EvB 2002

Major Categories of issues/work

- Performance (event rate)
- Hardware
  - Next generation of PCs
  - Network upgrade
- Control (Sean)
  - Solving CORBA speed/reliability problems
  - Detailed monitoring of components
- Infrastructure (Sean)
  - Compiler, STL implementation
  - Error logging

EvB Performance

Current Status

- Data rate
  - Have achieved rates of > 150 Mbyte/s.
  - No throughput limitations observed so far.
  - Have x3 headroom w/ existing hardware.
- Event rate
  - Currently limited to ~ 1.3 kHz for full system.
  - Individual SEBs can run as fast as 2.3 kHz.
    - ⇒ e.g. GL1 SEB On 933 MHz machines
    - ⇒ More typically: 1.8 kHz for “larger” granules
  - Evidence that event rate depends on data volume (issue w/ PCI performance ?)

EvB Performance – Event Rates

How to increase event rates?

- Use full functionality of JSEB
  - Multiple event buffering
  - Interrupts instead of polling
    - ⇒ Effects of these changes unknown.
    - ⇒ But will be studied in existing system post-run.
- Code improvements
  - Have a number of code improvements “in hand” that we have not been able to fully test.
  - Many more places for optimization.
  - Reasonably achievable goal:
    - ⇒ > 3 kHz with existing hardware

EvB Performance – Event Rates (2)

- Hardware upgrade
  - Dual processor machines
    - $\Rightarrow \approx$ factor of 2 (will verify after run using ATP PCs)
  - Faster CPUs: e.g. > 2 GHz P4 w/ DDR memory
    - $\Rightarrow$ SEB performance scales $\propto$ CPU clock speed.
    - $\Rightarrow \approx$ factor of 2
- Plan: 3 kHz $\times$2 (dual) $\times$2 (cpu) = 12 kHz
- Contingency
  - JSEB optimization
  - Better compiler (?) & improved STL
  - Improved PCI performance (66 MHz, 64 bit)
    - $\Rightarrow$ Will study after run using ATP PCs

Upgrade Goals as of Nov 01

- Goal of original PHENIX CDR: 2 Gbyte/s
  - Based on x10 “design” luminosity
  - Since then “design” has increased by a factor of 2 (4 if 116 bunches works)
    - $\Rightarrow$ Reasonable to expect that RHIC may reach x10 original design in next few years
- Nominally: increase bandwidth by factor of 4
  - Increase data throughput of SEBs
    - $\Rightarrow$ While keeping # $\approx$ constant
  - Increase number & throughput of ATPs
- Tony’s 10 kHz * 150 Mbyte/s = 1.5 Gbyte/s

EvB Upgrade goals – How to Achieve

Nominally

- Upgrade to 40 Gbit/s switch.
- SEBs w/ OC-12 connections (> 60 Mbyte/s)
- 128 ATPs w/ OC-12 connections.
- Problem:
  - prohibitively expensive if we buy from FORE

    ⇒ ~ $500k for ATM hardware

Alternative

- Switch to Gbit ethernet.

  ⇒ > ATM solution for ¼ cost

EvB Upgrade – Gbit Ethernet

Possible implementation

- HP Procurve (256 Gbit/s)
  - Provides 64 Gbit ethernet ports
  - Inefficient to equip w/ fast ethernet – use Gbit for ATPs or split w/smaller switches.
- In principle, get 4 Gbyte/s throughput
- Cost estimate

Component	Quantity	Component Price	Total
HP Procurve 9308 switch (256 Gbit/s backplane)	1	$15,000.00	$15,000.00
8-Port Gbit ethernet Card for HP 9308	8	$9,000.00	$72,000.00
HP 2524 Fast ethernet switch w/ Gigabit uplink	2	$969.00	$1,938.00
1000Base-T uplink card for HP2524	4	$375.00	$1,500.00
Intel Gbit ethernet NIC	64	$150.00	$9,600.00
Cat5e (Gigabit capable) cables	64	$60.00	$3,840.00
Quad CPU Rackmount Server w/ RAID system	2	$8,200.00	$16,400.00
Dell 1550 dual 1.4 GHz rackmount systems (SEBs)	32	$2,500.00	$80,000.00
Quad cpu Beowulf nodes	32	$4,000.00	$128,000.00
Total			$328,278.00

Practical Considerations

Technical Issues

- There were (are) some good reasons for preferring ATM over ethernet:
  1) Multicasting
  2) Flow control (to protect against output port overloading)
  3) Hardware CRC and frame re-assembly
- Professional performance studies:
  - Layer-3 switches now available can handle #1 & #2
  - #3 is still an issue. But some NICs moving this functionality to hard(firm)ware.
- Would still choose ATM but for cost difference

Practical Considerations

How hard would the switch to Gbit be ?

- In principle – not that hard.
  - Implementation of ATM communication uses socket interface (Winsock2)
  - Change to TCP/UDP "straight-forward" as we already have C++ classes that wrap Winsock2 TCP sockets.
  - Winsock provides mechanism for establishing multi-cast groups/flow control.
- If we stick with NT for now, migration to Gbit ethernet should be easy.

Gbit Ethernet – Study Plan

How to make sure Gbit ethernet will work

1. Study single PC send/receive rates
   a) Use ATP machines (66 MHz, 64 bit PCI)
   b) Evaluate for both TCP and UDP
      i. achievable rates
      ii. CPU utilization
   c) Check performance of different NICs.
   d) Learn how to use flow control in Winsock.
2. Experiment w/ fast ethernet using Martin's switch (?)
   a) In same machines compare w/ ATM.
3. Test JSEB reading/sending to evaluate PCI

EvB – Robustness Issues

- SEBs
  - Hanging/crashing (unhandled exception) on bad FEM data.
    - ⇒Simply needs further debugging
  - GL1 data errors, FEM bad event numbers
    - ⇒Need logging system to alert operator
  - Unhandled exception in transmit thread (rare but more frequent at high rate)
    - ⇒Needs debugging.
- EBC
  - Problem with end run in optimized code
    - ⇒Disappears in v3_0c

EvB Robustness (cont)

- EBC (cont)
  - Occasional crashes w/ high-rate Au-Au running
    - ⇒ Source of problem understood. Needs work to clean up end run procedure in EBC.
- ATP
  - Problem with ET failures largely solved.
  - Problem w/ ATP thread killing/restarting (Sean)
  - Problem with ATPs starting to generate unhandled exceptions every event (@ high rate)
    - ⇒ Needs debugging time.
  - ATP access to Obj database
    - ⇒ Need to delay database initialization

EvB Open Issues / Other work

- CORBA
  - Slow setup of Corba connections (Sean)
  - Upgrading Orbix on EvB machines
- Logging
  - Critically important for reporting failures (GL1, FEM, EvB, ...) to operator.
- Flow control
  - Need to automate flow control settings.
- Use (redundant) application/file server
  - Simplify system setup.
  - Avoid need to manually distribute code.