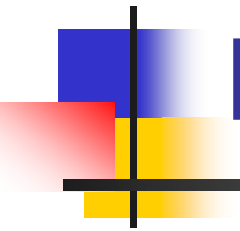
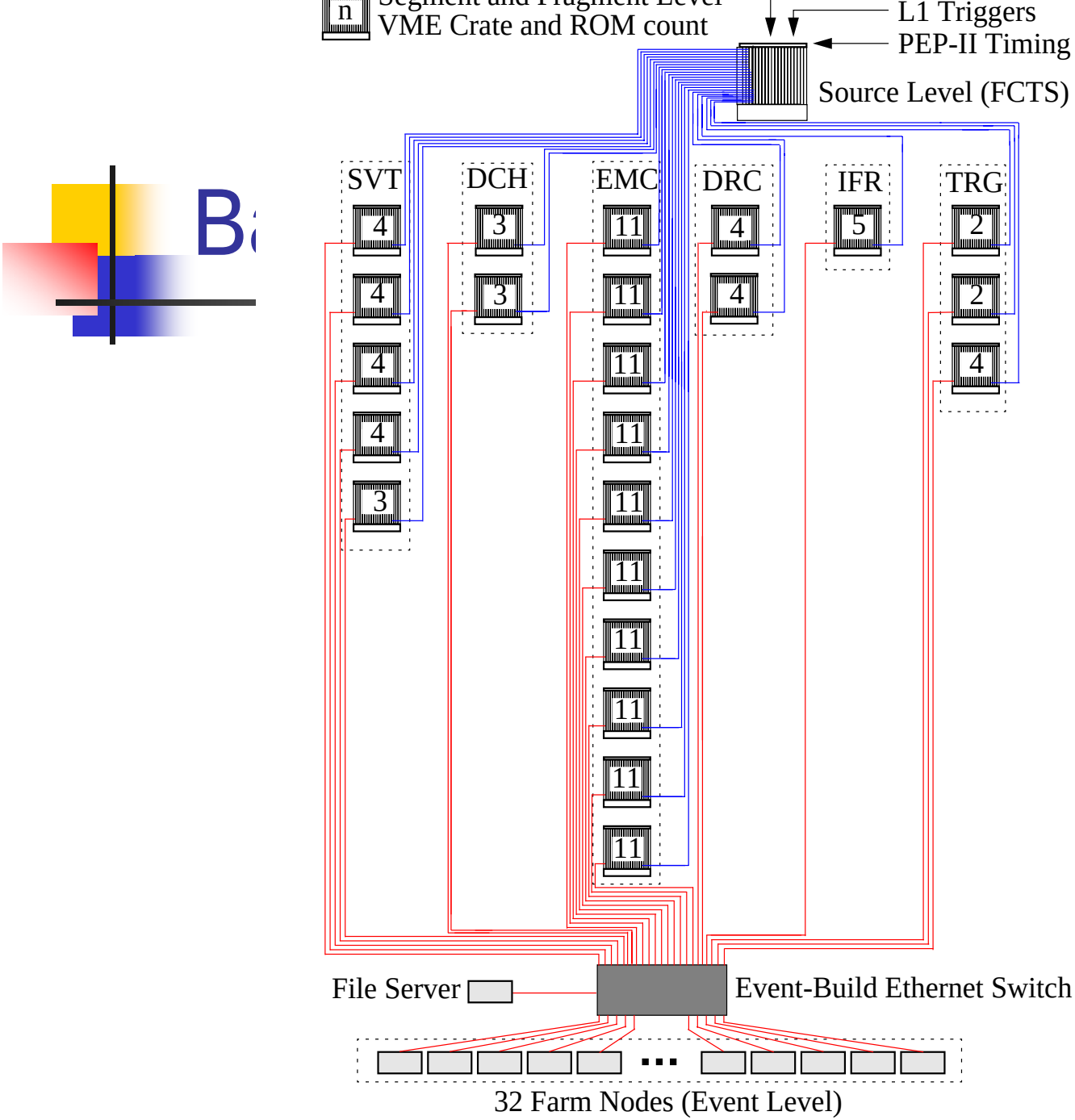


Babar DAQ Experiences

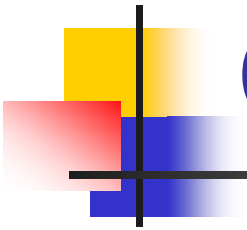
Ideas that worked,
Ideas that were problematic ...

Christopher O'Grady, SLAC
BaBar Dataflow Group leader since 2000


n
VME Crate and ROM count
L1 Triggers
PEP-II Timing
Source Level (FCTS)
SVT
DCH
EMC
DRC
IFR
TRG
4
4
4
4
3
3
3
11
11
11
11
11
11
11
11
11
11
4
4
5
2
2
4
File Server
Event-Build Ethernet Switch
32 Farm Nodes (Event Level)

DAQ System Overview

- 5kHz L1, 2% deadtime (we hope).
- ~200 CPUs (vxWorks, UNIX), ~400 fiber optic cables, GBit/100Mbit ethernet, VME.
- First 4 event buffers in front-end elex.
- “Partitionable”.
- About $1M in equipment.



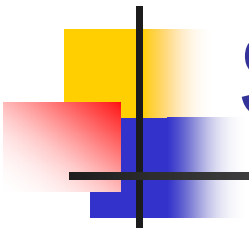
Caveats

- Opinions are mine. Others do disagree.
- I don’t have perfect answers for the problematic areas.
- I was not the system designer (but worked on the system since 1997).

Problem Area: Control of Firmware

- Version control not well done
- Engineers don't know about code archival tools.
- Changing firmware sometimes harder than changing hardware!!
  - Chips no longer supported by new compilers and programming tools
  - Old compilers/programming-tools no longer exist, or won't run because old license server nodes gone away. Old OS versions not permitted because of security concerns. Old software support gone because of corporate mergers.
  - => Firmware "cast in stone" quickly compared to software.
- Interestingly, firmware coding is now more like big software coding. Hardware engineers need to learn from software engineers.

Problem Area: Software Coupling (I)



- Software re-used by both offline and online
  - sometimes re-used for good reasons.
  - sometimes re-used for less-than-ideal reasons
    - time pressure at startup
    - Programmers unaware of the real cost of software dependencies.
- Led to coupled online/offline compilers and operating systems. e.g. offline wanted compiler upgrade, Dataflow didn't need it. Not optimal. I don't have "the answer", but...
- Lesson: Be careful what you re-use. Keep dependencies/interfaces SMALL. Sometimes better to "copy the .hh" file.

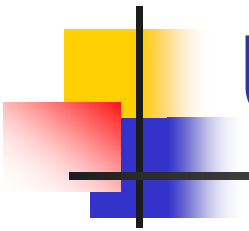
Problem Area: Software Coupling (II)

- Security concerns required OS changes that weren't required from a performance standpoint (although arguably were needed from a maintenance standpoint). I don't have "the answer" but...
- In a future DAQ system, perhaps have a small secure entry into the online system, allowing more freedom for rest of online?

Problem Area: Miscellaneous

- Working with GHz optical links hard.
  - 100 BaBar boards have optical links that don't work at spec. Should have tested earlier with optical attenuators.
- In general, not enough good test software for either the software or hardware. Unavoidable for non space-based experiment, because of cost?. (my brother likes the "write your tests first" approach).
- Neutron radiation and "Single Event Upsets" are a concern at Babar.
- Persistent data format was hard to change.
- Got locked into the mechanical format of a particular Single Board Computer.

Problem Area: Use of UDP and Multicast

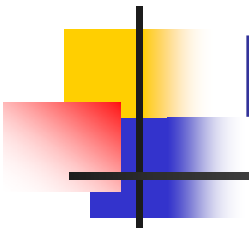


- Can't tolerate packet loss (biases physics).
- Since not widely used for reliable data transfer, some tricky intermittent failures:
  - Different network drivers had different buffering, creating drops
  - Some drops hard to understand (e.g. crossing switch backplanes).
  - 100Mbit switch modules dropping multicasts, even when no other traffic.
  - Network drivers not registering reliably with NICs for multicast groups

Success Area: Use of UDP and Multicast

- UDP is connectionless and lighter weight than TCP.
- We use multicast for “control signals”: scales (up to a point) trivially. Maybe multicast the data in future? Registering for switch multicast groups was difficult to do robustly.
- Connectionless nature of UDP allowed use of switch port-spanning to “secretly” test new hardware in real data-taking situation.
- VxWorks network driver too slow. UDP allowed us to write a zero-copy Gbit ethernet driver more easily => big performance increase for $300 per board !!!
- Forced us to learn. Would do it again.

Success Area: Dynamic Linking



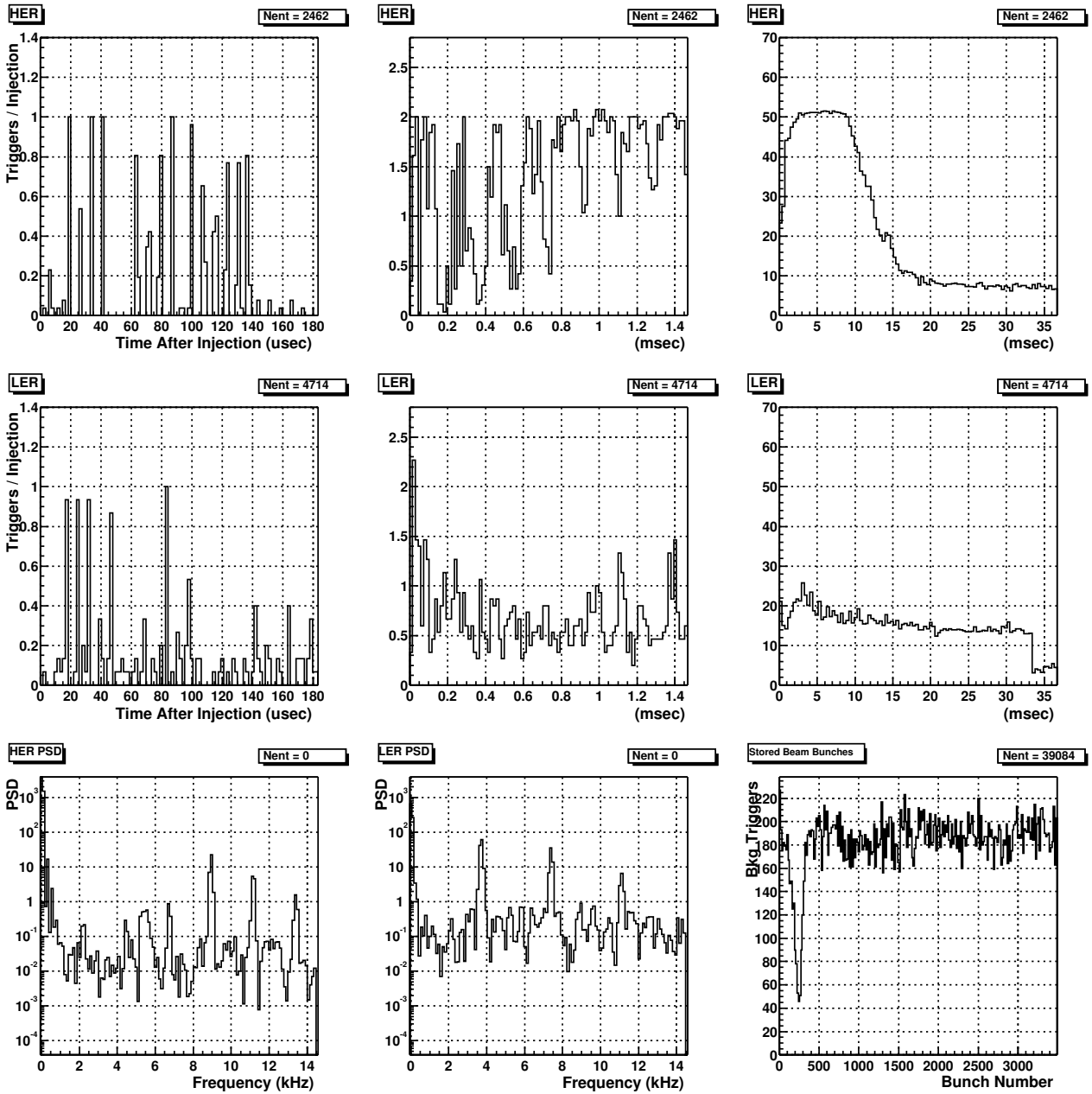
- Used in VxWorks. Enabled packages to change at their own rate.
- Compiling faster, easier to manage. especially when iteratively debugging.
- Installed code base smaller.
- Harder on UNIX (we almost did it): LD_LIBRARY_PATH and path hardwired in library difficult to control well.
- Dynamic linking on VxWorks was a joy.
- Would like everywhere, but would require physicists to manage interfaces/versioning better. Perhaps unrealistic, sadly?

Success Area: Realtime Info to Accelerator (I)

- Multicast 3 longwords on DAQ subnet every trigger (unaffected by deadtime!):
  - mask of trigger bits (1 longword)
  - timestamp tied to accelerator clock, 16ns resolution (2 longwords)
- Analyzed in various ways (including FFT) and presented to PEP in real-time (10Hz update rate). Can “see” individual bunches.
- They tune the accelerator using this, even when Babar is not taking data. Wealth of information (e.g. synchrotron oscillations).
- In future would do more: multicast finer granularity trigger information.

What we already send to PEP

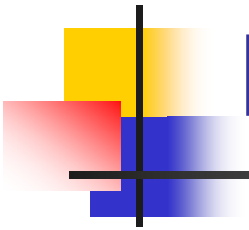
- L1 time distribution after injection (1M triggers)
- Fourier transform
- All real time data: PEP can "knob" on the histograms
- This information already proved a great success
  - see Mike Sullivan talk in trickle injection session



Success Area: Real-time Info to Accelerator (II)

- Only recently began delivering real-time beamspot (position, shape) information to accelerator with full Silicon tracking.
- Again a wealth of information (e.g. bunch-length measurements, beam-beam interaction effects).
- Should have done this sooner.

Success Area: Miscellaneous



- Event batching (to reduce effect of data transfer overhead). We retrofitted it, would have been nice to have it designed in more cleanly.
- Test stands and real system "look the same". Data taking is a "special case" of calibration.
- Tolerance of errors:
  - e.g. crashed "Level 3" node is detected and eliminated in seconds.
  - e.g. events can be "damaged" with almost no impact on DAQ performance.
  - Requires hard work to keep timeouts small