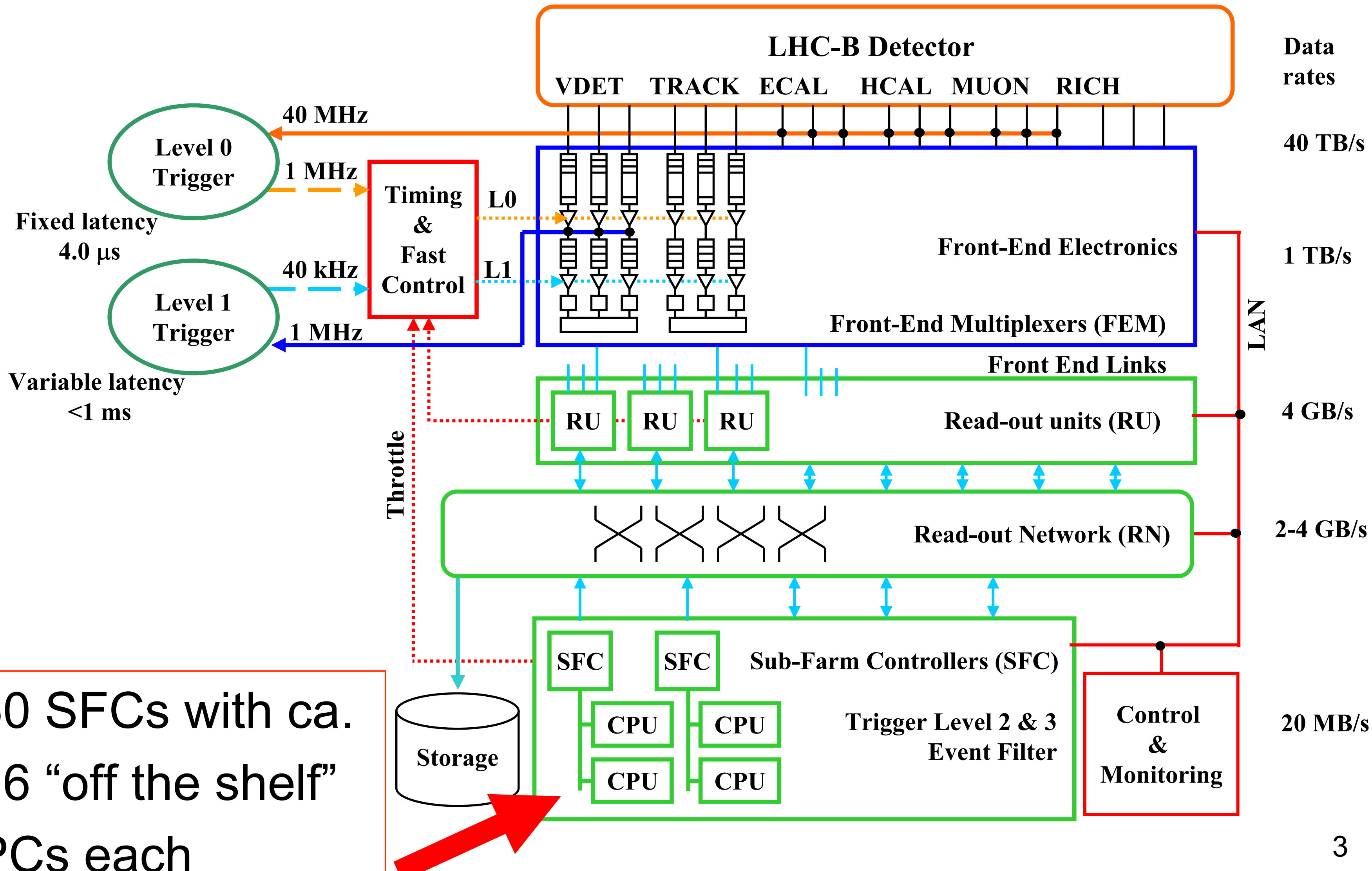
Network Performance Optimisation and Load Balancing

Wulf Thannhaeuser

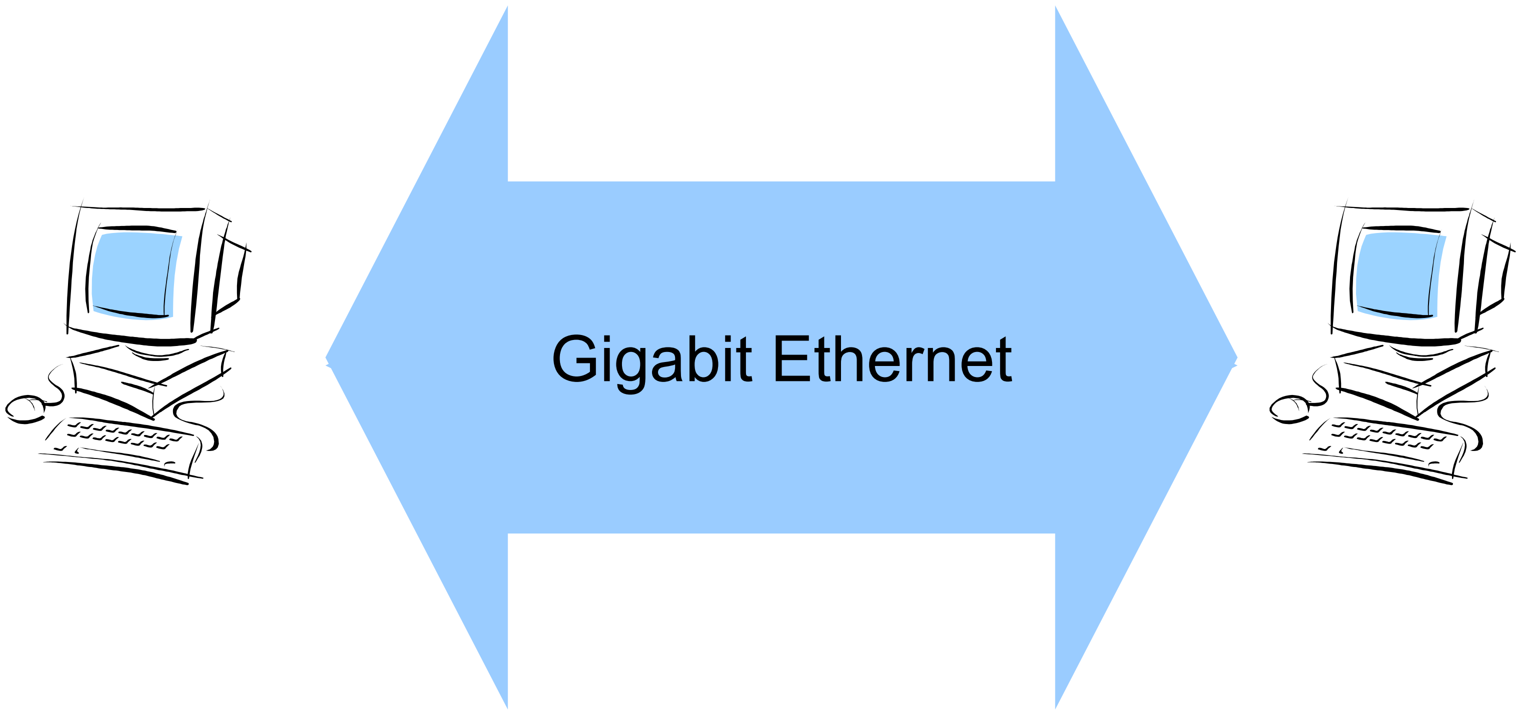
Network Performance Optimisation

Network Optimisation: Where?



60 SFCs with ca. 16 “off the shelf” PCs each

Network Optimisation: Why?

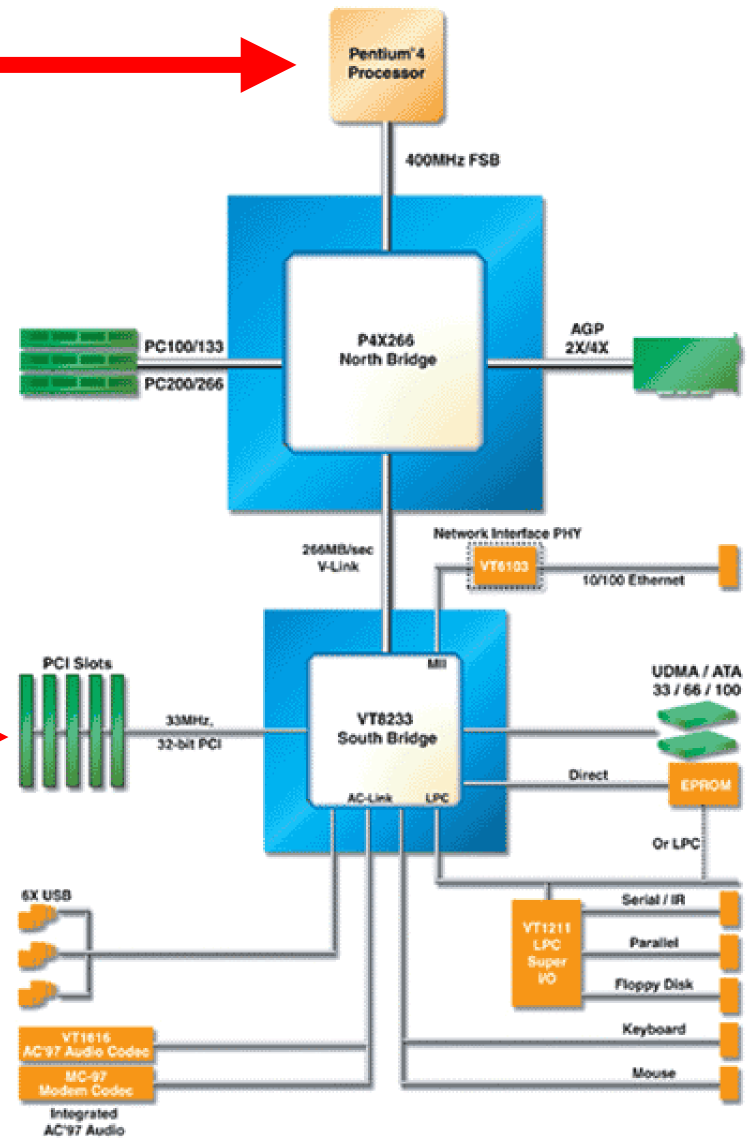


Ethernet Speed:	10 Mb/s
Fast Ethernet Speed:	100 Mb/s
Gigabit Ethernet Speed:	1000 Mb/s
(considering full-duplex:	2000 Mb/s)

Network Optimisation: Why?

An “average” CPU might not be able to process such a huge amount of data packets per second:
-TCP/IP Overhead
-Context Switching
-Packet Checksums

An “average” PCI Bus is 33 MHz, 32-bit wide.
Theory: 1056 Mbit/s
Actually: ca. 850 Mbit/s
(PCI overhead, burstsize)



Network Optimisation: How?

An "average" CPU might not be able to process such a huge amount of data packets per second:
-TCP/IP Overhead
-Context Switching
-Packet Checksums

Reduce per packet Overhead:
Replace TCP with UDP

An "average" PCI Bus is 33 MHz, 32-bit wide.
Theory: 1056 Mbit/s
Actually: ca. 850 Mbit/s
(PCI overhead, burstsize)

TCP / UDP Comparison

- TCP (Transfer Control Protocol):
  - connection-oriented protocol
  - full-duplex
  - messages received in order, no loss or duplication

  $\Rightarrow$ *reliable but with overheads*

- UDP (User Datagram Protocol):
  - messages called “datagrams”
  - messages may be lost or duplicated
  - messages may be received out of order

  $\Rightarrow$ *unreliable but potentially faster*

Network Optimisation: How?

An "average" CPU might not be able to process such a huge amount of data packets per second:
- -TCP/IP Overhead
- -Context Switching
- -Packet Checksums

An "average" PCI Bus is 33 MHz, 32-bit wide.
Theory: 1056 Mbit/s
Actually: ca. 850 Mbit/s
(PCI overhead, burstsize)

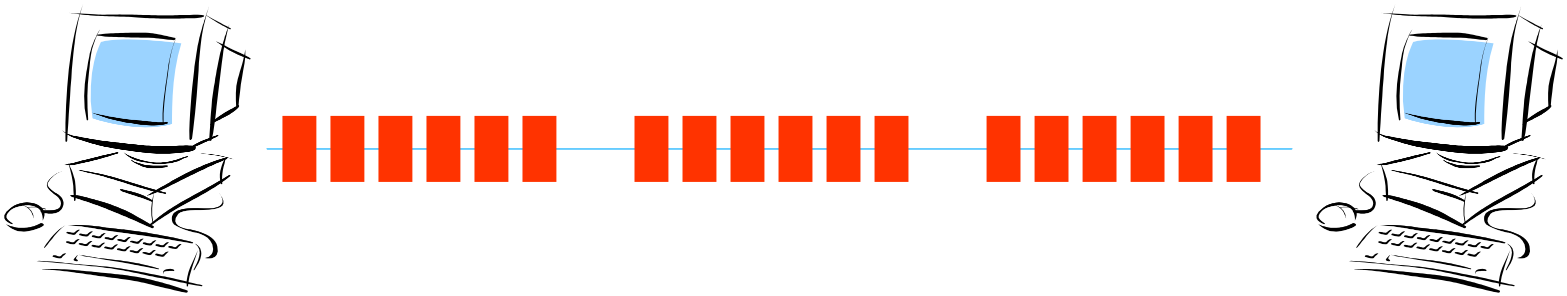
Reduce per packet Overhead:
Replace TCP with UDP

Reduce number of packets:
Jumbo Frames

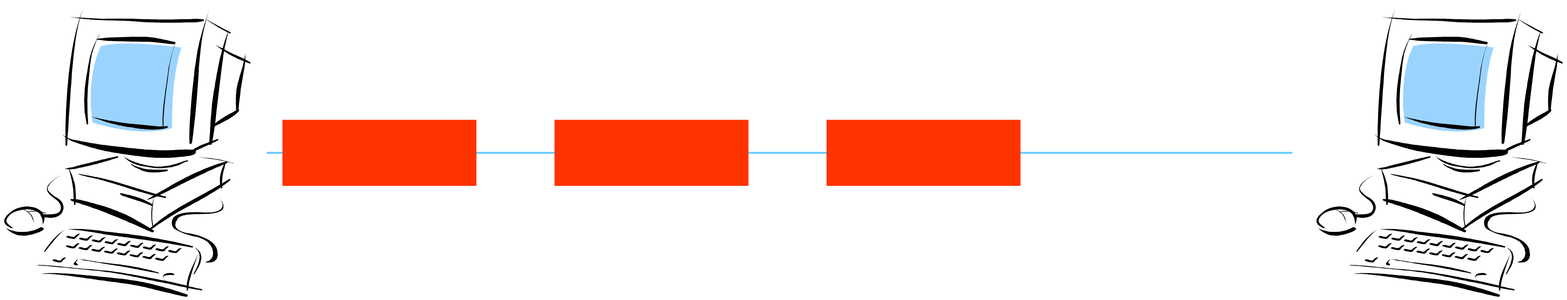
Jumbo Frames

Normal Ethernet

Maximum Transmission Unit (MTU): 1500 bytes



Ethernet with Jumbo Frames MTU: 9000 bytes

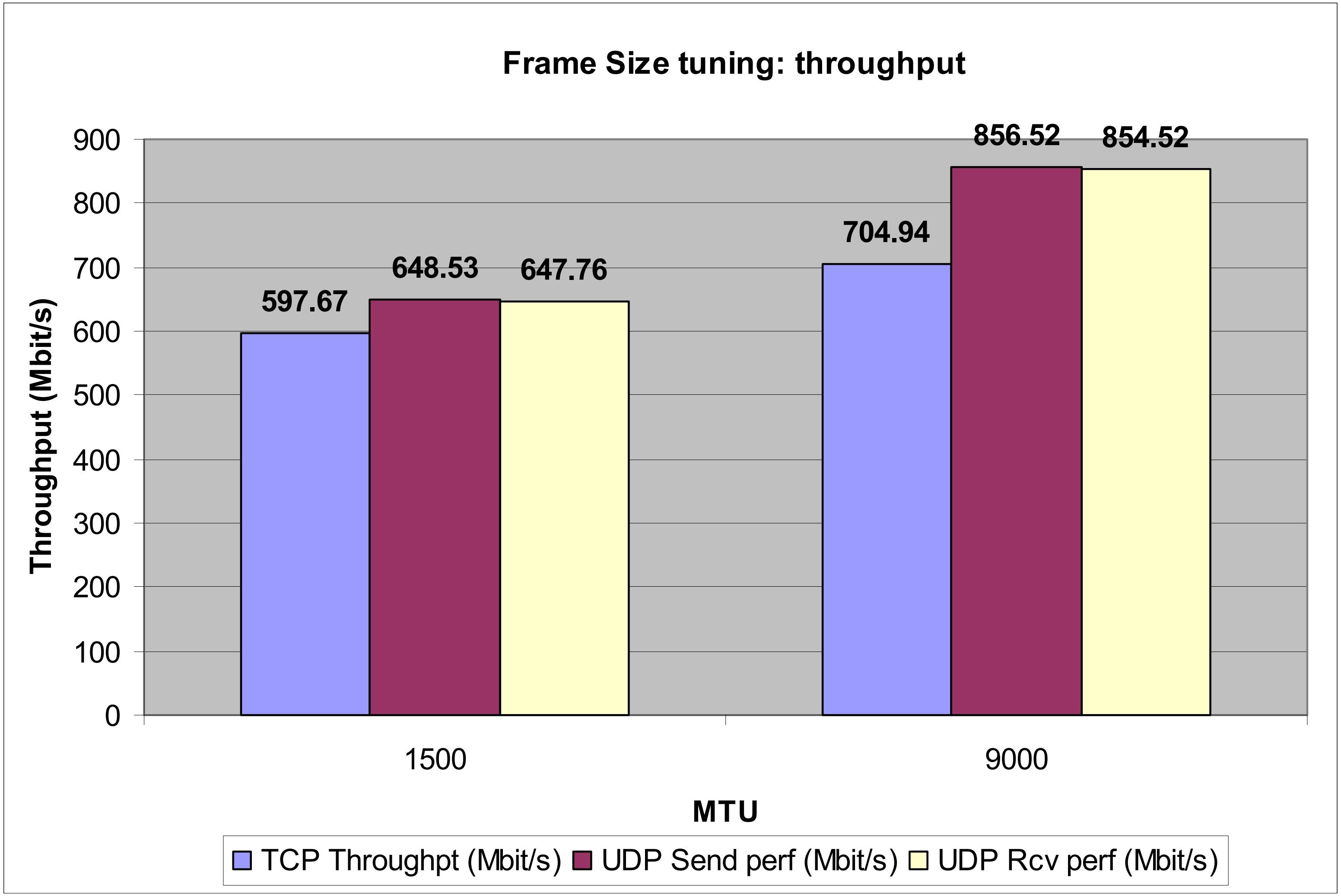


Test set-up

- Netperf is a benchmark for measuring network performance
- The systems tested were 800 and 1800 MHz Pentium PCs using (optical as well as copper) Gbit Ethernet NICs.
- The network set-up was always a simple point-to-point connection with a crossed twisted pair or optical cable.
- Results were not always symmetric:

  With two PCs of different performance, the benchmark results were usually better if data was sent from the slow PC to the fast PC, i.e. the receiving process is more expensive.

Results with the optimisations so far



Network Optimisation: How?

An "average" CPU might not be able to process such a huge amount of data packets per second:
-TCP/IP Overhead
-Context Switching
-Packet Checksums

An "average" PCI Bus is 33 MHz, 32-bit wide.
Theory: 1056 Mbit/s
Actually: ca. 850 Mbit/s
(PCI overhead, burstsize)

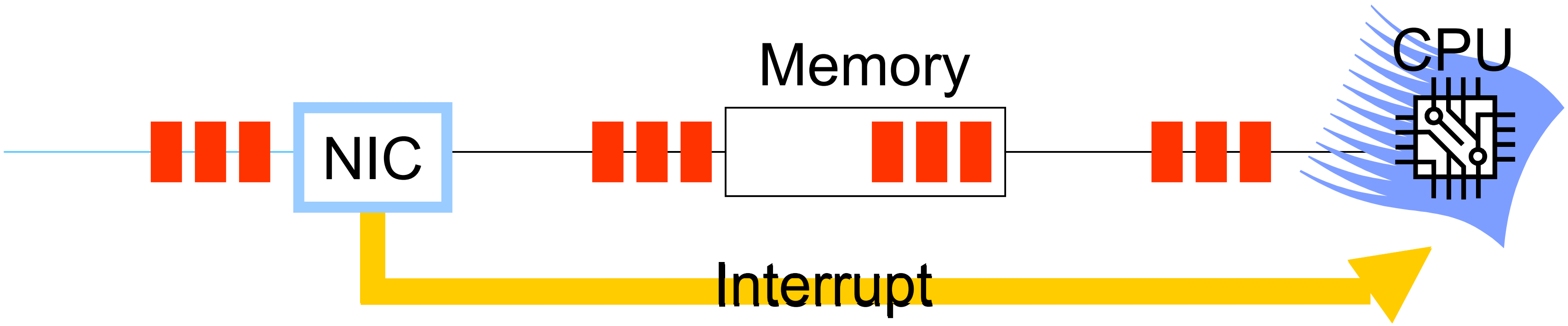
Reduce per packet Overhead:
Replace TCP with UDP

Reduce number of packets:
Jumbo Frames

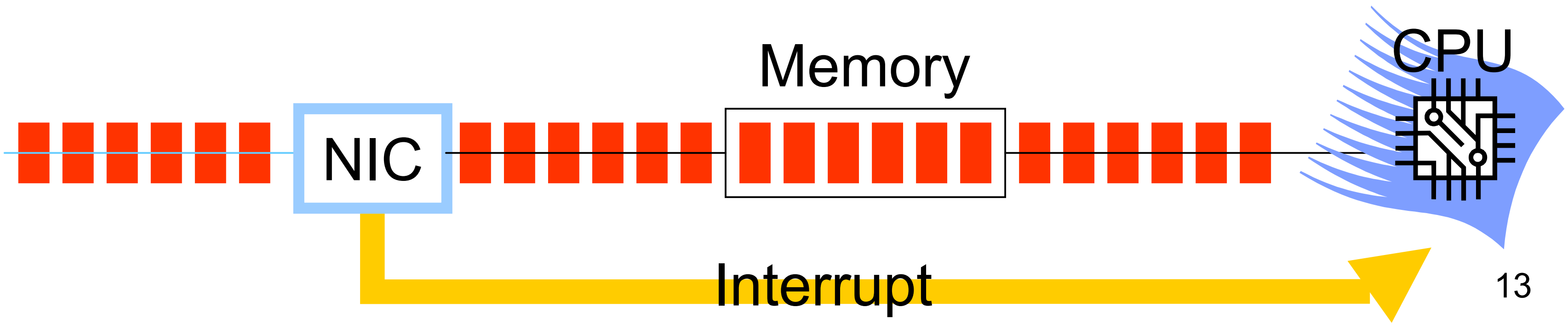
Reduce context switches:
Interrupt coalescence

Interrupt Coalescence

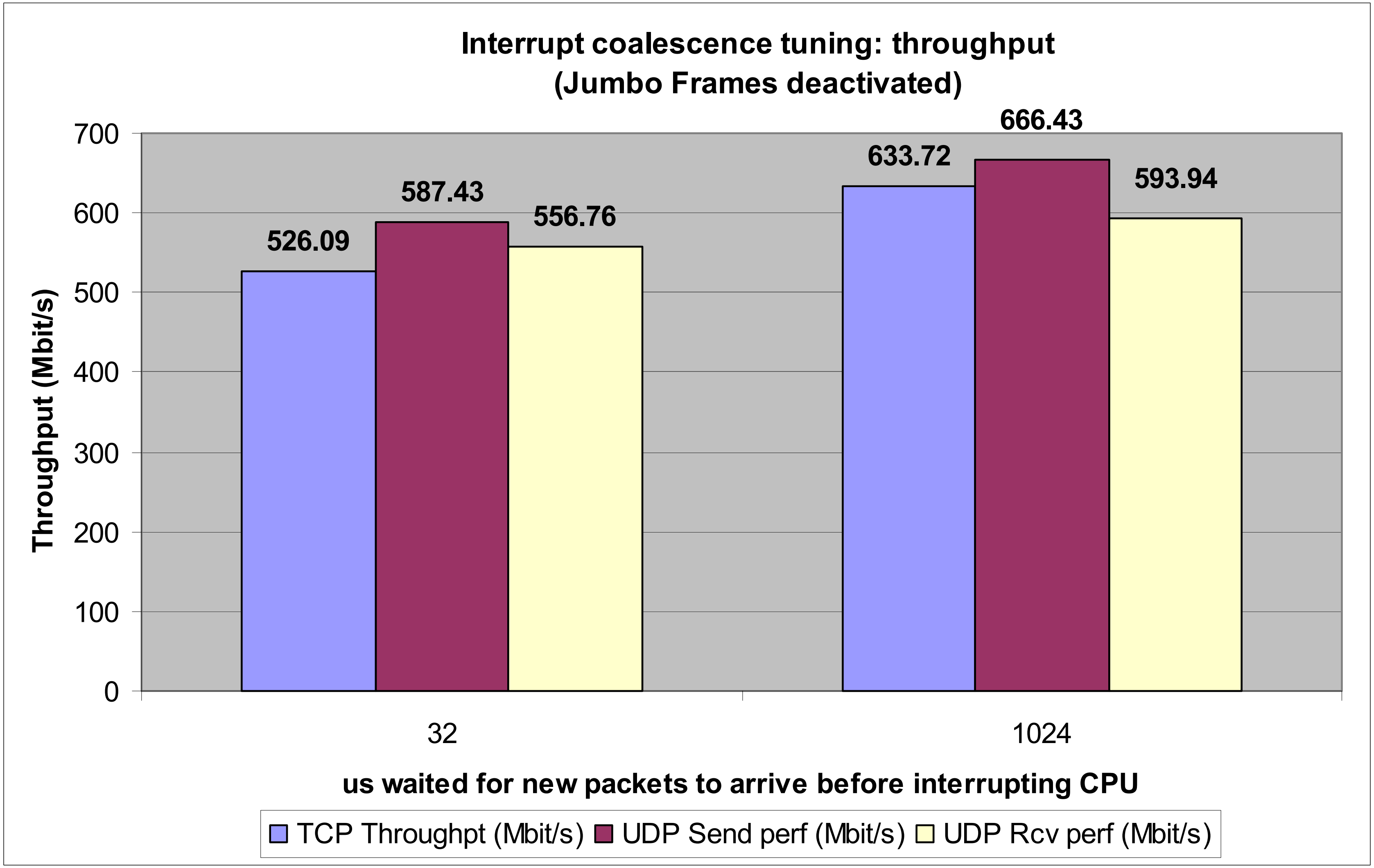
Packet Processing without Interrupt coalescence:



Packet Processing with Interrupt coalescence:



Interrupt Coalescence: Results



Network Optimisation: How?

An "average" CPU might not be able to process such a huge amount of data packets per second:

- -TCP /IP Overhead
- -Context Switching
- -Packet Checksums

An "average" PCI Bus is 33 MHz, 32-bit wide.
Theory: 1056 Mbit/s
Actually: ca. 850 Mbit/s
(PCI overhead, burstsize)

Reduce per packet Overhead:
Replace TCP with UDP

Reduce number of packets:
Jumbo Frames

Reduce context switches:
Interrupt coalescence

Reduce context switches:
Checksum Offloading

Checksum Offloading

- A checksum is a number calculated from the data transmitted and attached to the tail of each TCP/IP packet.
- Usually the CPU has to recalculate the checksum for each received TCP/IP packet in order to compare it with the checksum in the tail of the packet to detect transmission errors.
- With checksum offloading, the NIC performs this task. Therefore <u>the CPU does not have to calculate the checksum</u> and can perform other operations in the meanwhile.

Network Optimisation: How?

An "average" CPU might not be able to process such a huge amount of data packets per second:
- -TCP/IP Overhead
- -Context Switching
- -Packet Checksums

An "average" PCI Bus is 33 MHz, 32-bit wide.
Theory: 1056 Mbit/s
Actually: ca. 850 Mbit/s
(PCI overhead, burstsize)

Reduce per packet Overhead:
Replace TCP with UDP

Reduce number of packets:
Jumbo Frames

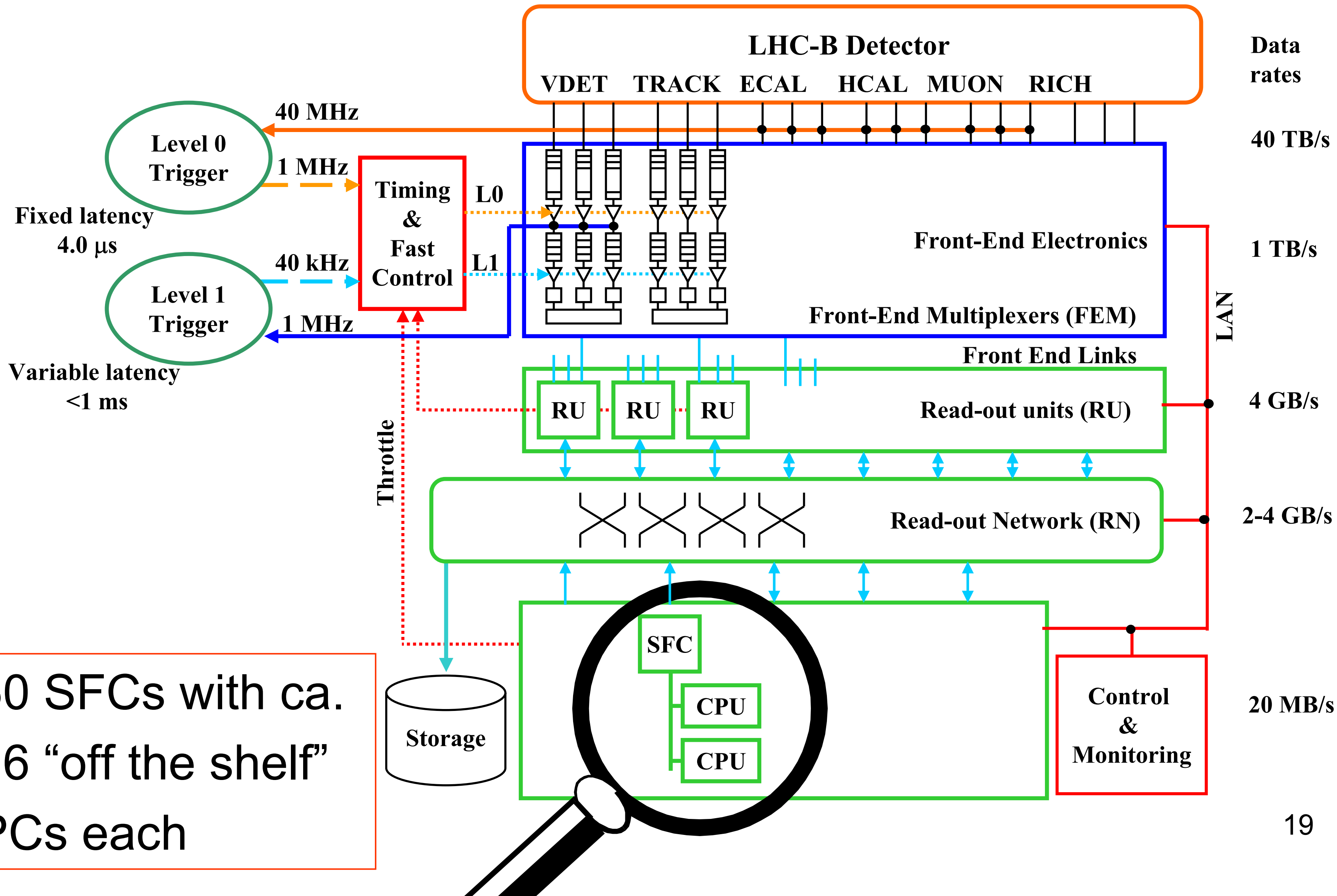
Reduce context switches:
Interrupt coalescence

Reduce context switches:
Checksum Offloading

Or buy a faster PC with a better PCI bus… ☺

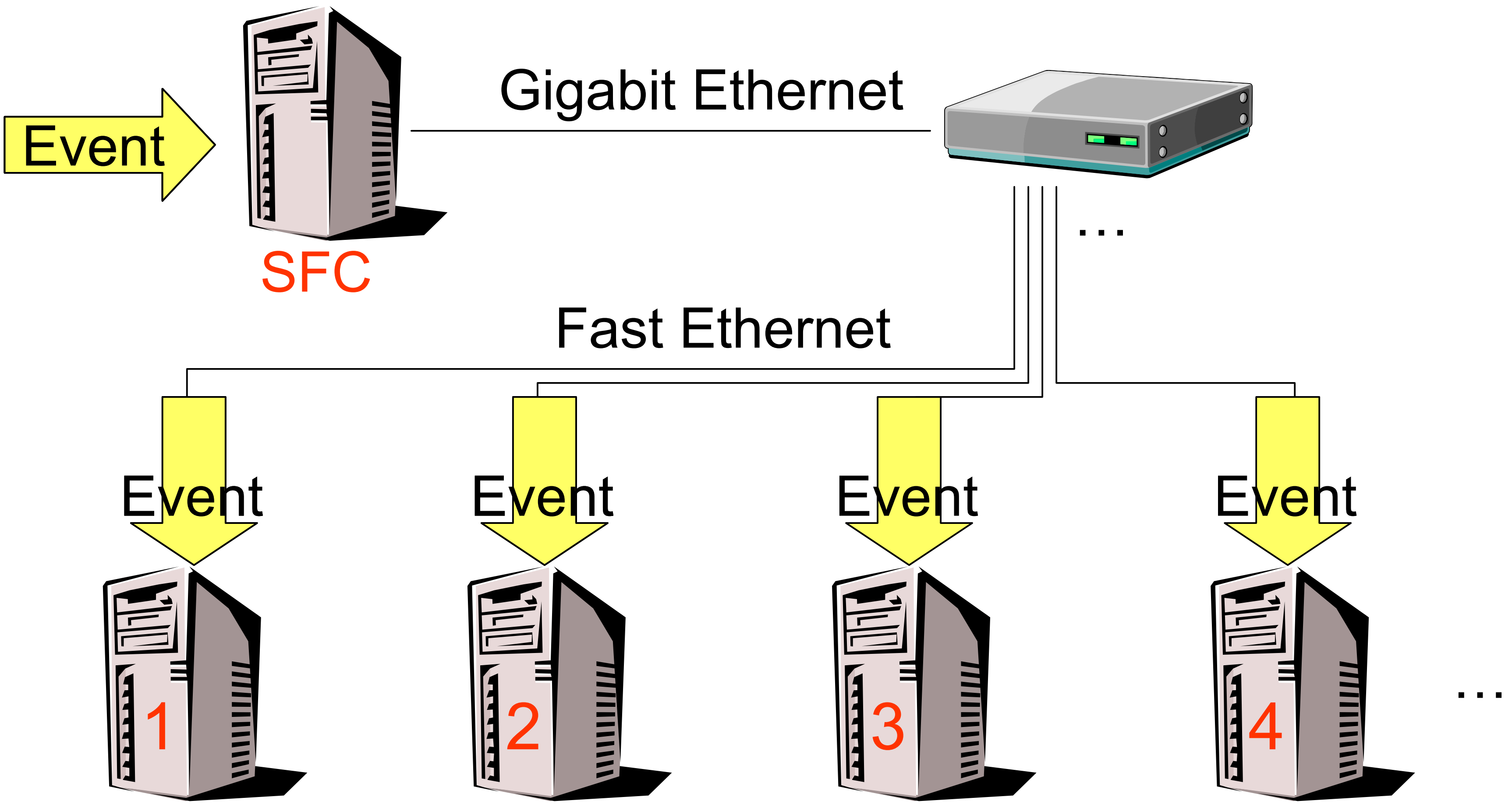
Load Balancing

Load Balancing: Where?



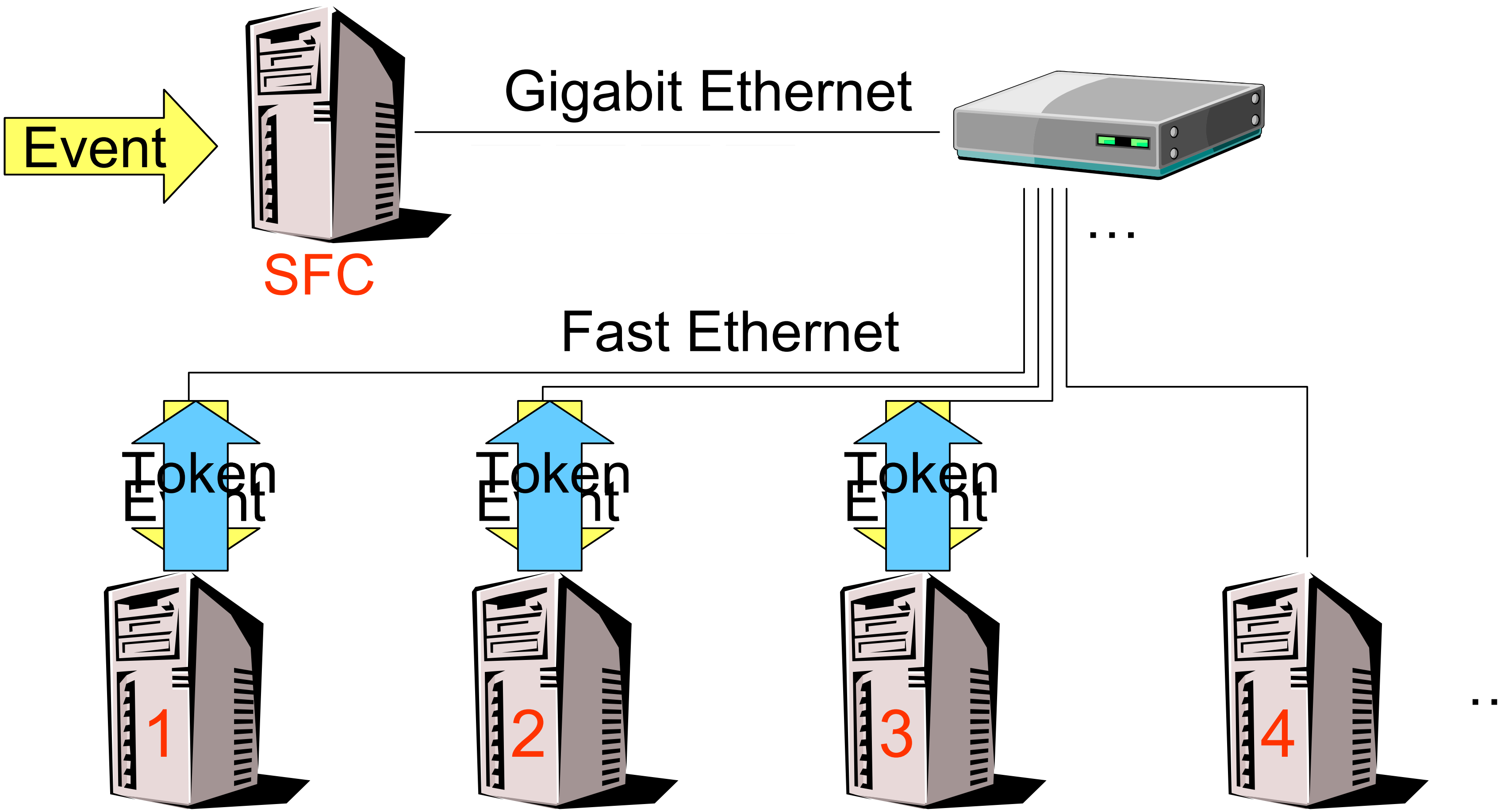
60 SFCs with ca. 16 "off the shelf" PCs each

Load Balancing with round-robin



Problem: The SFC doesn’t know if the node it wants to send the event to is ready to process it yet.

Load Balancing with control-tokens



With control tokens, nodes who are ready send a token, and every event is forwarded to the sender of a token.

LHC Comp. Grid Testbed Structure

100 cpu servers on GE, 300 on FE, 100 disk servers on GE (~50TB), 10 tape server on GE