Der UltraSPARCT1 Prozessor ("Niagara") Erste Erfahrungen

25. Treffen des ZKI Arbeitskreises Supercomputing
31. März 2006, Köln

Dieter an Mey, Samuel Sarholz, Alexander Spiegel, Christian Terboven

Rechen- und Kommunikationszentrum der RWTH Aachen

Overview

- **Processor Architecture**
- **System Architecture**
- **Memory Performance**
- **Applications**

Overview

- **Processor Architecture**
- **System Architecture**
- **Memory Performance**
- **Applications**

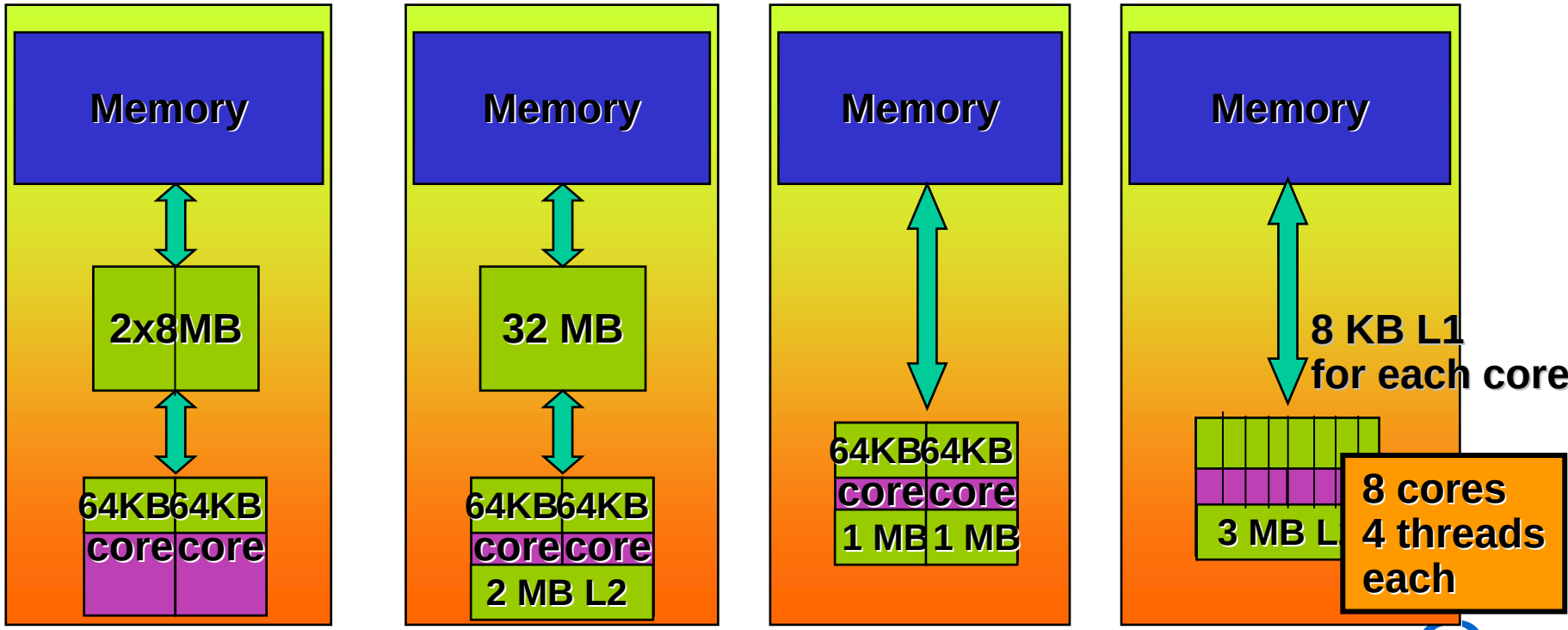
Multi Core-Processors

UltraSPARC IV
1.2 GHz
130 nm
~66 Mio trans.
108 Watt

UltraSPARC IV+
1.5 GHz
90 nm
295 Mio trans.
90 Watt (?)

Opteron 875
2.2 GHz
90 nm
199 mm²
233 Mio trans.
95 Watt

UltraSPARC T1
1.0 GHz
90 nm
378 mm²
300 Mio trans.
72 Watt



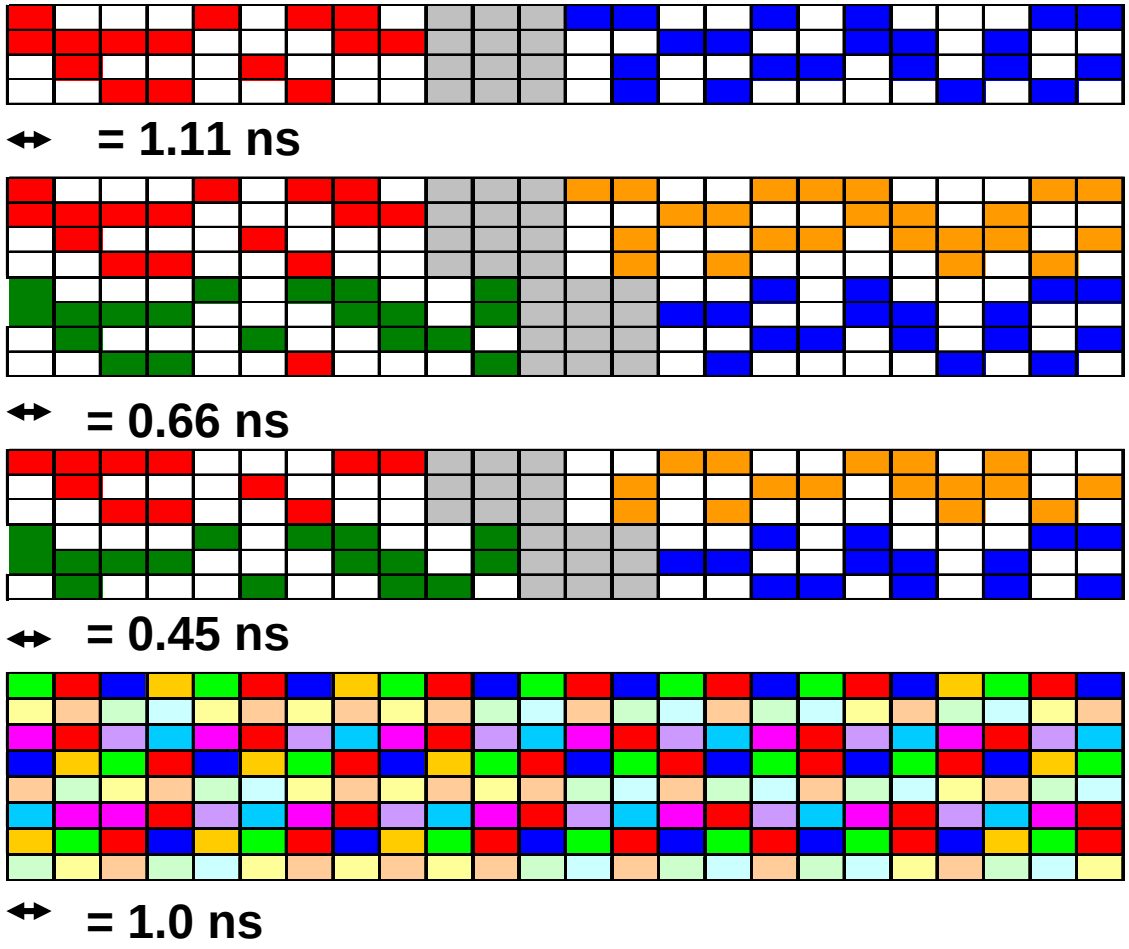
UltraSPARC T1 („Niagara“), D. an Mey, et.al.

Terminology – multi core processors

- **Thread** = LWP = light weight process (OS view)
  Thread = Instruction Stream (HW view)
- **Processor** = Central Processing Unit = CPU
  = the part of a computer that interprets instructions and processes data
- **Multi-Core Processors**: #CPU > #processor chips = # sockets
- **Chip(-level) multithreading (CMT)**
  executes multiple threads within one processor **chip (or core?)** at the same time possibly with time-slicing.
- **Chip(-level) multiprocessing (CMP)** :
  multiple processor "cores" are included in the same integrated circuit
- **Simultaneous multithreading (SMT):**
  Issue multiple instructions from multiple threads in one cycle in one core. (Intel HyperThreading)
- What is an **n-way** (SMP) system**?**

UltraSPARC T1 („Niagara“), D. an Mey, et.al.

Chip Level Parallelism



UltraSPARC III
superscalar, single core
4 sparc v9 instr/cycle
1 active thread per core

UltraSPARC IV+, CMP
superscalar, dual core
2 x 4 sparc v9 instr/cycle
1 active thread per core

Opteron 875, CMP
superscalar, dual core
2 x 3 x86 instr/cycle
1 active thread per core

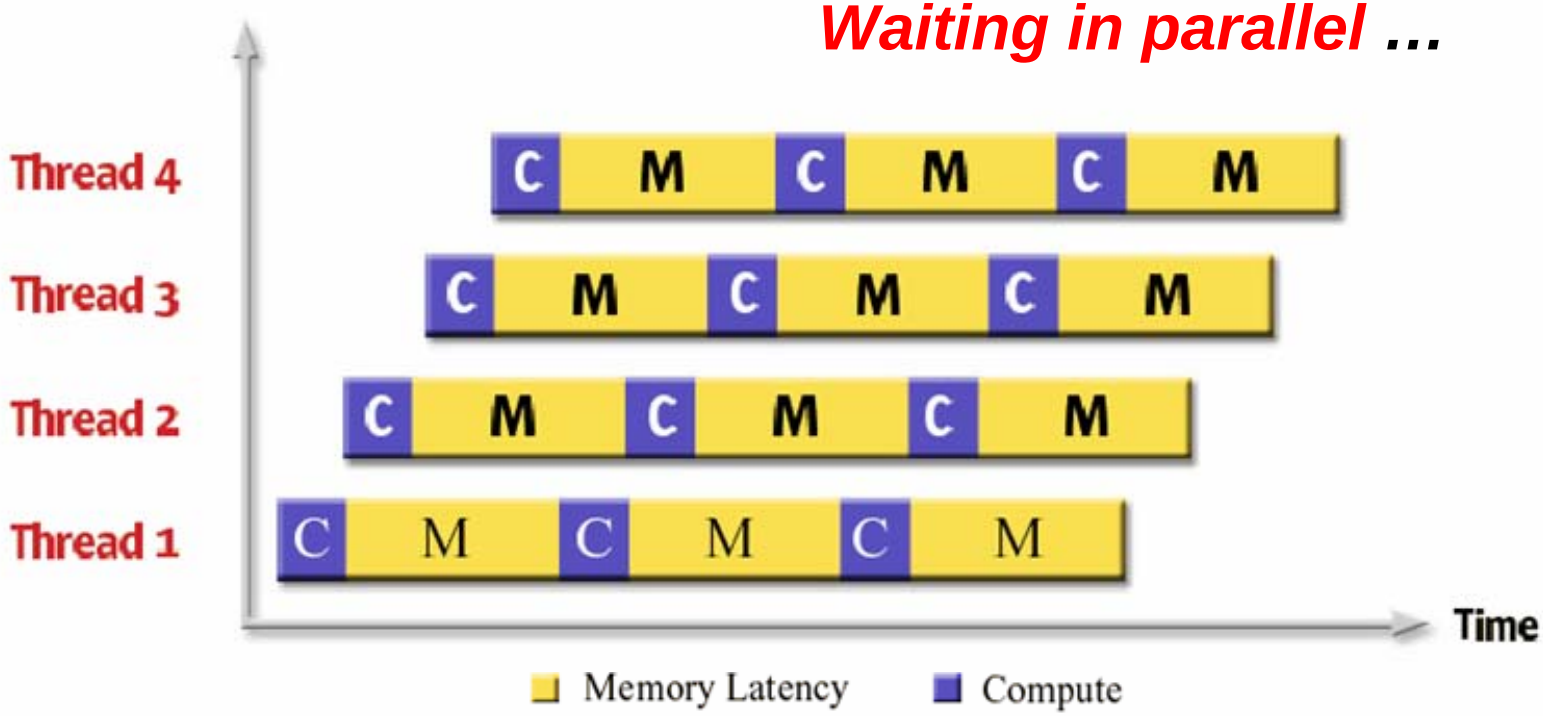
UltraSPARC T1, CMP+CMT
Single issue, 8 cores
8 x 1 sparc v9 instr/cycle
4 active threads per core
context switch comes for free

What to do with all these Threads …

Chip Multithreading (CMT)

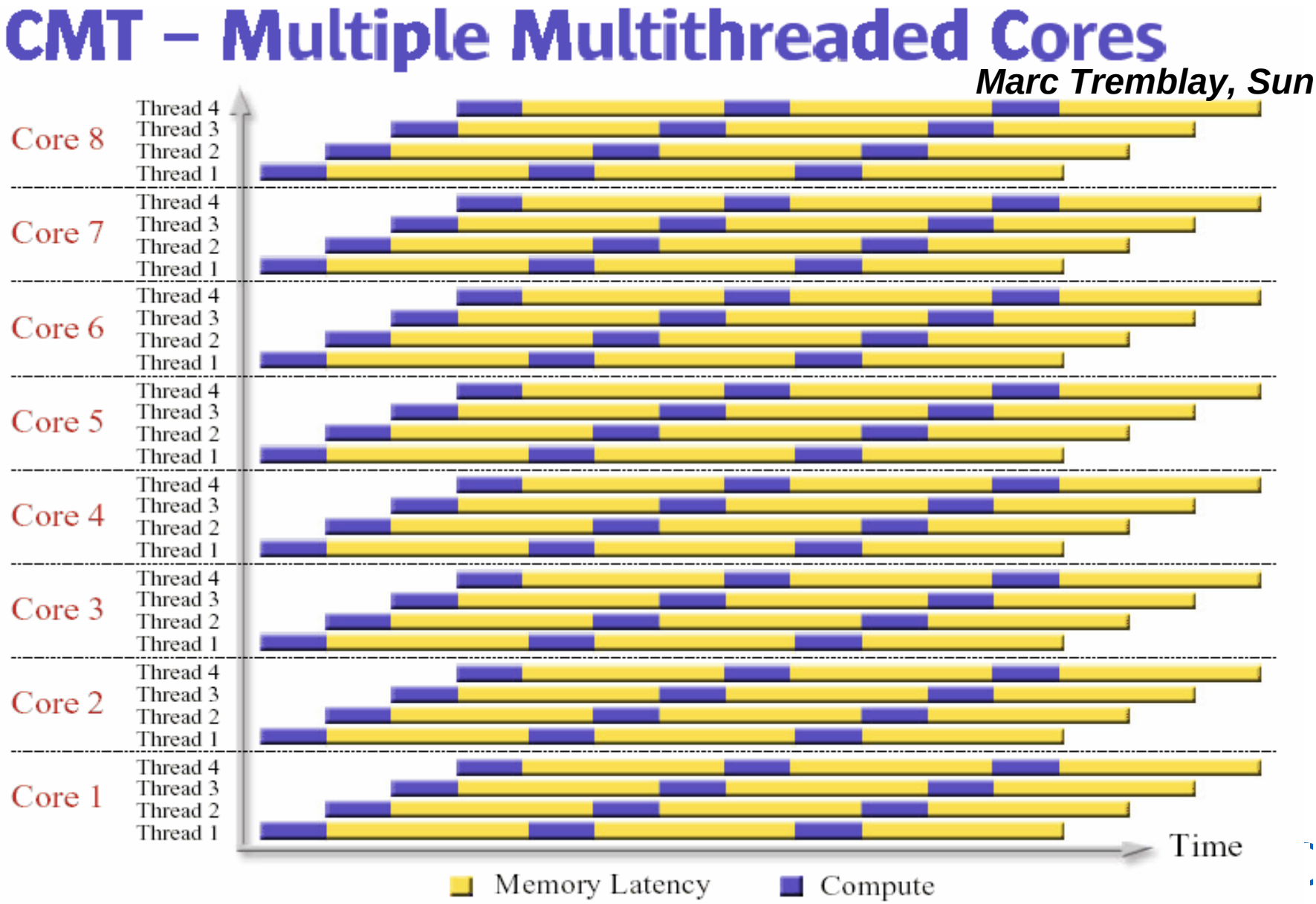
Marc Tremblay, Sun

Waiting in parallel …



UltraSPARC T1 („Niagara“), D. an Mey, et.al.

What to do with all these Threads …



Comparing Processors …

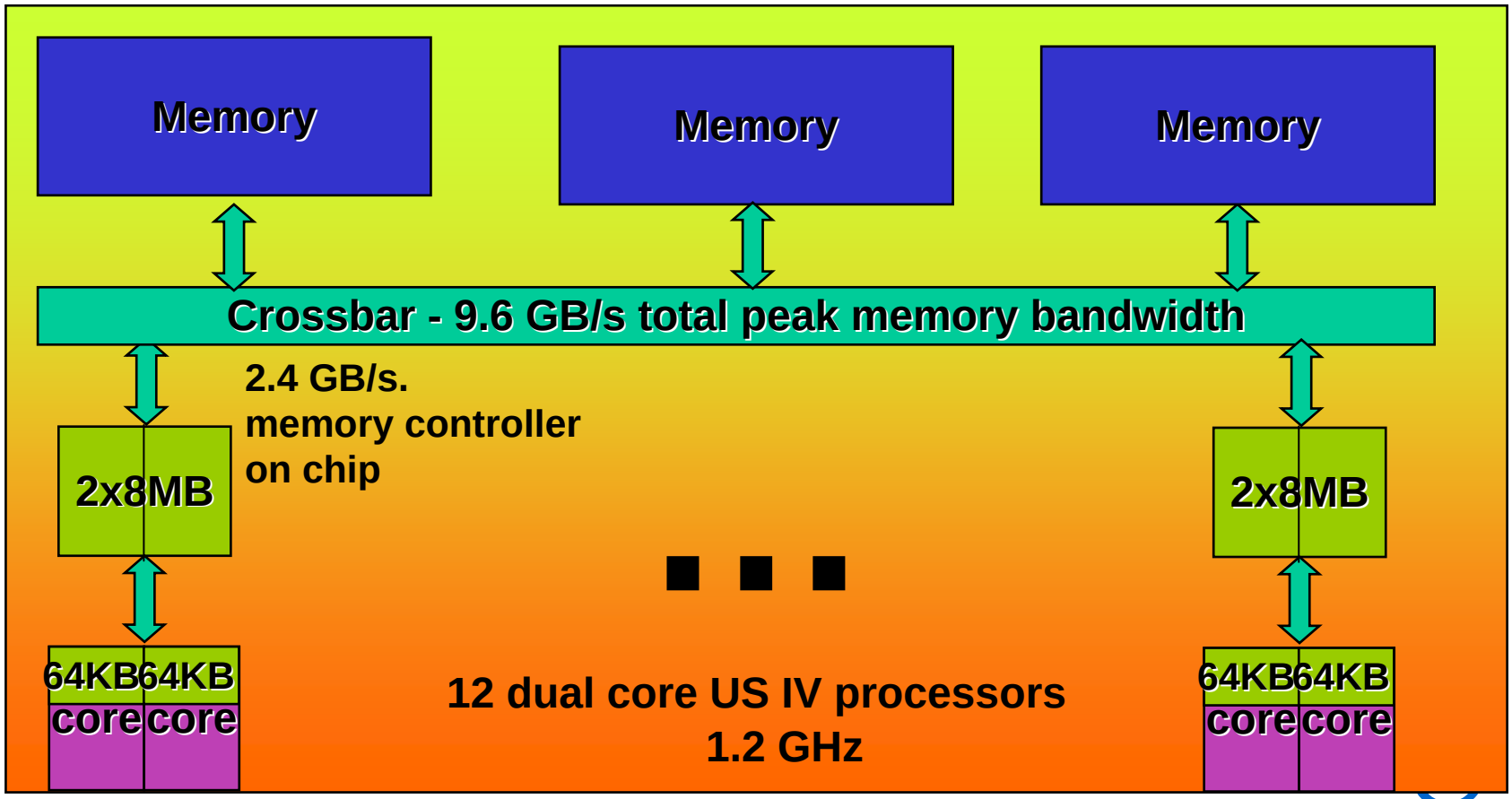
Proc chip	US III	**US IV**	US IV+	**Opt 875**	**US T1**
Cycle [GHz]	0.9	**1.2**	1.5	**2.2**	**1.0**
instr [GIPS]	3.6 sparc v9	**9.6 sparc v9**	12.0 sparc v9	**13.2 x86**	**8.0 sparc v9**
Float.point [Gflop/s]	1.8	**4.8**	6.0	**8.8**	**0.1**
technology	150 nm	**130 nm**	90 nm	**90 nm**	**90 nm**
On chip cache	64 KB	**128 KB**	2 MB	**2 MB**	**3 MB**

Overview

- Processor Architecture
- System Architecture
- Memory Performance
- Applications

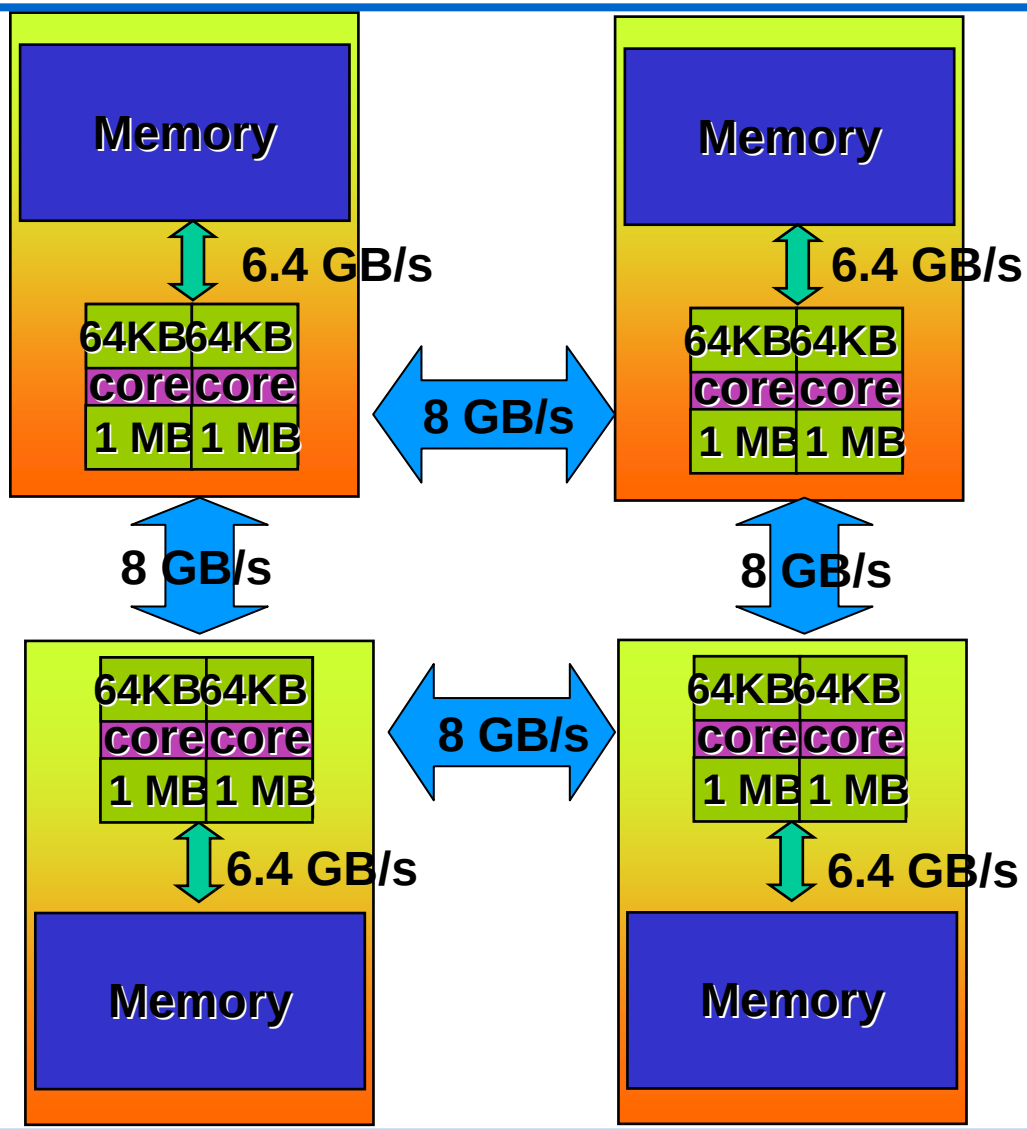
Sun Fire E2900 at Aachen

- *simplistic view*
- *programer's perspective*
- *rather uniform memory access*



UltraSPARC T1 („Niagara"), D. an Mey, et.al.

Sun Fire V40z at Aachen



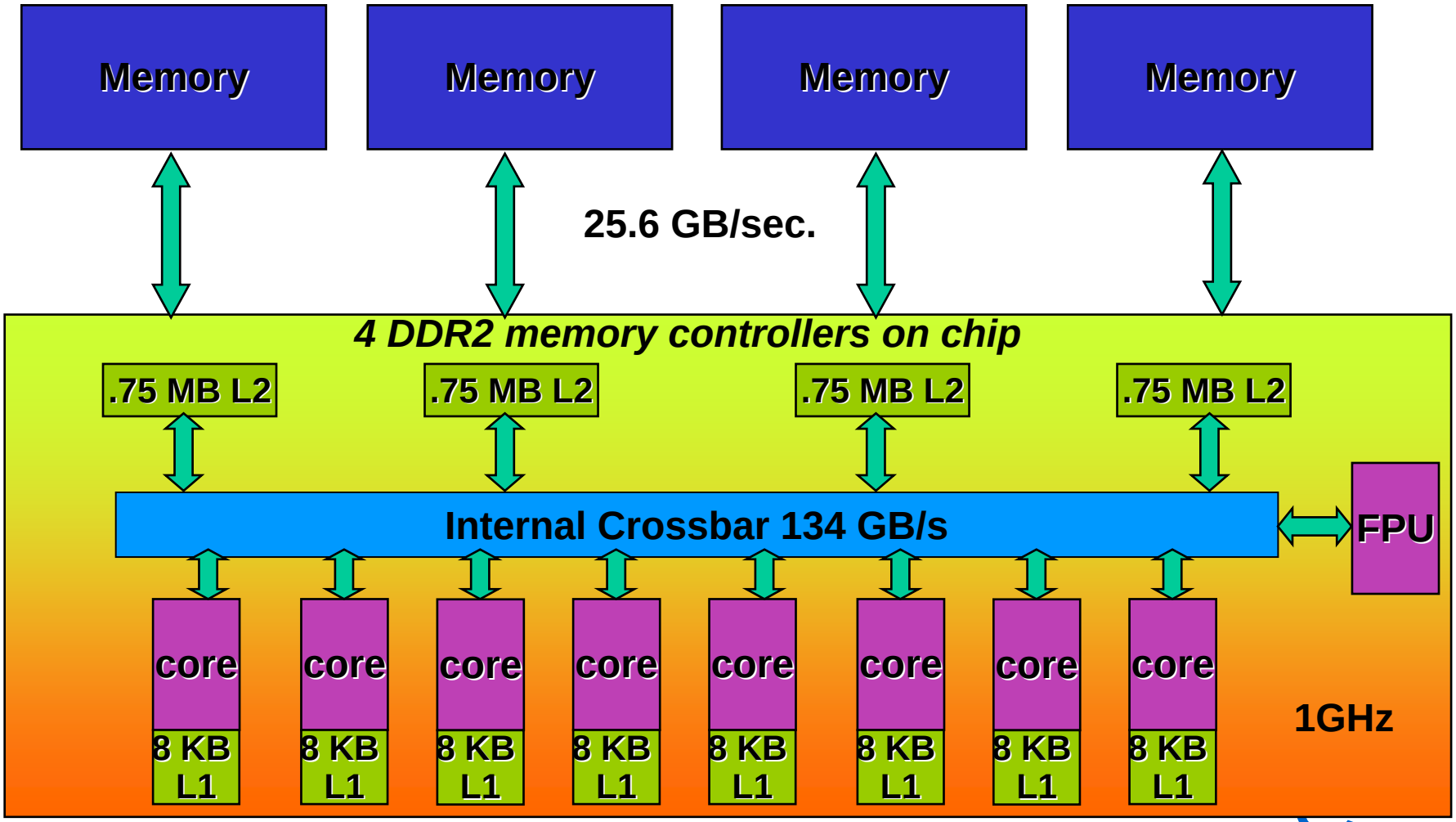
- *simplistic view*
- *programer's perspective*
- *non-uniform memory access*

DDR 400
memory controller
on chip

2.2 GHz

UltraSPARC T1 („Niagara"), D. an Mey, et.al.

Sun Fire T2000 at Aachen



UltraSPARC T1 („Niagara“), D. an Mey, et.al.

Sun Fire T2000 versus Sun Fire E2900

2 U

12 U

3 U

List prices Sun online shop
Germany, March 7, 2006

Sun Fire T2000 12,300 €
1 US T1, 1 GHz, 8GB

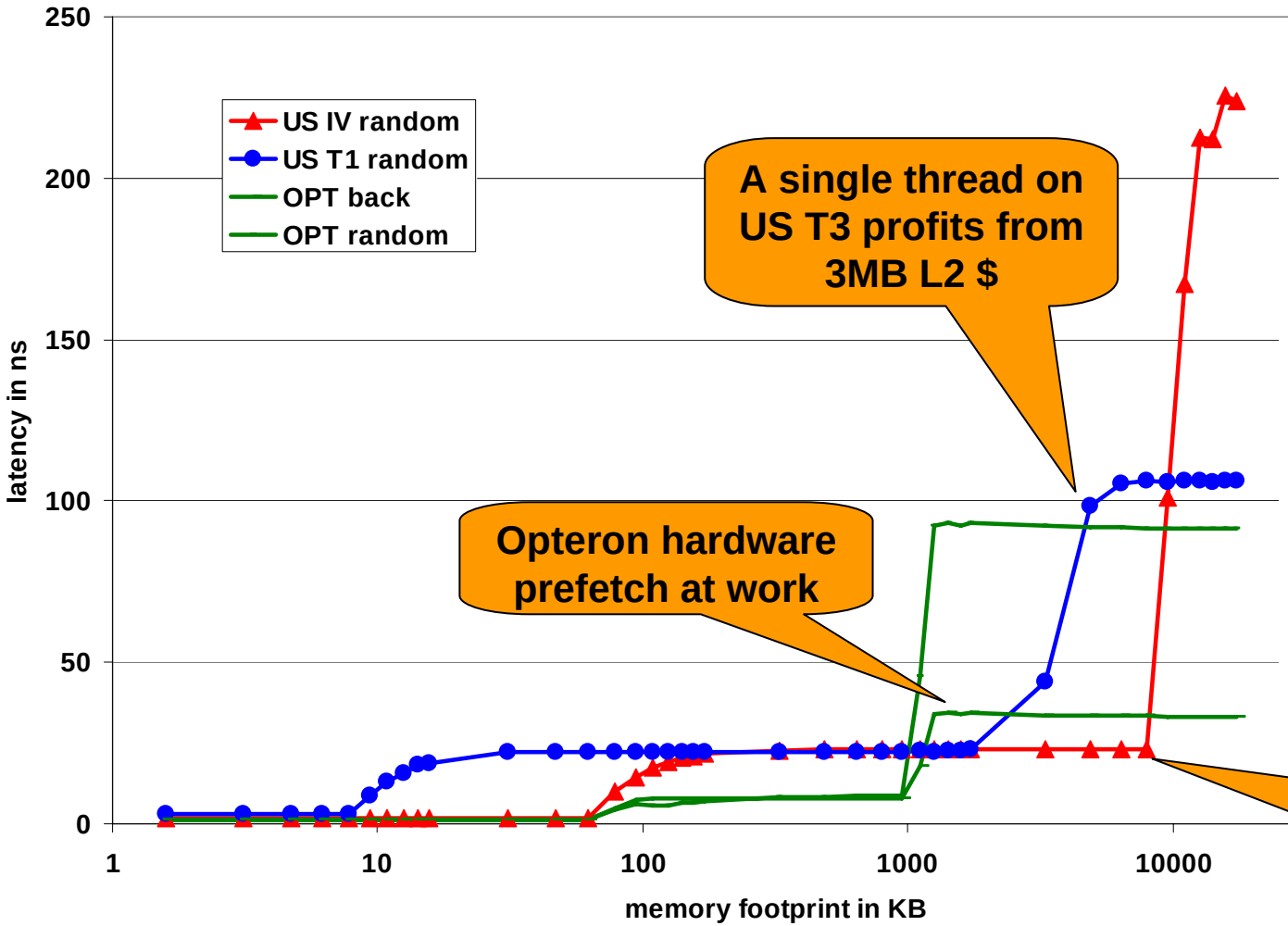
Sun Fire V40z 21,100€
4 Opt 875 dual core, 2.2GHz,16 GB

Sun Fire E2900 223,800 €
12 US IV,1.2GHz, 48 GB

Comparing Boxes …

Server	Sun Fire E 2900	Sun Fire V40z	Sun Fire T2000
Proc chip	US IV 1.2 GHz	Opt 875 2.2 GHz	US T1 1.0 GHz
# Proc chips	12	4	1
# cores	24	8	8
# threads	24	8	32
Instructions [GIPS]	12x9.6=115.2 sparc v9	4x13.2=52.8 x86	8.0 sparc v9
Float.point [Gflop/s]	12x4.8=57.6	4x8.8=35.2	0.1
Total cache	24x64 KB=1.5 MB on chip 24*8=192 MB off chip	8x1 MB on chip	3 MB on chip
Main Memory	48 GB	16 GB	8 GB

Memory Performance - Read Latency (serial)

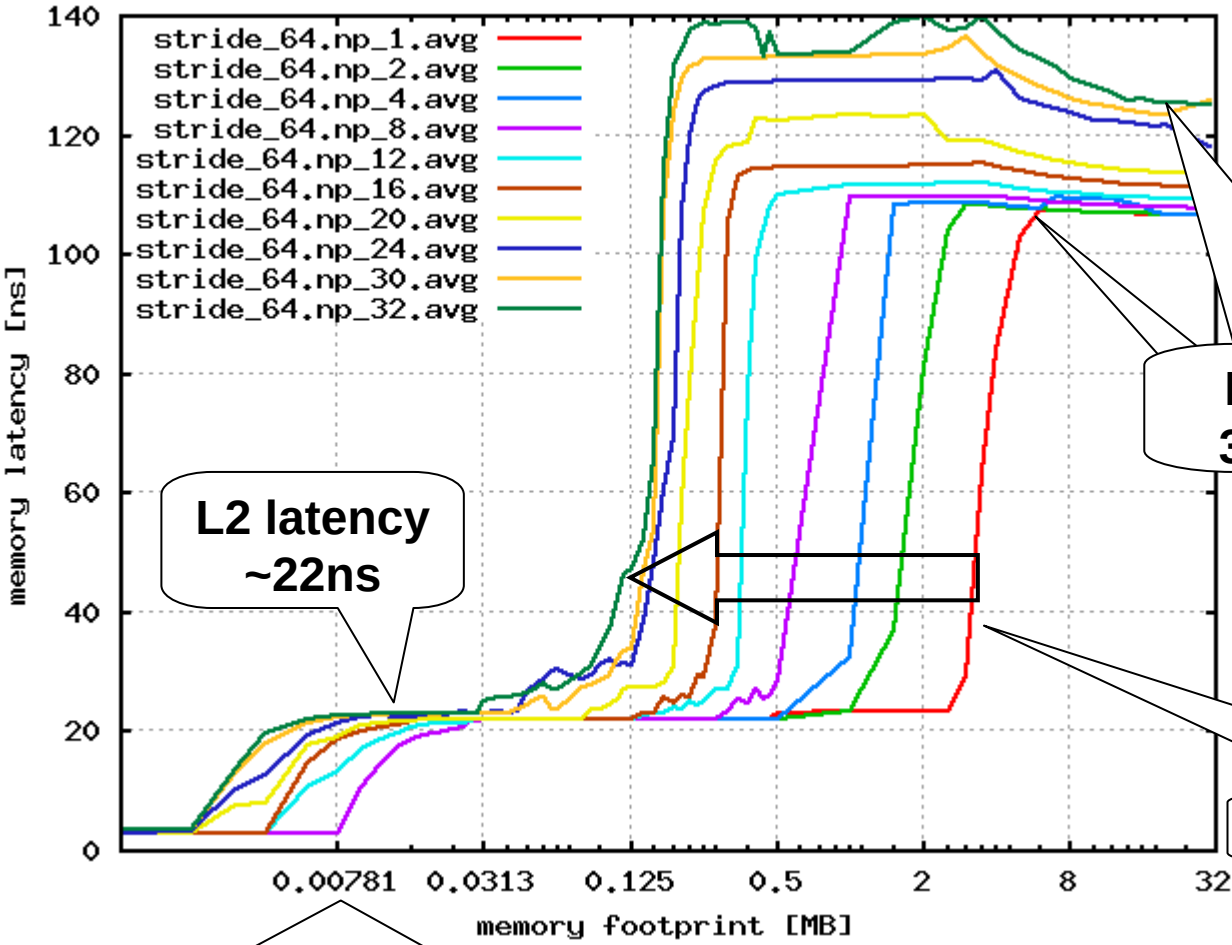


Pointer chasing
`p=(char**)*p;`

The Opteron profits from hardware prefetches when consecutive cache lines are loaded.
The US IV and US T1 perform the same for forward, backward and random loads within the given footprint.

Kernel provided by Partha Tirumalai, Sun

Memory Performance – Read Latency (MPI)
UltraSPARC T1



Pointer chasing
p = (char) *p;**

Stride = 64 Bytes
= 1 L2 cache line

Memory latency, 32 procs ~132 ns

L2 latency ~22ns

Memory latency, 1 proc ~107 ns

1 shared L2 cache 3 MB

8 KB L1 cache for each core

LMbench lat_mem_rd kernel parallelized with MPI

Memory Performance - Read Latency (MPI)

#MPI procs	SF E2900 latency [ns]	SF V40z latency [ns]	SF T2000 latency [ns]
1	232	35	107
2	232	36	107
4	249	36	108
8	250	55	109
16	262		113
24	314		
32			144

The Opteron profits from hardware prefetch when consecutive cache lines are loaded (backward or forward)
~90 ns otherwise

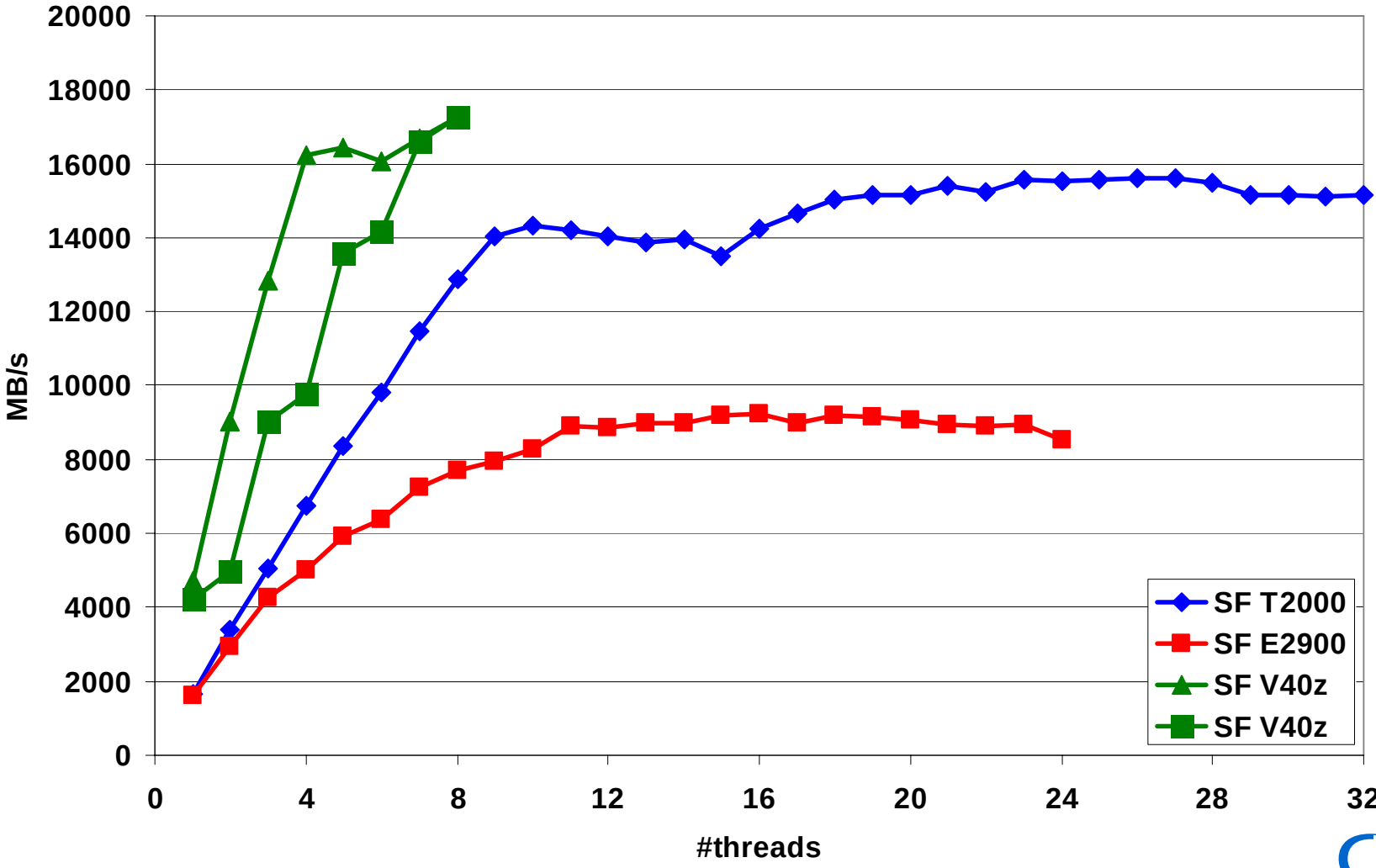
LMbench lat_mem_rd kernel parallelized with MPI

Memory Performance - Read Bandwidth

```
for (j = 0; j < count; j++)
    {
       for (i = 0; i < mymem; i+=8)
          {
             PREFETCH_ONCE(&heap[i+64]);
             r0 += heap[i];
          }
    }
```

Method
- parallelization using **pthreads**
- we read data with a stride of 64 bytes to load 64 byte L2 $ lines from memory
- we add the data to prevent unwanted compiler optimization
- explicite processor **binding**
- default **first touch memory placement**
- **manual prefetching** to load cache lines ahead:
  Niagara: Heap is long long int, sparc_prefetch_read_once() prefetches data into L2 $
  UltraSPARC IV: Heap is double, sparc_prefetch_read_once() prefetches into prefetch $ only.
  Opteron: Heap is double, inline assembler instruction prefetchenta() prefetches directly into L1 $

Memory Performance - Read Bandwidth

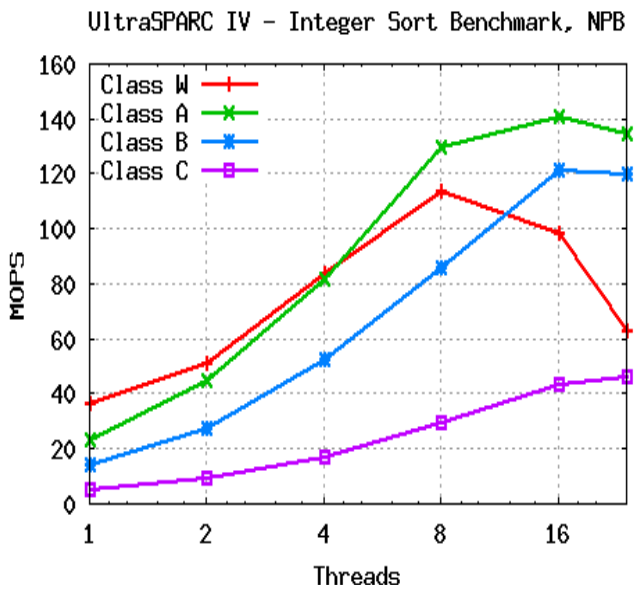


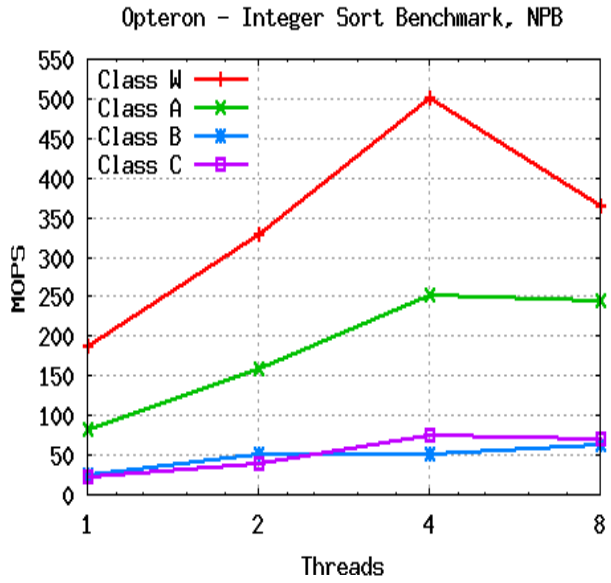
UltraSPARC T1 („Niagara“), D. an Mey, et.al.

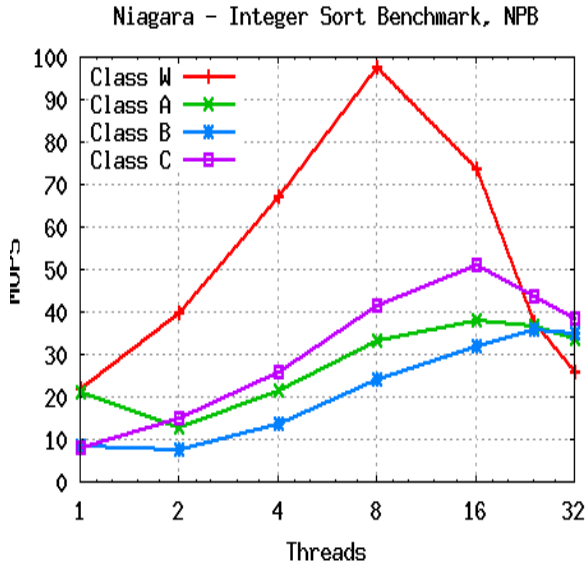
Overview

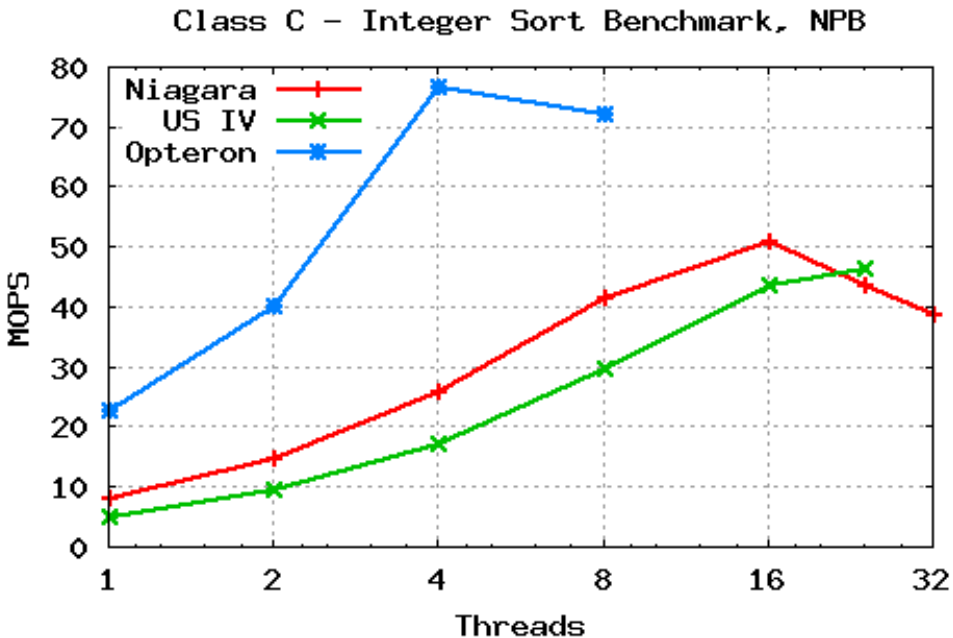
- **Processor Architecture**
- **System Architecture**
- **Memory Performance**
- **Applications**

NAS Parallel Benchmark NPB 3.2 OpenMP - Integer Sort
Sun Fire T2000 versus Sun Fire E2900









- *Sun Fire E2900 profits from the large caches of the UltraSPARC IV proc for the small cases W, A, and B.*
- *SF T2000 outperforms SF E2900 for the large case C.*

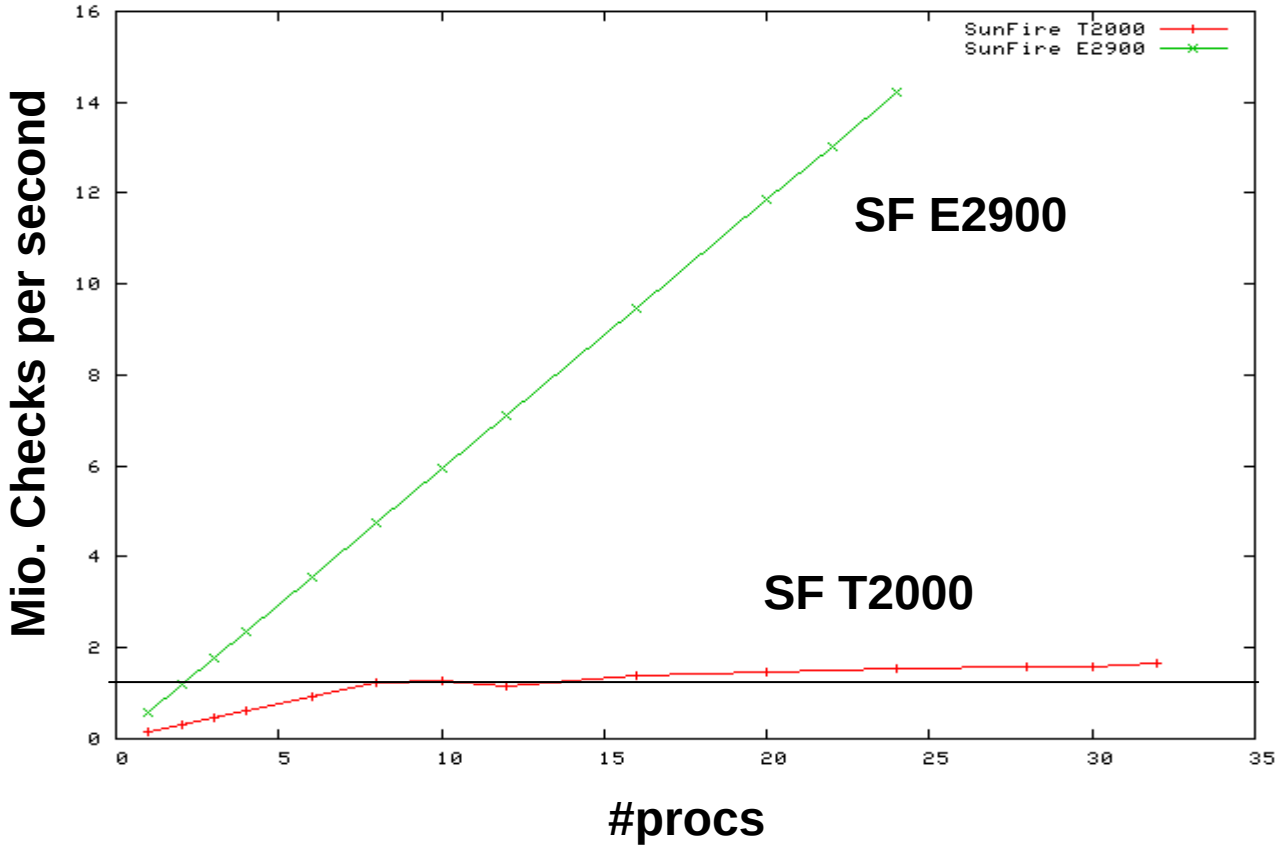
Parallel Partitioning of Graphs with ParMETIS

- ParMETIS is used for partitioning unstructured graphs, in order to minimize the communication of large-scale numerical simulations.
- The test case (mtest) is used to compute a k-way partitioning of a mesh containung elements of different types on p processors.
- The measurements cannot be used to judge scalability, as the partitioning depends on the number of processors used.
- Listed is the total runtime in seconds

#MPI processes	SF E2900	SF T2000	SF V40z
1	1.4	3.1	0.5
2	1.5	3.3	0.5
4	1.0	2.1	0.3
8	0.9	1.6	0.3
16	0.8	1.8	
24	0.8	2.4	
32		3.3	

Password Cracking with "John the Ripper"

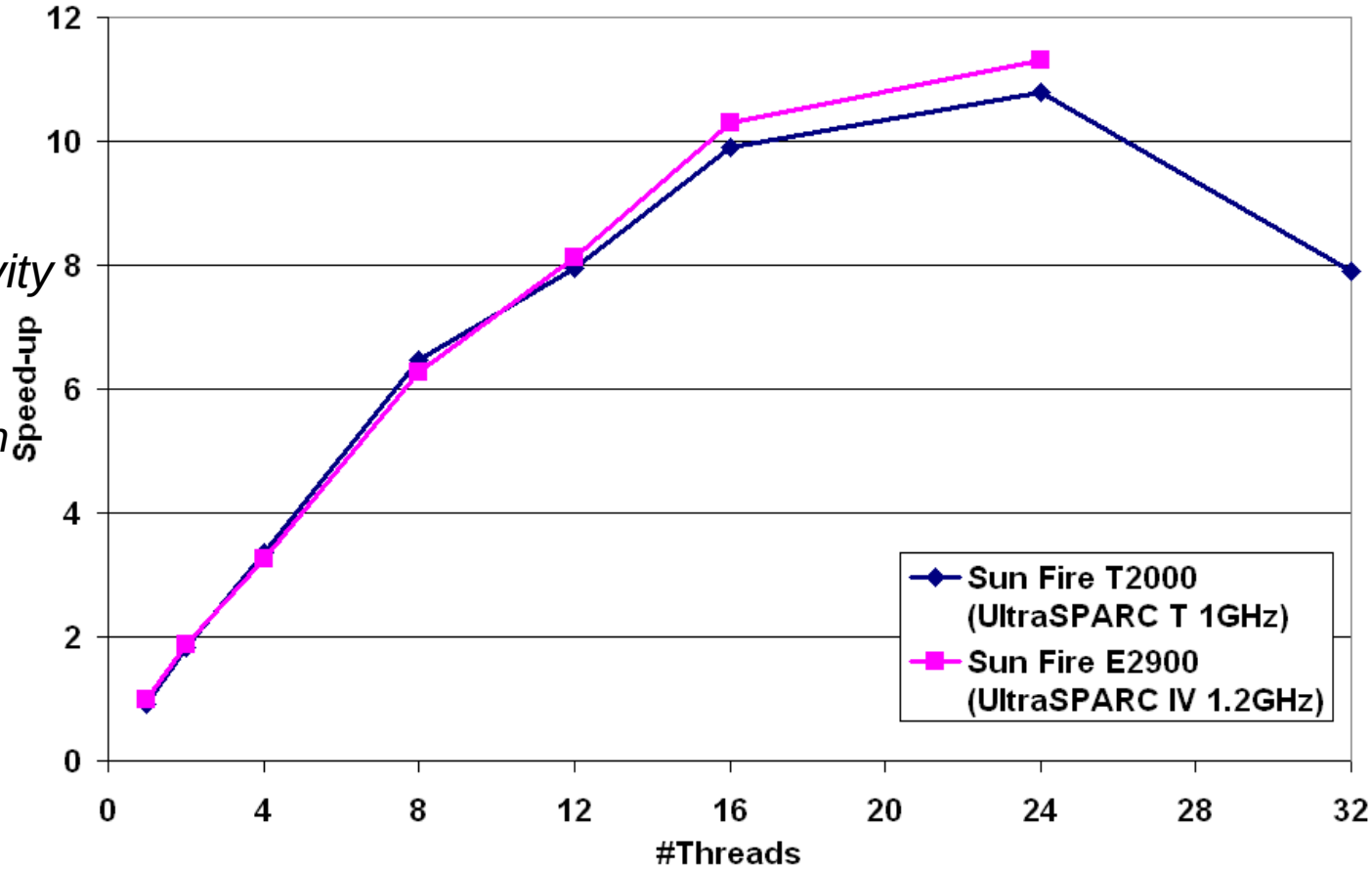
- "**John the Ripper**" is a popular password crack program.
- MPI version linked with mpich2
- The performance measure is checks per second for traditional DES encryption.



UltraSPARC T1 („Niagara"), D. an Mey, et.al.

HPCS Graph Analysis Benchmark

An implementation of the kernels of the Scalable Synthetic Compact Applications (SSCA) benchmark suite, developed under the DARPA High Productivity Computing Systems (HPCS) program.
David Bader, Gorgia Tech

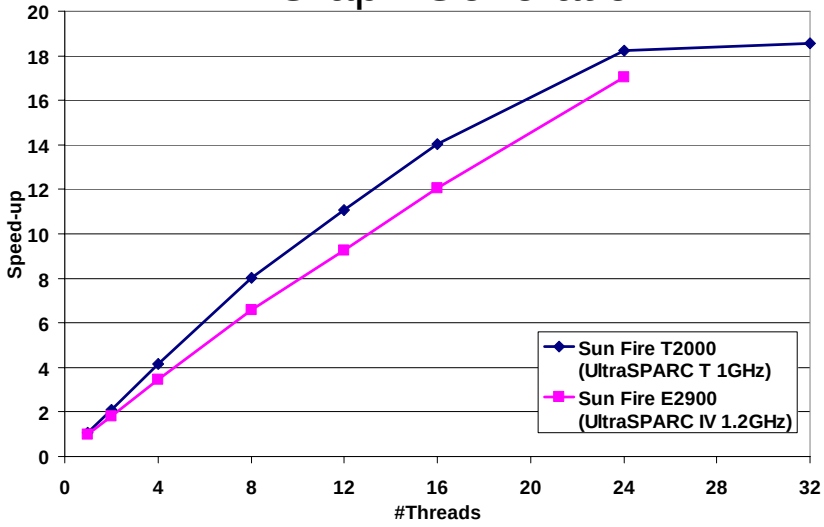


Speedup versus single thread on US IV

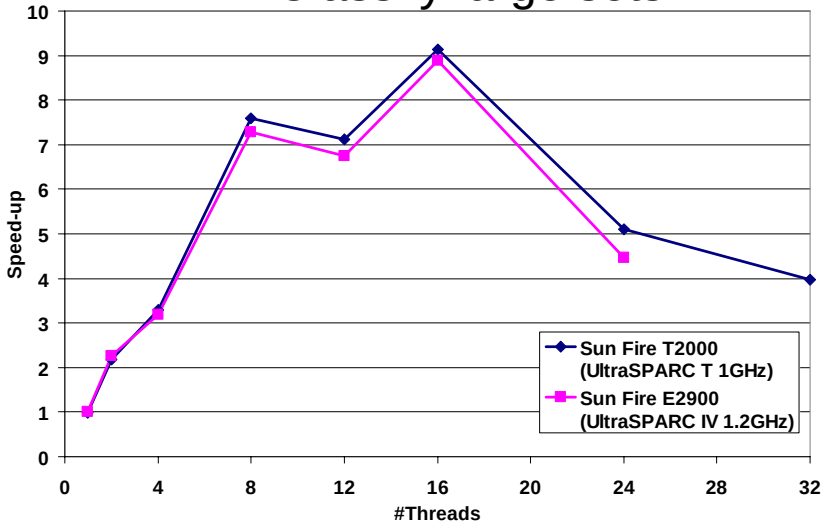
sum of 4 kernels

HPCS Graph Analysis Benchmark

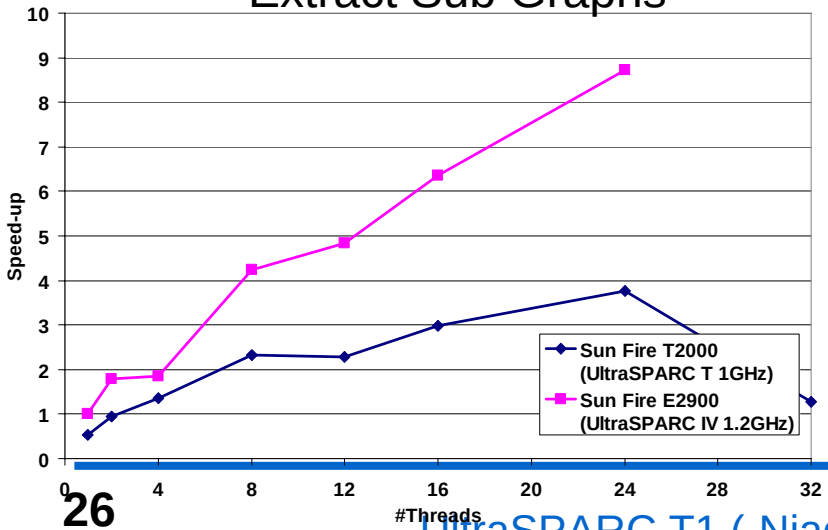
Graph Generation



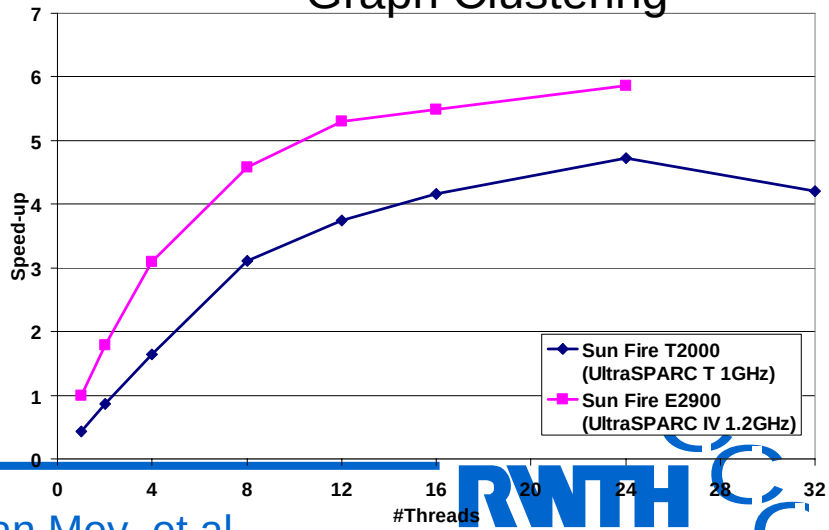
Classify large sets



Extract Sub Graphs



Graph Clustering

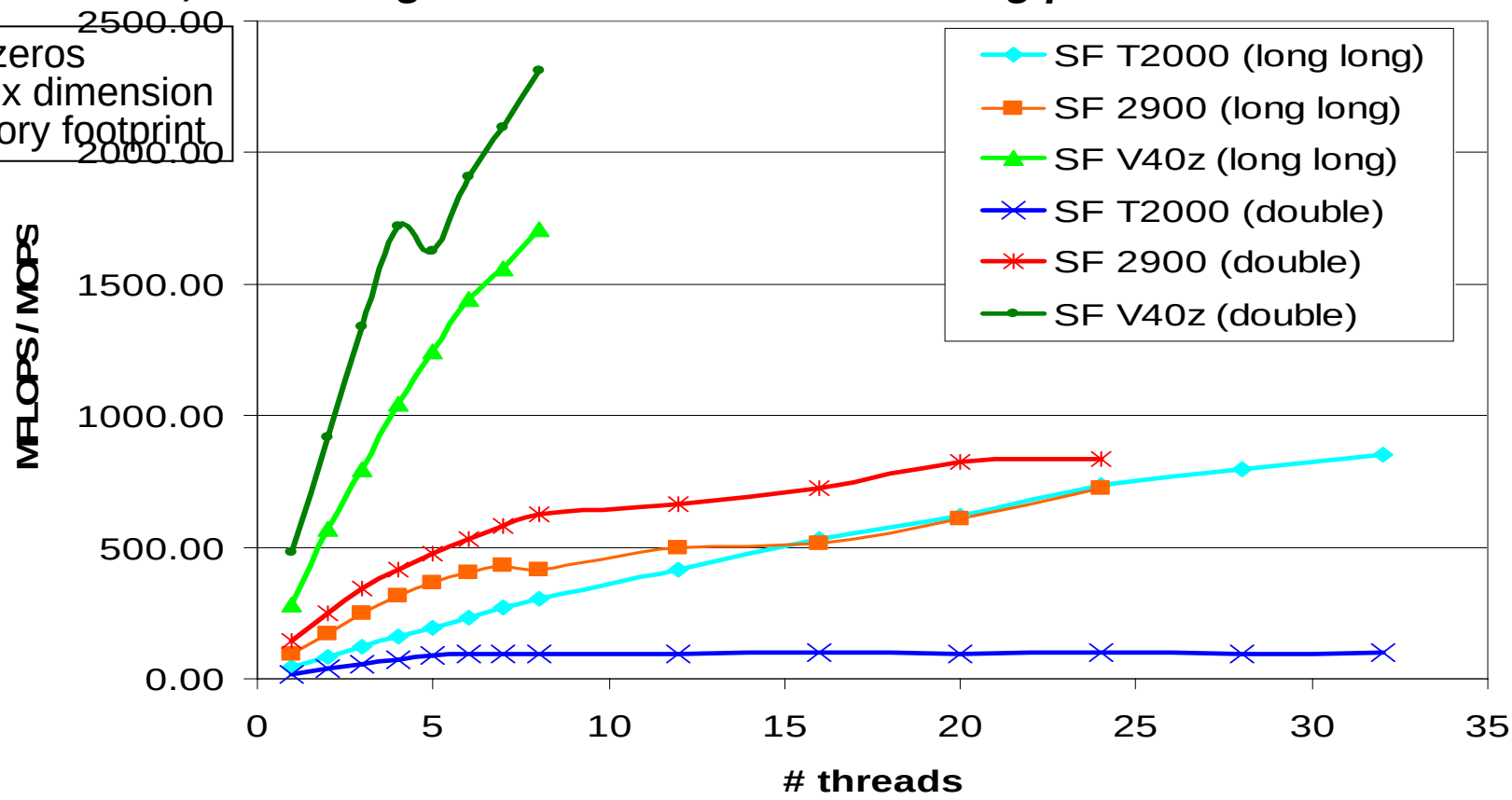


"UltraSPARC T1 („Niagara"), D. an Mey, et.al.

Sparse Matrix Vector Multiplication long int versus double

What, if the Niagara could count with floating point numbers …

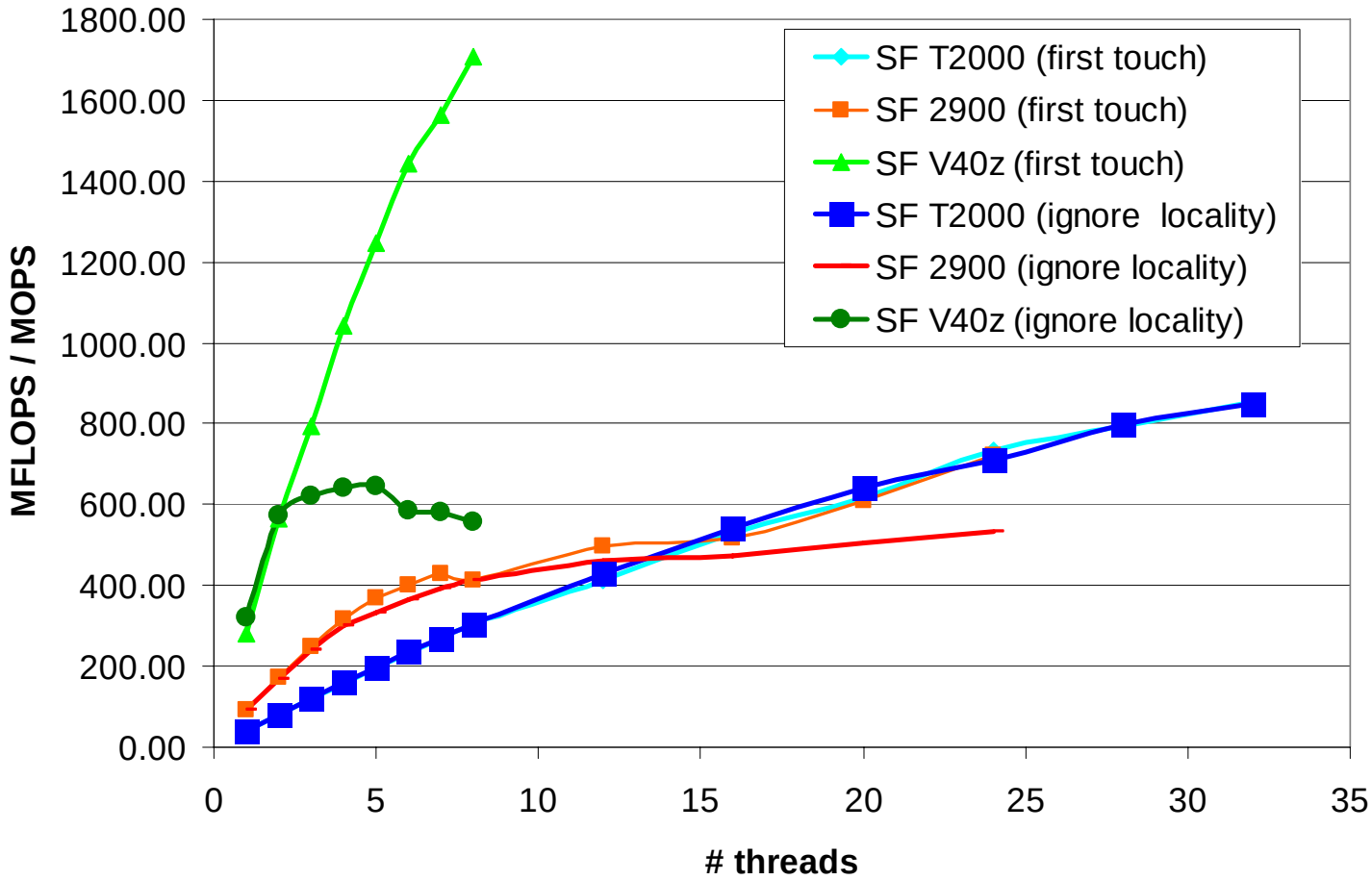
19,6 Mio nonzeros
233,334 matrix dimension
225 MB memory footprint



In many PDE solvers a CG type of linear equation solver is used. The time consuming part is a sparse matrix vector multiplication, which accesses the main memory using index lists.

UltraSPARC T1 („Niagara“), D. an Mey, et.al.

Pitfalls of a ccNUMA Architecture



The SF V40z is very sensitive to bad memory locality.
When ignoring data placement, the performance
of a shared-memory parallel code may heavily suffer.

UltraSPARC T1 („Niagara“), D. an Mey, et.al.

Conclusion

- The SMP box is shrinking.
- In a few years from now, many or all applications will be multi-threaded
- SMP boxes with small footprint will be building blocks of large systems.

- UltraSPARC IV-based SF E2900 one generation behind.
  UltraSPARC IV+-based systems performs 1.6-2.0 times better.

- Opteron delivers highest single thread performance
- ccNUMA architecture of Opteron based servers sensitive to bad memory locality.

- UltraSPARC T1 offers a very high memory bandwidth, low latency and flat memory system at a low price.
- CMT/CMP architecture very promising for large, memory bound numerical problems, as soon as floating point capabilities are expanded.

ALDI informiert

The SMP box is arriving in your living room …