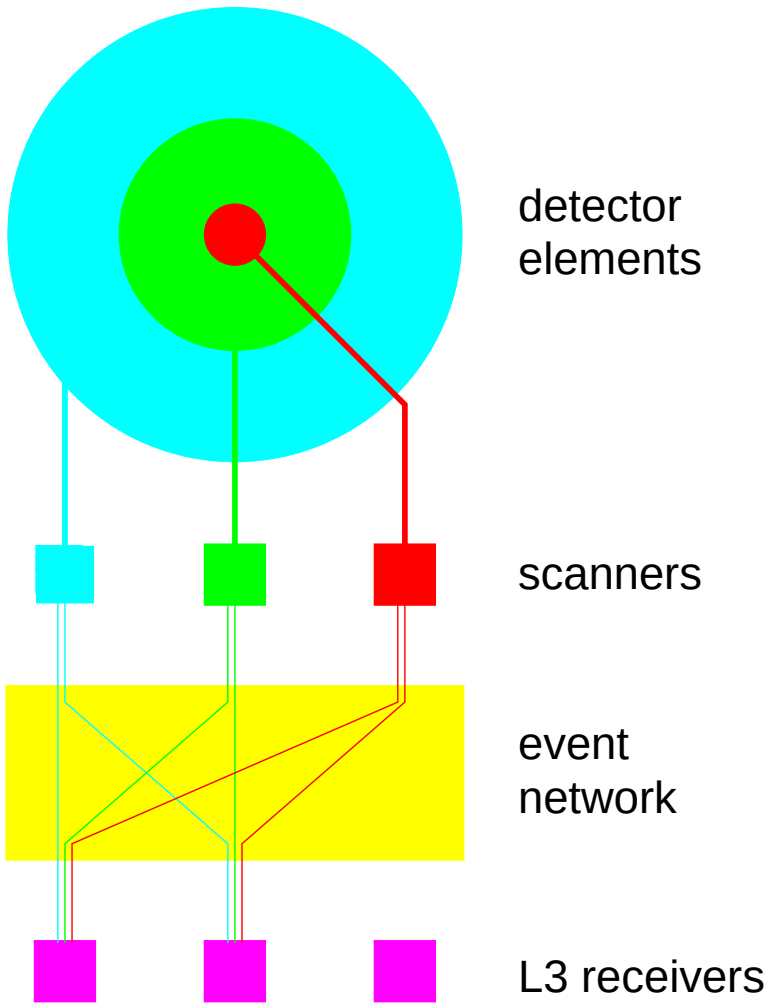
Run II Event Builder: Performance

J. Tseng
MIT

DAQ Upgrade Mini-Review
February 25, 1998

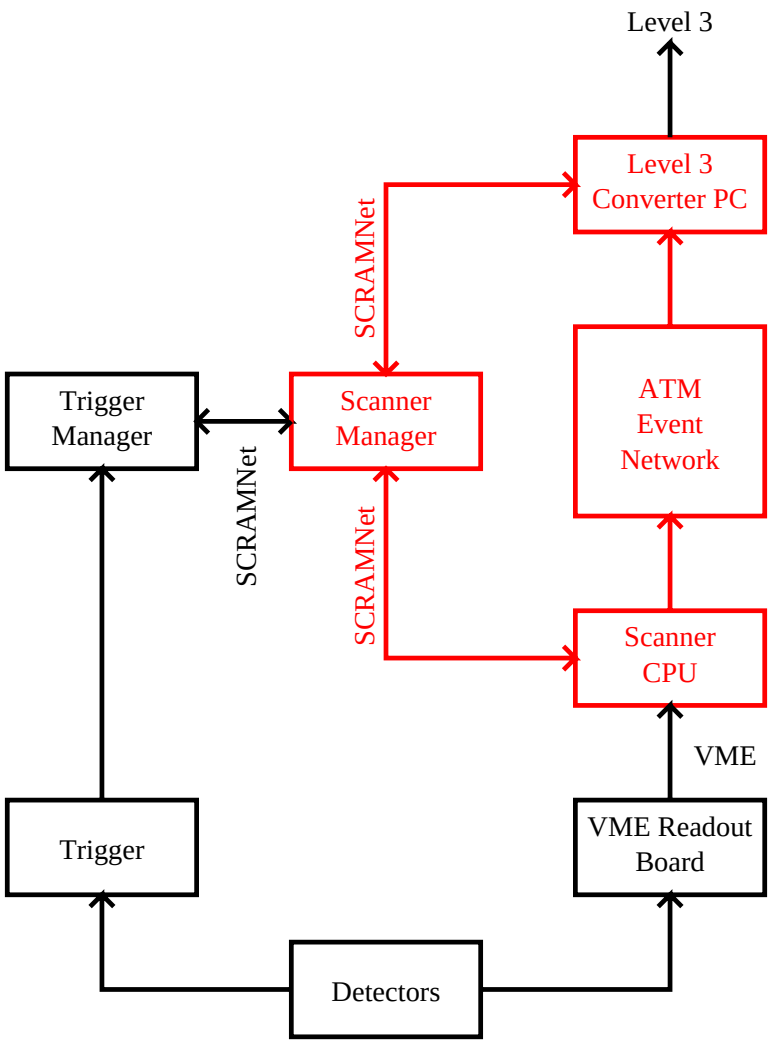
- Overview
- Tests with Expanded System
- VME Input
- Level 3 (PC) Output
- Conclusion

Overview



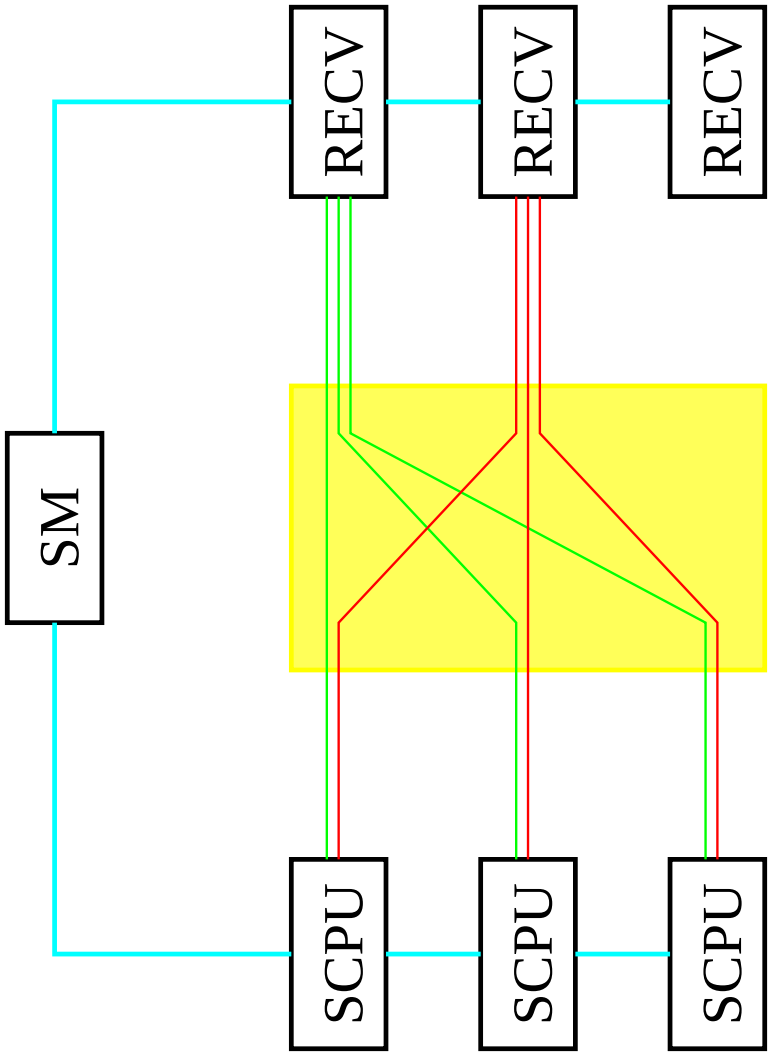
- concentrate $\leq$ 16 geographically separated event fragments into one receiver
- TDR: (300 ev/s)(150 KB/ev) $\approx$ 45 MB/s
  $\approx$ 4 MB/s per L3 receiver link
- working assumption: (300 ev/s)(200 KB/ev) $\approx$ 60 MB/s
- ultimate goal: up to (1000 ev/s)(250 KB/s) $\approx$ 250 MB/s
  $\approx$ 15 MB/s per L3 receiver link

Overview



- Scanner CPU: reads data fragments from VRB, sends fragments through event network
- Receiver: collect data fragments from event network, pass event data to Level 3
- Scanner Manager: talks with trigger manager, controls event flow through network

Overview



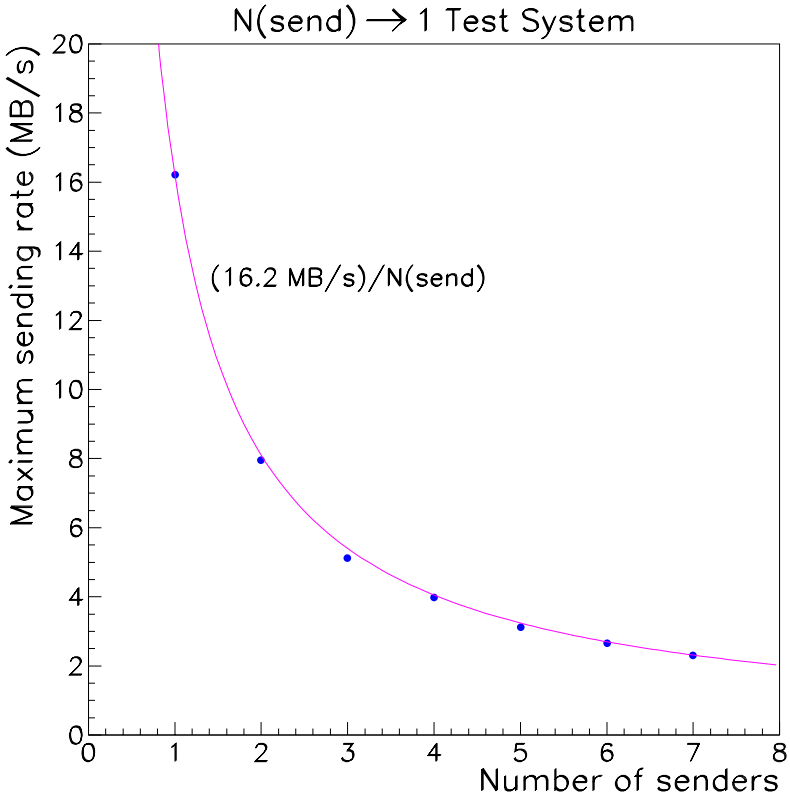
- establish quasi-permanent virtual connections between each scanner and receiver
- scanners send data simultaneously, but at a fraction of their full sending bandwidth so as not to overflow receiver
- use remaining sending bandwidth to send to other receivers at the same time
- number of receivers = number of events that can be built simultaneously

Previous ($\sim$ 1 year) Tests (I)

Idea:

- consider an $N_{send} \rightarrow 1$ system
- to avoid overflowing the receiver and losing cells, each sender throttles speed to

$$v_{limit} = \frac{v_{max}}{N_{send}}$$

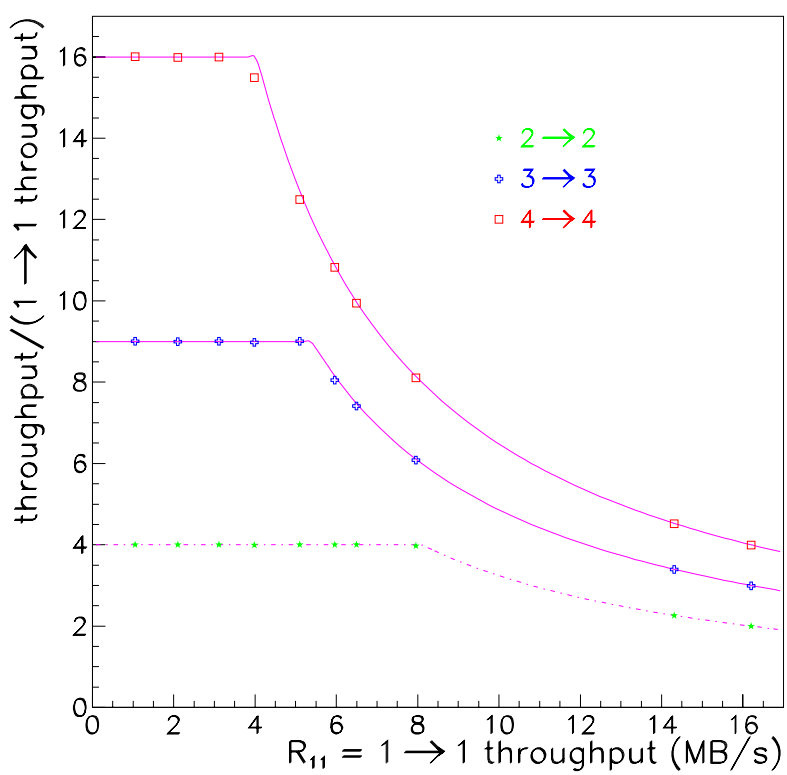


- receiver sees $N_{send} v_{limit} = v_{max}$

Previous ($\sim$ 1 year) Tests (II)

- use the rest of each scanner's bandwidth to send to other receivers
- in an $N_{send} \rightarrow N_{recv}$ system, total throughput is *linear* in N_{recv}:

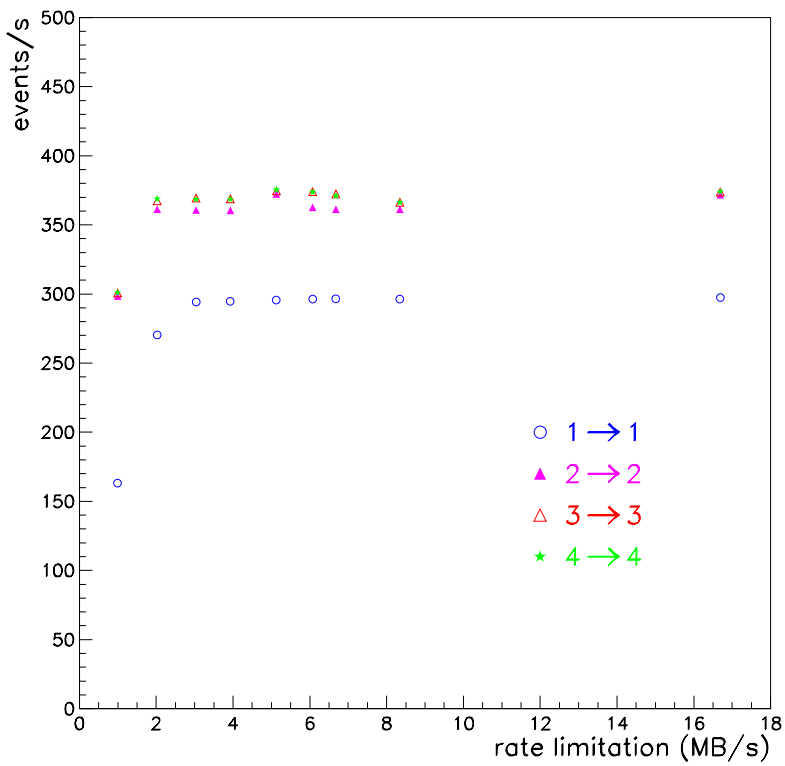
$$\begin{aligned} v_{total} &= N_{send} N_{recv} v_{limit} \\ &= N_{recv} v_{max} \end{aligned}$$



Previous ($\sim$ 1 year) Tests (III)

previous event builder system tests

- showed scaling behavior for $N > 1$

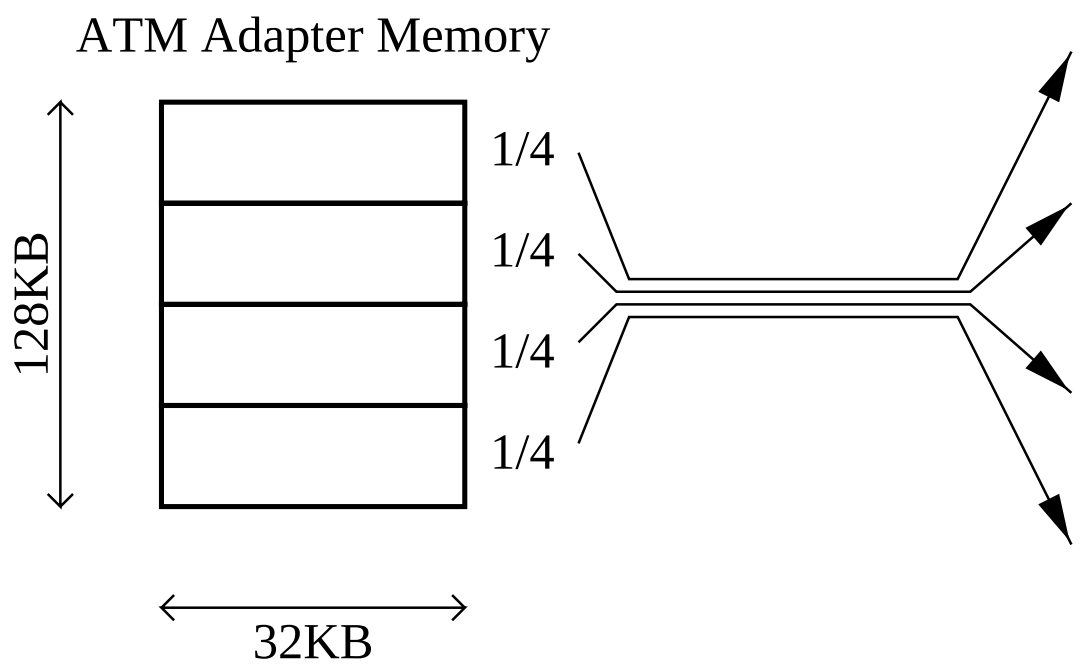


- slower computers (Radstones)
  $\Rightarrow$ large control overhead
  ($v_{max} \approx 6\ MB/s$)

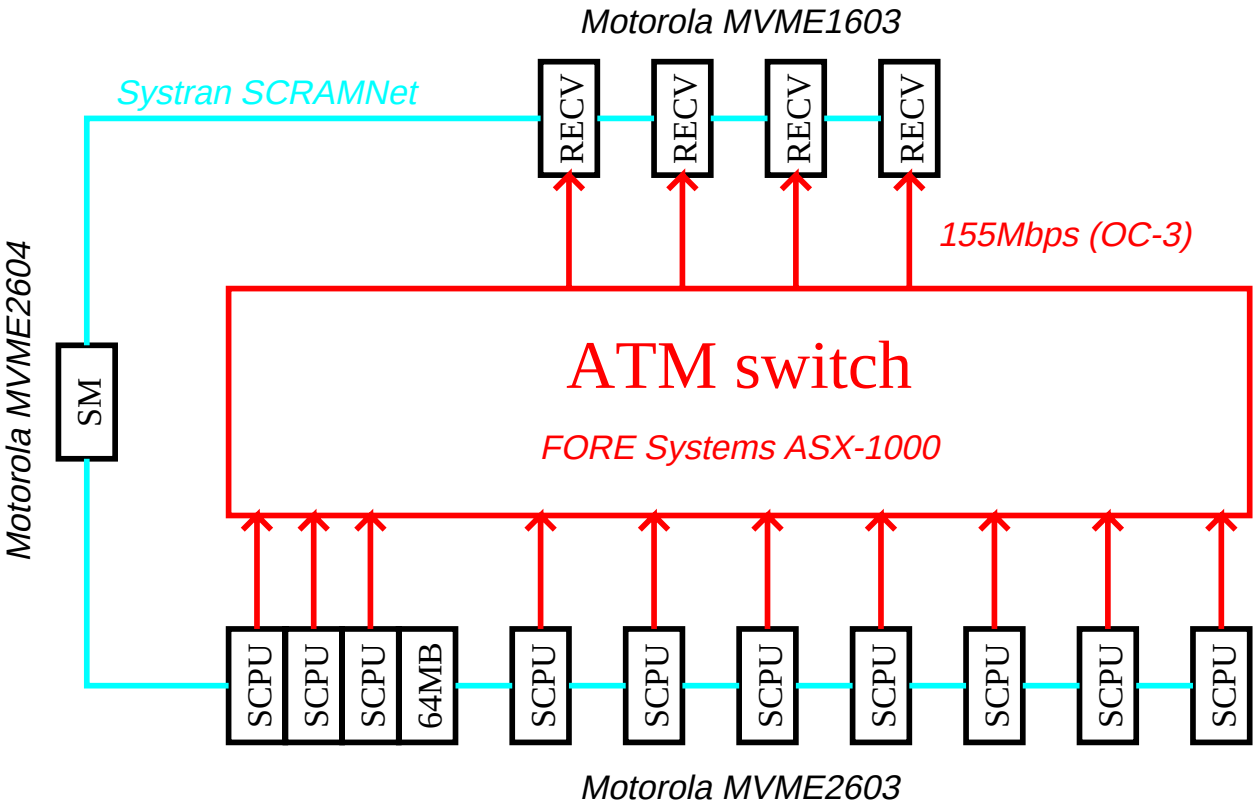
Previous ($\sim$ 1 year) Tests (IV)

limitations of previous tests:

- some computers had slower memory bus
- only up to 4 senders (and 4 receivers)
- fixed-size 16 KB fragments
- old ATM adapters had 128 KB on-adapter RAM
  (could only send 4 fragments simultaneously)



Tests with Expanded System (I)



- more ports on ATM switch
- same command network (SCRAMNet)
- Scanner Manager: MVME2604 (200 MHz)
- Scanner: MVME2603 (200 MHz), 1 MB ATM adapter
- Receiver: MVME1603 (66 MHz), 128 KB ATM adapter

Tests with Expanded System (II)

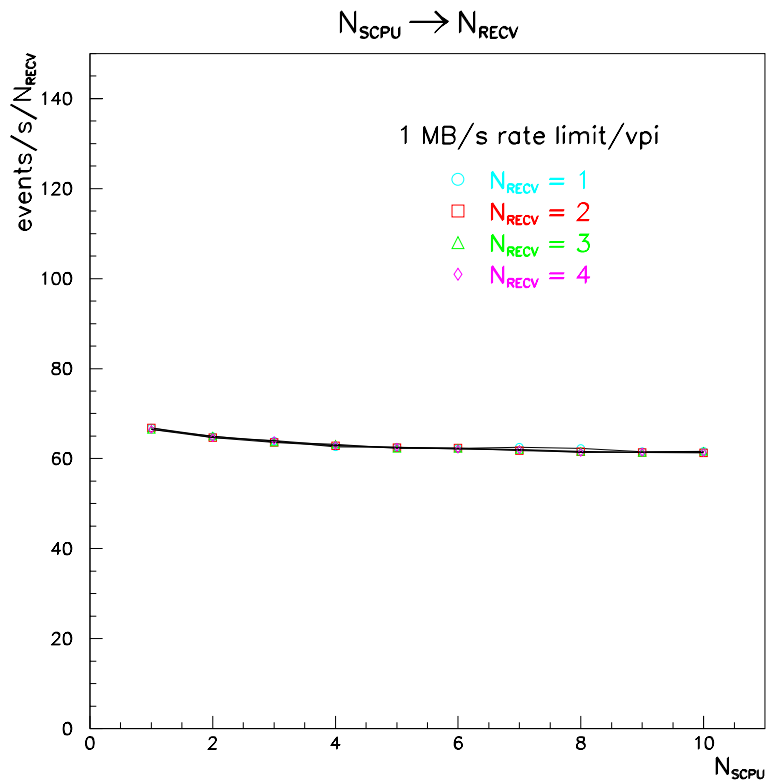
- new ATM adapter can send more fragments simultaneously (32×32 KB)
- can test up to 10 scanners and 4 receivers (homogeneous CPU types for scanners/receivers)
- each fragment varies in length as a truncated gaussian with mean 16 KB, std. dev. 4 KB, minimum 8 KB, maximum 32000 B

$\Rightarrow$ new tests show the scaling behavior of the previous tests

Tests with Expanded System (III)

- on each scanner, set rate limitation to that appropriate for a 16 $\rightarrow N_{recv}$ system

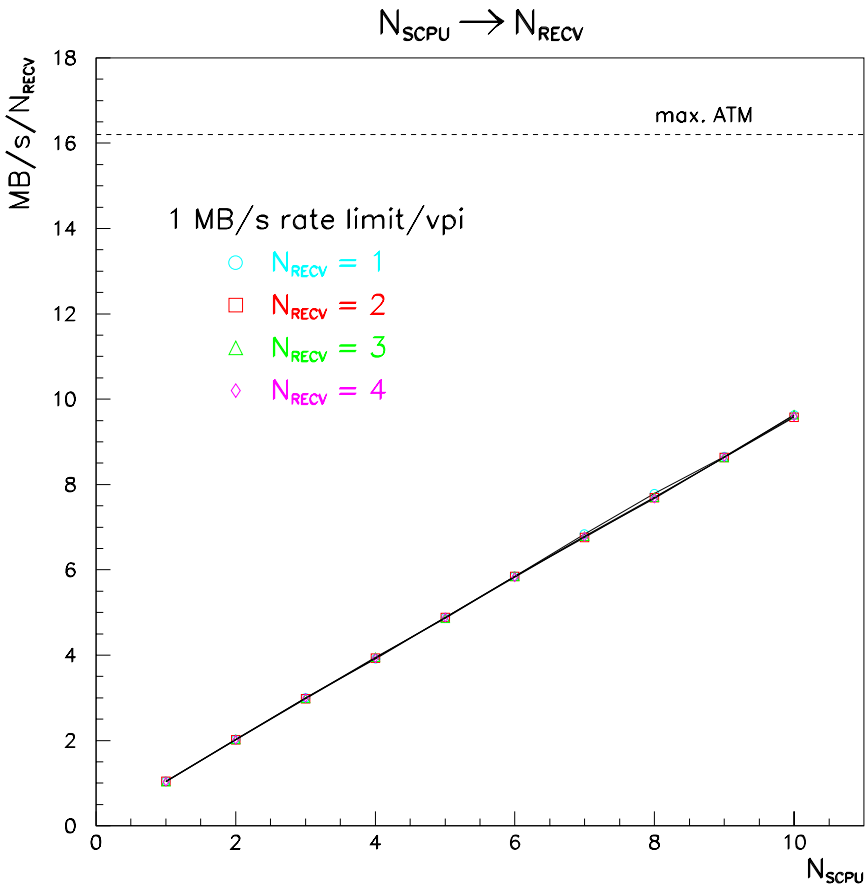
$$v_{send} = \frac{1}{16} v_{max} = 1\ MB/s$$



- event rate linear in N_{recv}
- slight drop as N_{SCPU} increases
- fixed fragment lengths eliminate drop
- suggests up to $\sim$ 1000 ev/s with 16 receivers

Tests with Expanded System (IV)

- set rate limitation to 1 MB/s

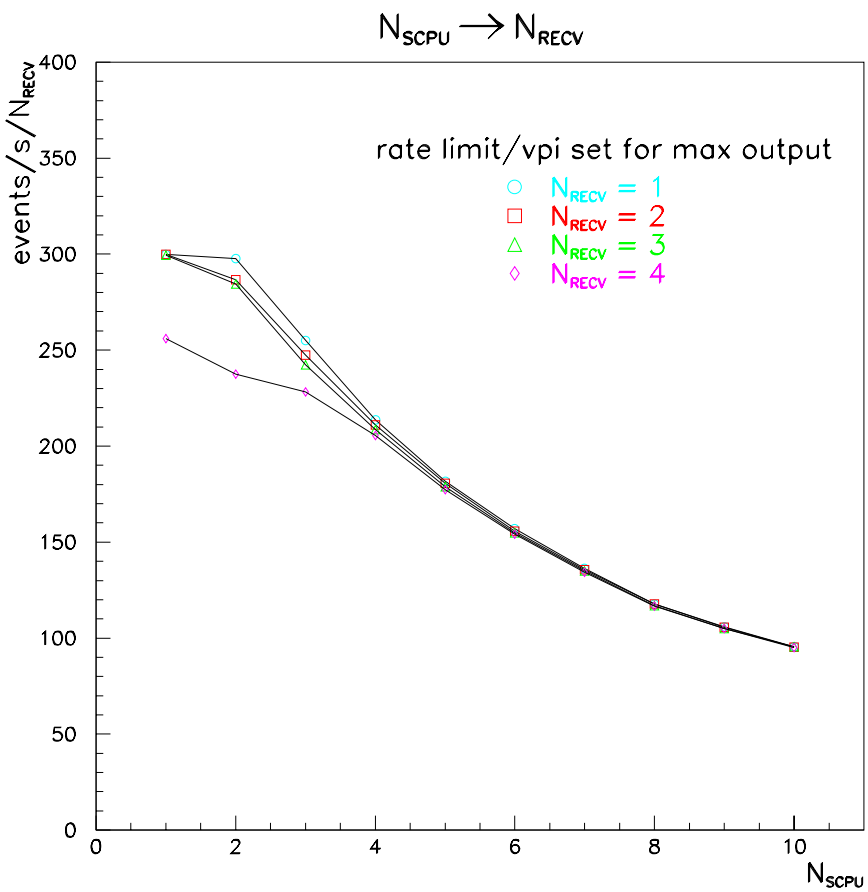


- data throughput scales with N_{SCPU}
- does not saturate senders ($v_{max} = 16.2$ MB/s)

Tests with Expanded System (V)

- set rate limitation on each scanner so as to saturate receivers:

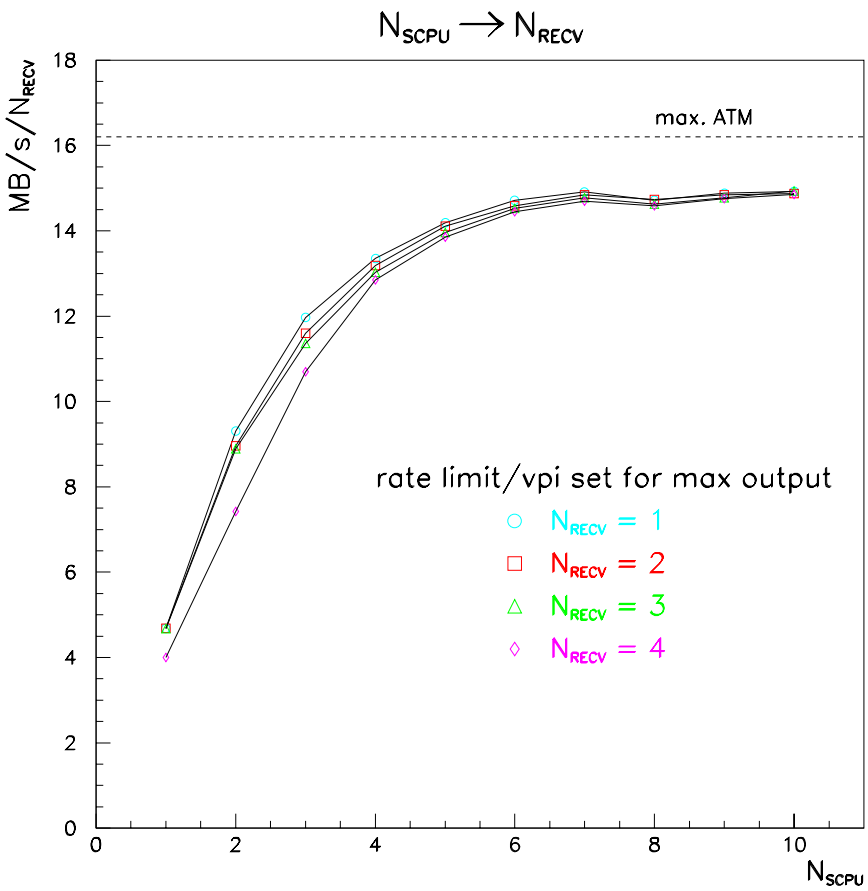
$$v_{send} = \frac{1}{N_{SCPU}} v_{max}$$



- falling event rate due to having to throttle transmission rate as increase N_{SCPU}

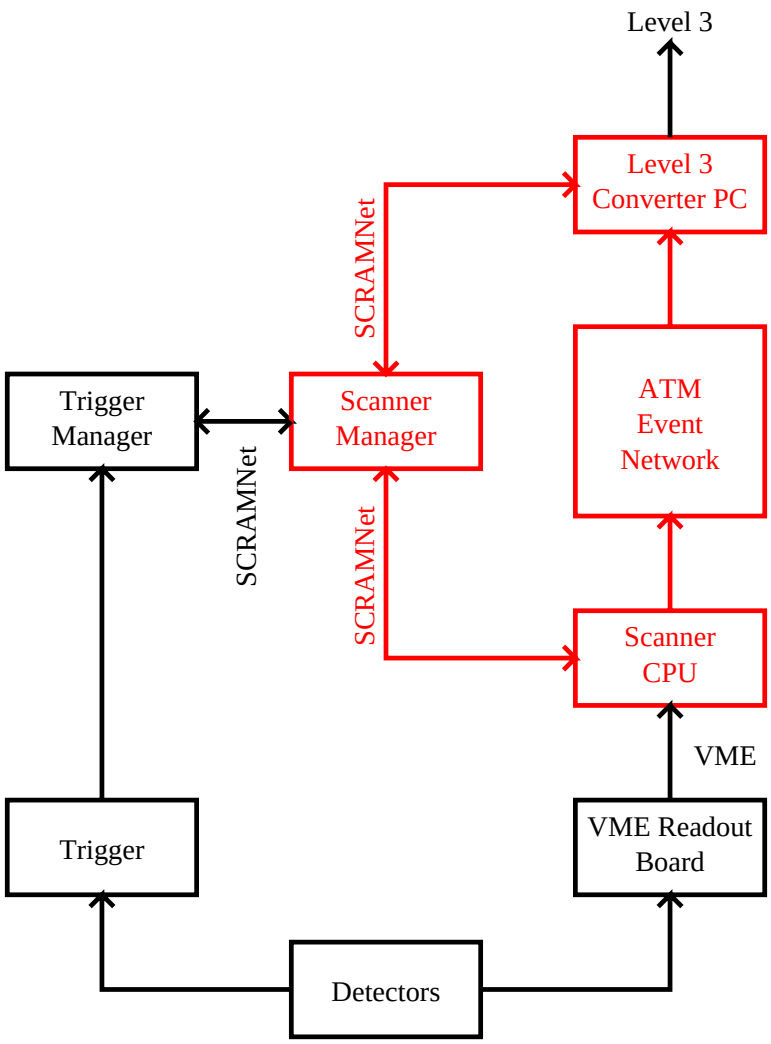
Tests with Expanded System (VI)

- set rate limitation on each scanner so as to saturate receivers



- plateau $\sim$ 14.5 MB/s with 8 scanners (we will have 16 scanners)
- overheads create inevitable idle times

Connections to Outside World



- trigger manager interface
  (prototype Scanner Manager code)
- VME input from a dedicated memory
- PC receiver

Add VME Input

- use J. Pangburn's FISION package
- scanners simultaneously transfer data over VME and ATM
- both transfers occur over the PCI bus
- independent DMA rates on MVME2603:
  - VME: 20 MB/s with FISION defaults (30 MB/s without)
  - ATM: 56 MB/s (without sending over fiber)
- see if event throughput drops
- if any bottlenecks are revealed, work on them

Tests with VME Input

configuration	SCPU throughput without VME input		SCPU throughput with VME input	
	ev/s	MB/s	ev/s	MB/s
$1 \rightarrow 4$	1024	16.0	850	13.3
$1 \rightarrow 3$	898	14.0	875	13.7
$1 \rightarrow 2$	600	9.4	600	9.4
$1 \rightarrow 1$	300	4.7	300	4.7

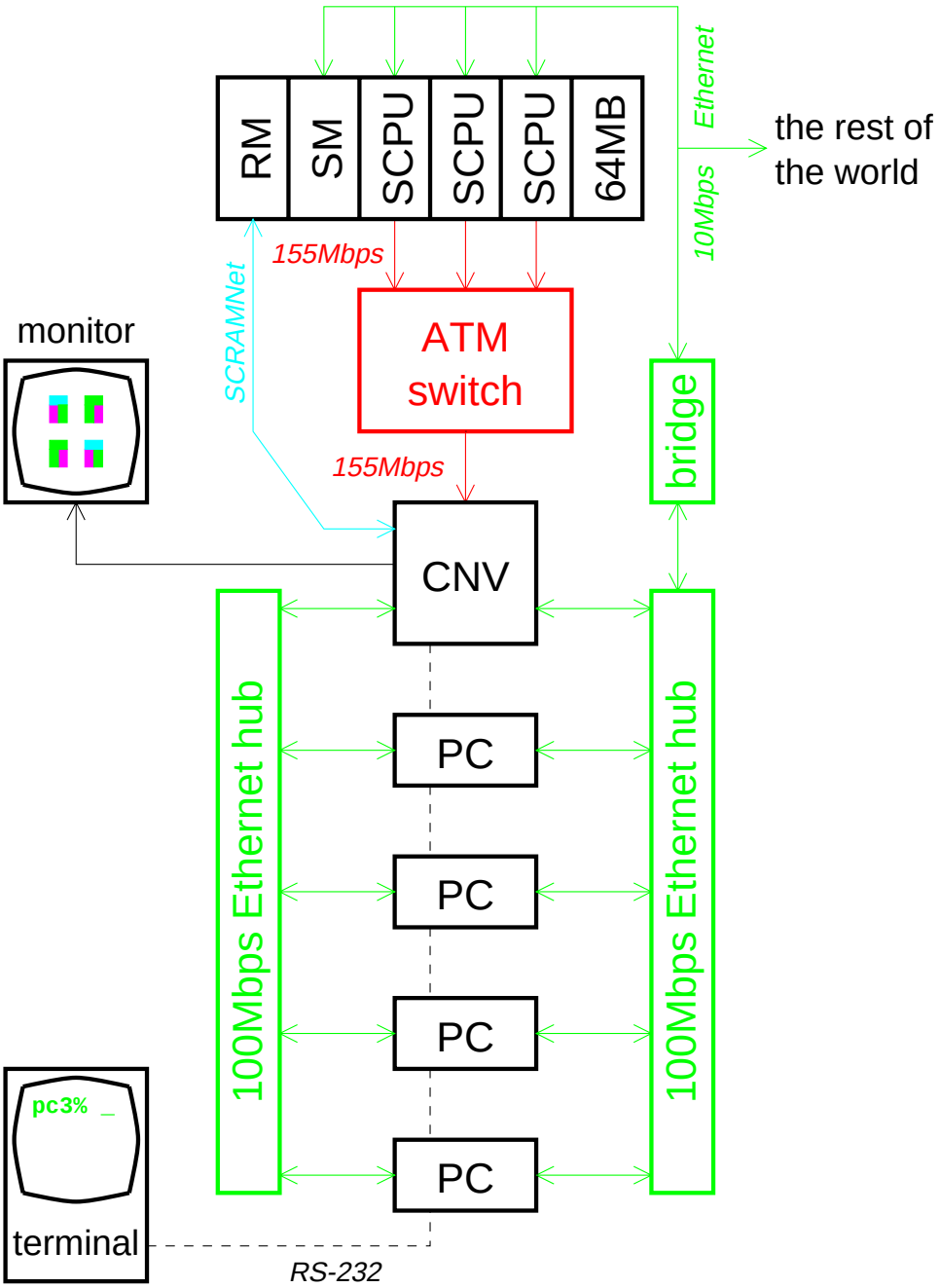
- VME input prevents data throughput from reaching 16 MB/s
- probable bottleneck:
  - current VME code blocks all other tasks while waiting for DMA to complete
  - could also be hitting PCI bus pretty hard
- investigating other synchronization methods:
  - FISION asynchronous mode
  - interrupt signals

Tests with PC Receiver (I)

- use an "off-the-shelf" Pentium-based personal computer equipped with
  - PCI-SCRAMNet interface (Systran)
  - PCI-ATM adapter (FORE Systems)
  - dual 100Mbps Ethernet interface
  - Fermilab-supported Linux (Red Hat 5.0) (with customizable kernel and drivers)
- receiver code is the same as that used on the MVME1603's
  (POSIX threads instead of VxWorks tasks)
- runs control software based on Run 1 Level 3 framework
- this node is often called the "converter" node since it converts its ATM input to Fast Ethernet output to the "compute" nodes

Tests with PC Receiver (II)

Demonstrated at SC97
San Jose, CA, November 1997

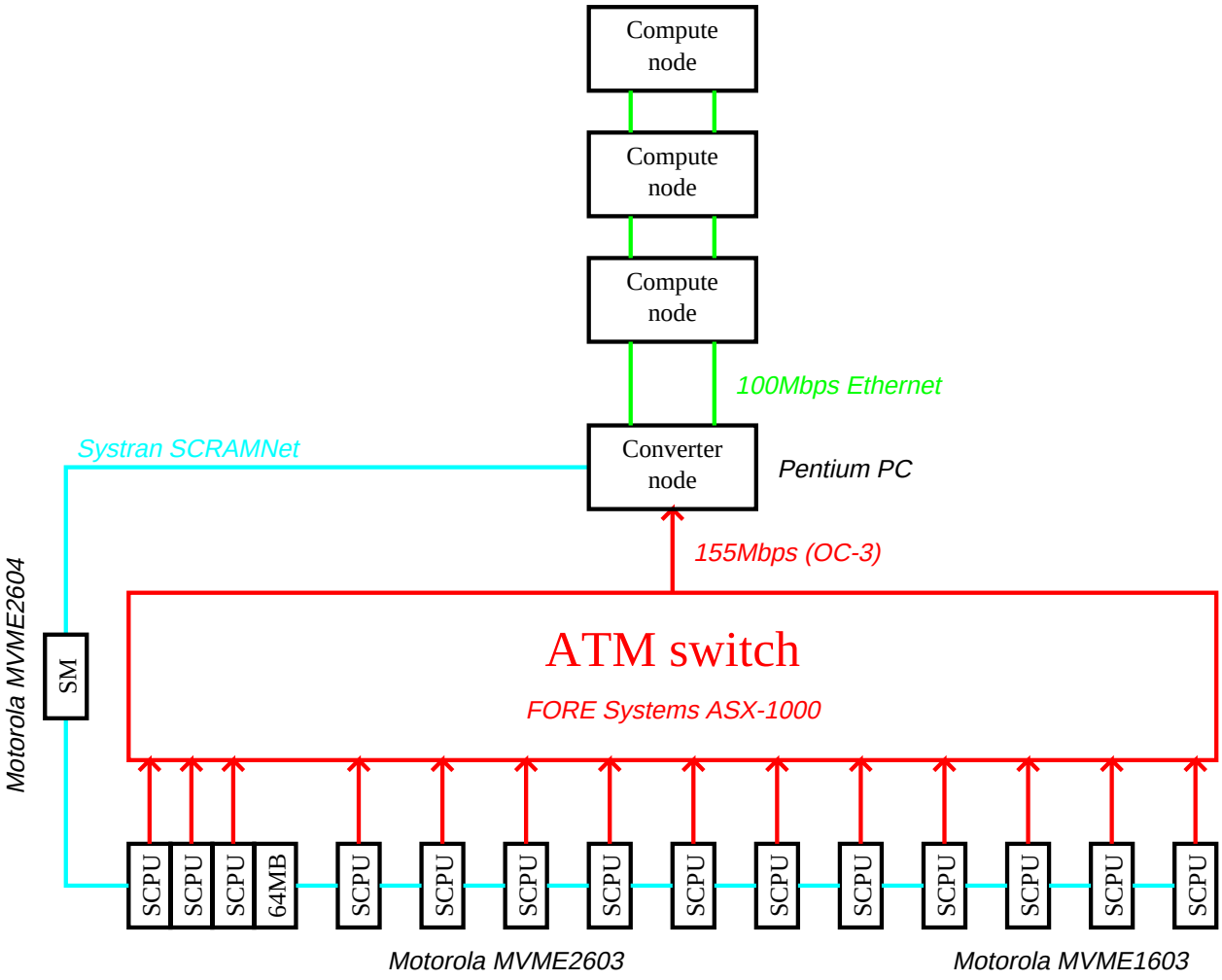


Tests with PC Receiver (III)

- VME input (Run 1 DST events)
- send to 1 converter node, which discards the event
- variable fragment sizes, (16 ± 4) KB
  $\Rightarrow$ small systems perform identically to those in scaling tests; $8 \rightarrow PC$ was 5% slower than $8 \rightarrow 1$ with Motorola
- under development:
  - optimizing Level 3 control framework
  - optimizing ATM driver, including increase in receiving buffer space

Tests with PC Receiver (IV)

Largest Slice Yet



52 ev/s

caveat: converter node moves 150000 bytes regardless of how much it receives (for now...)

Conclusion

- event builder tests continue to show scaling behavior
  $\Rightarrow \sim$ 1000 ev/s with 16 receivers
- implemented prototype interfaces to trigger manager and VME input
- current VME input method slows the system, but further development looks promising
- PC receiver works;
  optimization well underway
- largest slice yet: $14 \rightarrow PC \rightarrow 3$