in both cells are used, interleave memory makes sense due to accesses to remote memory are very costly
  - Bus contention have an impact on the SpMV performance
  - Explicit data and thread allocation outperforms OS scheduler
- Influence of the memory hierarchy:
  - A locality optimization technique specially tuned for multicores has been evaluated
  - Taking advantage of Itanium2 is critical, accesses to memory incur in a high penalty in comparison with other systems
- Different compiler optimization options and parallelization strategies were studied
- Power efficiency is considered:
  - Itanium2 Montvale processors near the top
- After study, a set of optimization were applied to SpMV: noticeable improvements with respect to naïve implementation

CESGA


Thank you!
Questions?
CESGA