Architecture of Hitachi SR-8000

Uwe Küster

University of Stuttgart
High-Performance Computing-Center Stuttgart (HLRS)
www.hlrs.de

Most of the slides from Hitachi

the problem

- modern computer are data movers
- Floating Point Performance increases with 60% per year (Moore's law)
- memory access is the bottleneck
  - bandwidth is too small
  - latencies are decreasing slowly but increasing in clocks

Latency from processor to various locations

	seconds	clocks	future development
clock rate	1-4 nsec		decreasing
register		1	
L1 cache		1-2	
L2 cache		4-8	
L3 cache		8-20	
memory	30-200 nsec	30-200	slowly decreasing
remote memory		500-700	slowly decreasing
shmem	~3 µsec		
MPI	6-20 µsec		slowly decreasing
disk	~8 msec		
tape	~30 sec		

Behandlung von Latenzzeiten

- hiding of latency is necessary
- different possibilities:
  - ↑ Prefetching and Preloading from own and remote memory
  - ↑ Hardware-Multithreading
  - ↑ decreasing the number of transactions by pipelining
  - ↑ prozessor in memory (IRAM)

prefetch, example

```
do n=1,nmax
 !    prefetch b(n+100), c(n+100)  ! compiler generated
      a(n) = b(n) + c(n)
enddo
```

handling of latencies

compiler can hide latency by prefetching variables

but only in a semantical clear situation

modern processors support prefetching

explicit loops allow prefetching because future addresses can be calculated directly

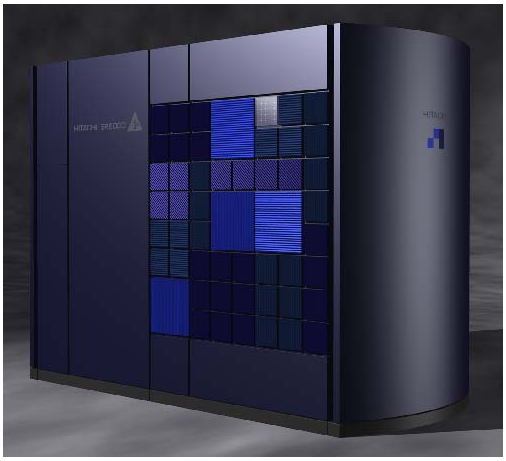
branches are dangerous for prefetching if wrong data are prefetched and vaste the bandwidth:

avoid unnecessary branches

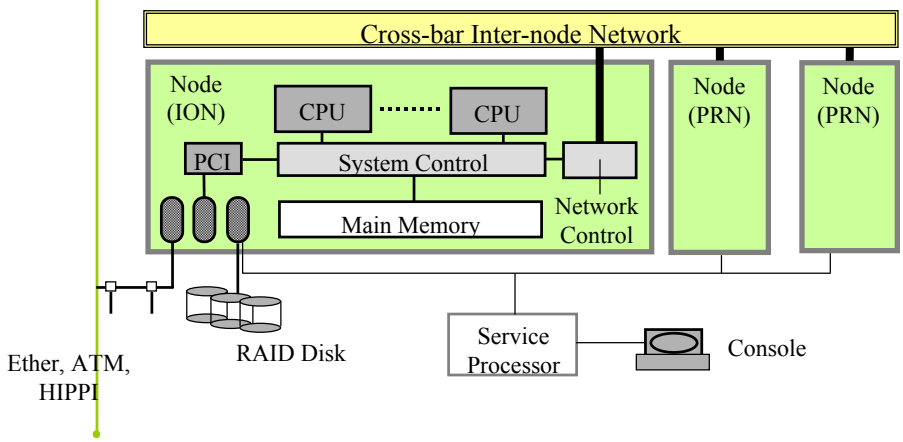
calls break compiler optimization:

avoid frequent calls

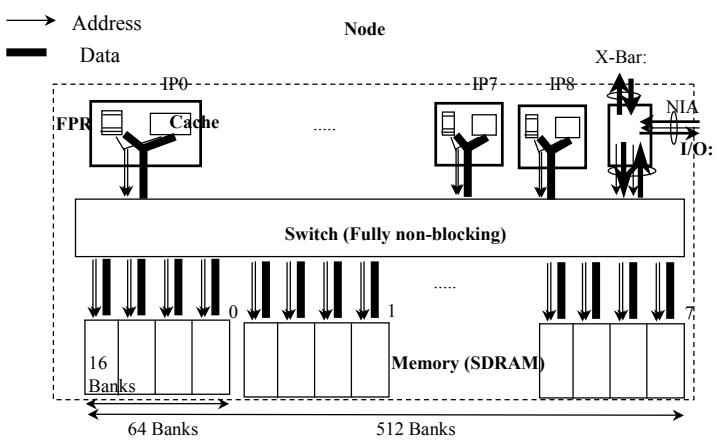
16 processor SR8000



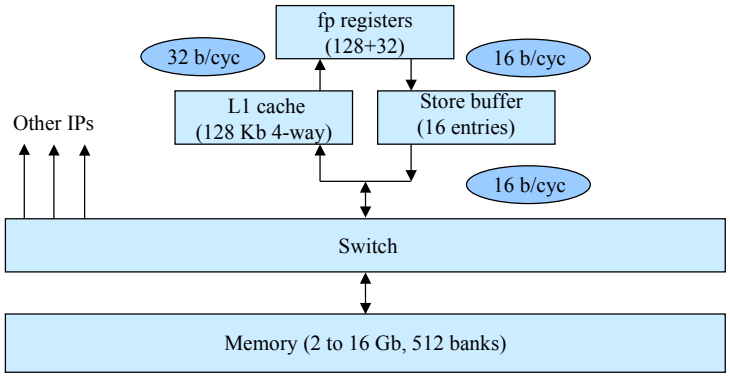
System Architecture SR 8000



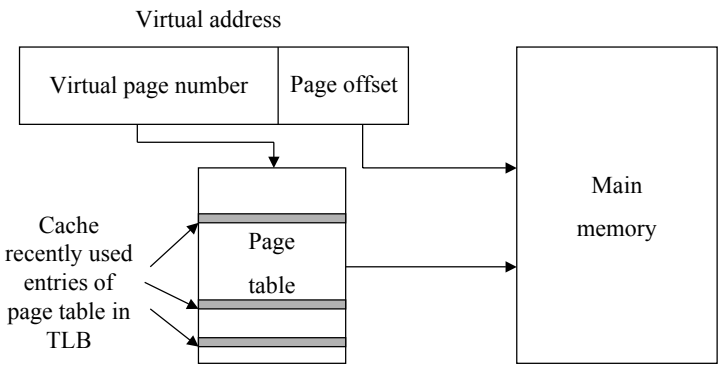
Hitachi SR-8000 node



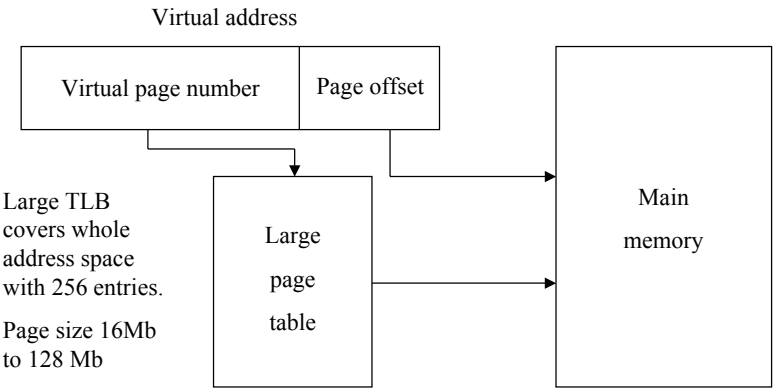
SR-8000: Memory Hierarchy



Address Translation



Large TLB



Large TLB covers whole address space with 256 entries.

Page size 16Mb to 128 Mb

Topology SR-8000

3D Torus

8X8X8 (=512) processors largest size

Parameter debis-SFR Hitachi SR-8000

see Simmendinger: http://thales.rz.st.dlr.de/cgi-bin/fom?file=16

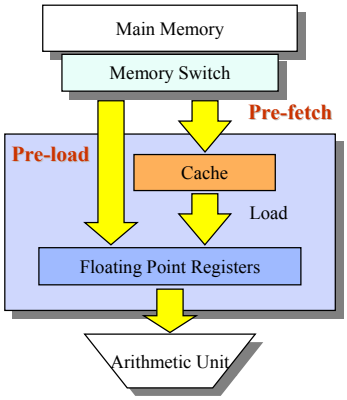
Peak performance	128 Gflop/s
Processors	16x8=128
Frequency	250 MHz
Memory	16x8=128 GB
Disk	440 GB
Inter-Node Bandwidth Hardware	1 GB/s
Inter-Node Bandwidth MPI/RDMA	950 MB/s
Inter-Node Bandwidth MPI	770 MB/s
Inner-Node Bandwidth Memory	32 GB/s/Node
L1 Cache	direct 4-way assoc
L1 Cache size	8x128KB/Node
L1 Cacheline size	128 B
Bandwidth Register - L1 Cache	32 B/cycle
Bandwidth L1 Cache - Memory	16 B/cycle
Bandwidth Register - Memory	8 B/cycle

Hitachi SR 8000:

- nodes with 8+1 processors
- processors have rotating registers (pseudo vector registers)
- 8 processors are cooperating for data parallel code
- or for shared memory parallelism

CPU Architecture

- 16 bytes/cycle memory BW
- 128 Kbyte L1 cache
- Pre-fetch ; 16 open transactions
- pre-load bypassing cache
- 160 FP registers
- 2 FP pipelines
- 4 flops/cycle



Slide Window Registers

Logical	126-7	16 to 31	32 to 125	0 to 15	Base=2
	124-7	16 to 31	32 to 123	0 to 15	Base=4
Physical	Sliding part: 0 to 127			Global part: 128 to 159	

- Registers for all instructions
- Registers for extended instructions only
- Fixed registers: 4, 8, 16, 32 (16 illustrated)
- Fixed + sliding = 128
- see also in IA-64

Instruction Set Extensions

- Load and store with extended registers
- Floating point arithmetic with extended registers
- Slide window control
- Pre-fetch and pre-load
- Thread start-up and finish
- Predicate instructions

Programming Models

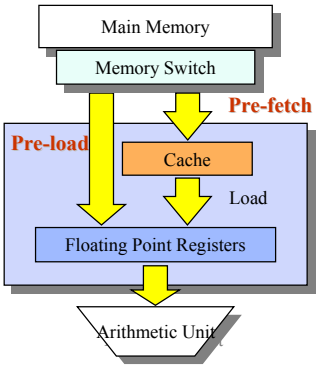
Hardware	Programming Model	Example
Single CPU	Pseudo-vector processing	Vector application
Single Node	Independent processing on each IP	Compilation, parallel make
	Message passing	MPP application
	DO loop distribution with COMPAS	Vector application
	Parallel processing of independent blocks of code	
Multiple Nodes	Message passing	MPP application
	COMPAS and message passing	Vector parallel application

SR8000 Parallel Programming Models

Pseudo-Vector Processing:	PVP
COoperative MicroProcessors in single Address Space:	COMPAS
Message Passing:	MPI

Pre-fetch and Pre-load

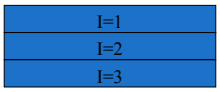
- Pre-fetch: load cache line from memory to cache
- Pre-load: load one word from memory to register
- 16 streams



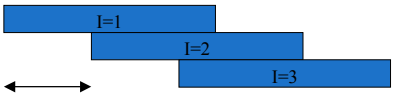
Software Pipelining

No SWPL

Infinite resource

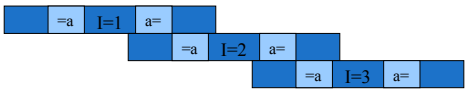


Finite resource



Initiation interval

Recurrence

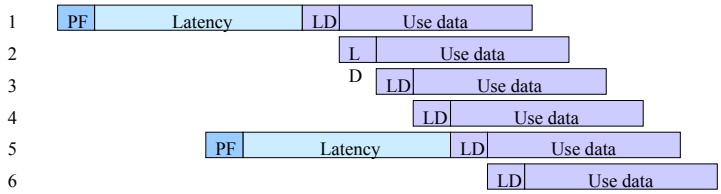


Resources:

registers, f.p. units, instruction issue, memory bandwidth etc

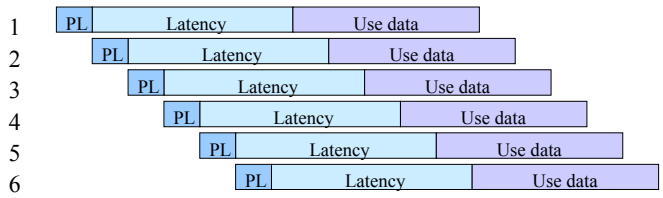
Pre-fetch

Iteration



- Pre-fetch 128 bytes to cache
- Follow by LD to register

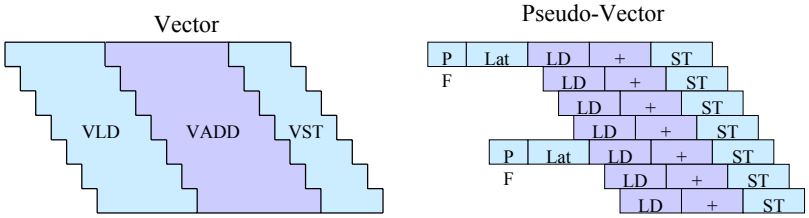
Pre-load



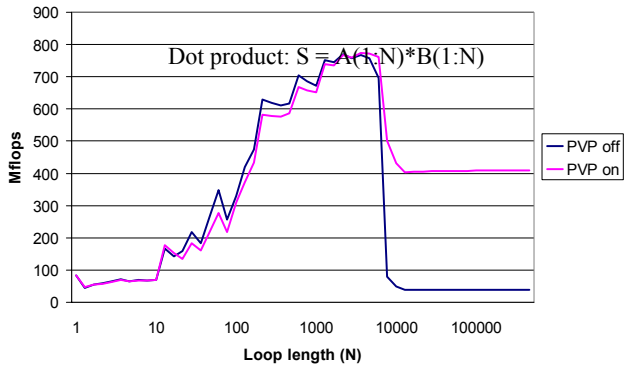
- Pre-load 8 bytes to register
- LD not required

Pseudo-vector Processing

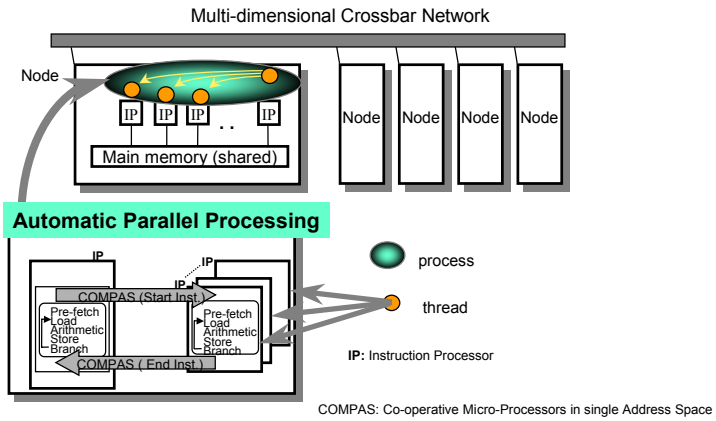
A(:) = A(:) + N



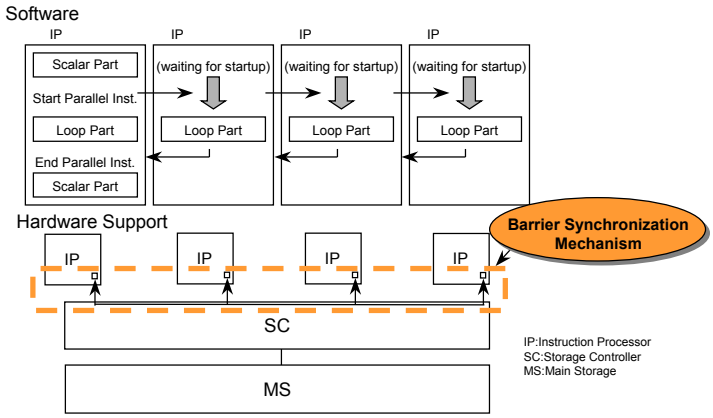
Effect of PVP



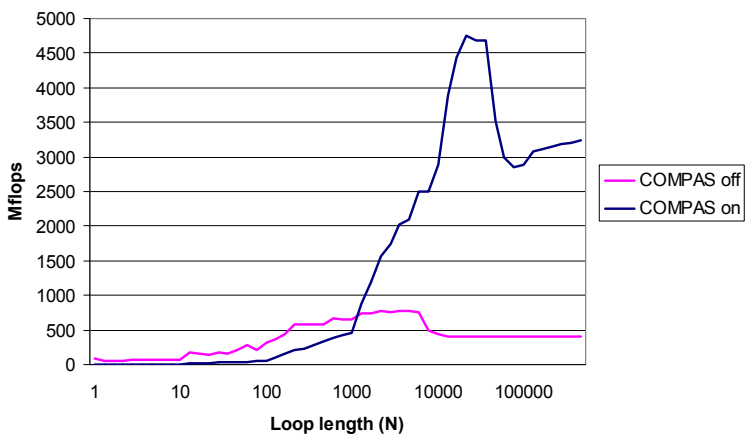
COMPAS



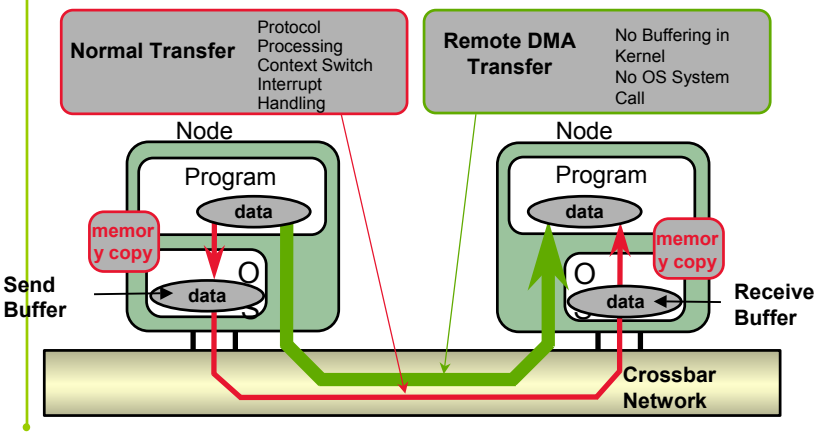
Hardware Support



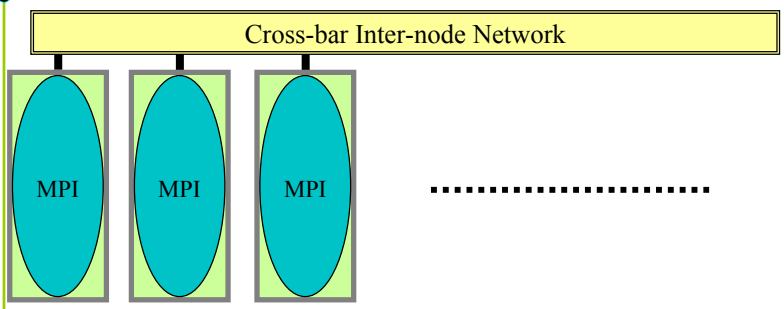
Effect of COMPAS: Dot product: S = A(1:N)*B(1:N)



Remote DMA

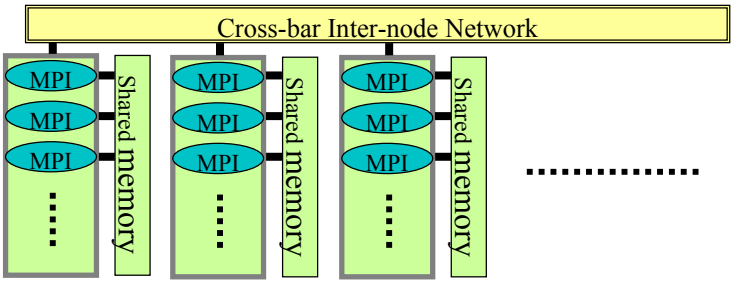


Inter-node MPI



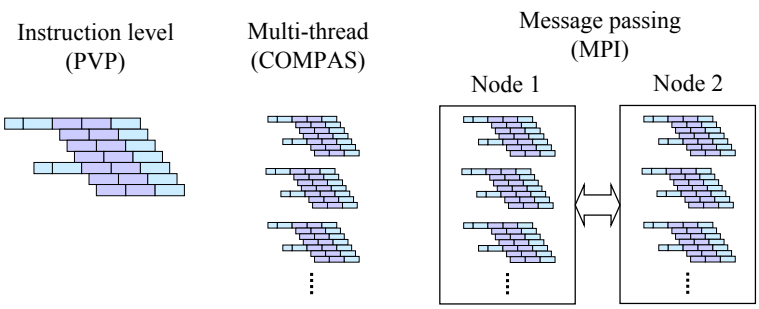
One MPI process per node; RDMA transfer possible

Intra-node MPI



One MPI process per IP; RDMA transfer not possible

SR8000: Parallelism on all levels



Key Features of SR8000

- High performance RISC CPU with PVP
- High performance node with COMPAS
- High sustained memory bandwidth
- High scalability with fast network
- Low energy and space requirements