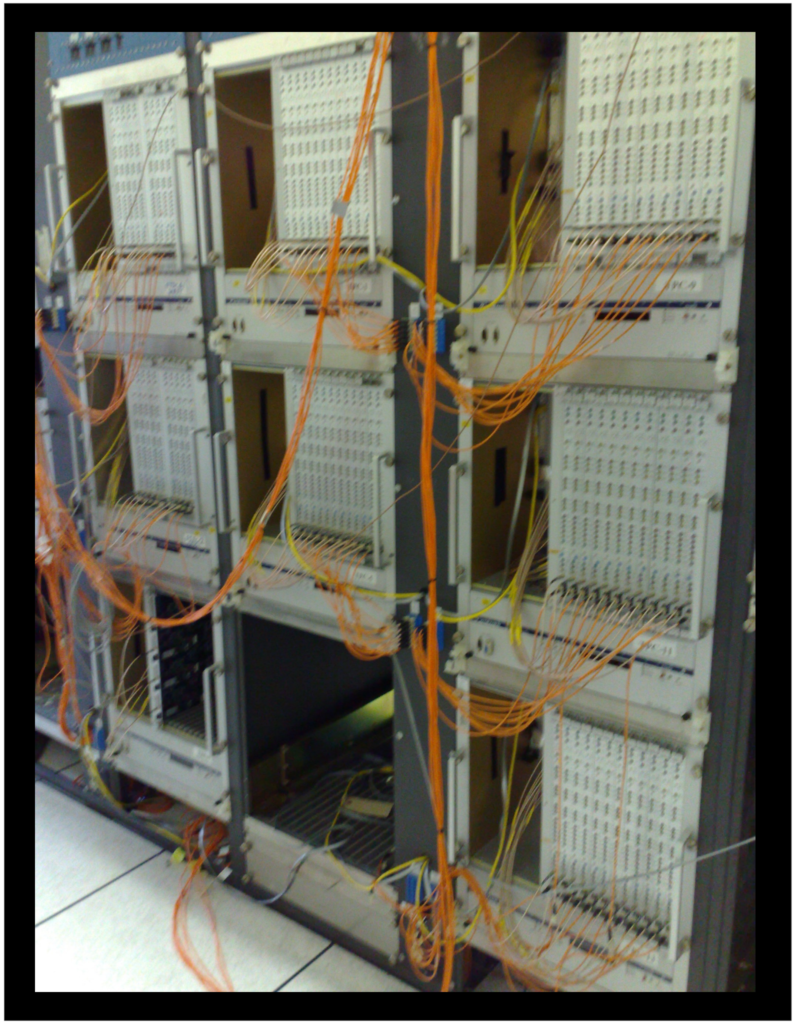
Overview and Evolution of the STAR DAQ & Trigger 2000 - 2013:



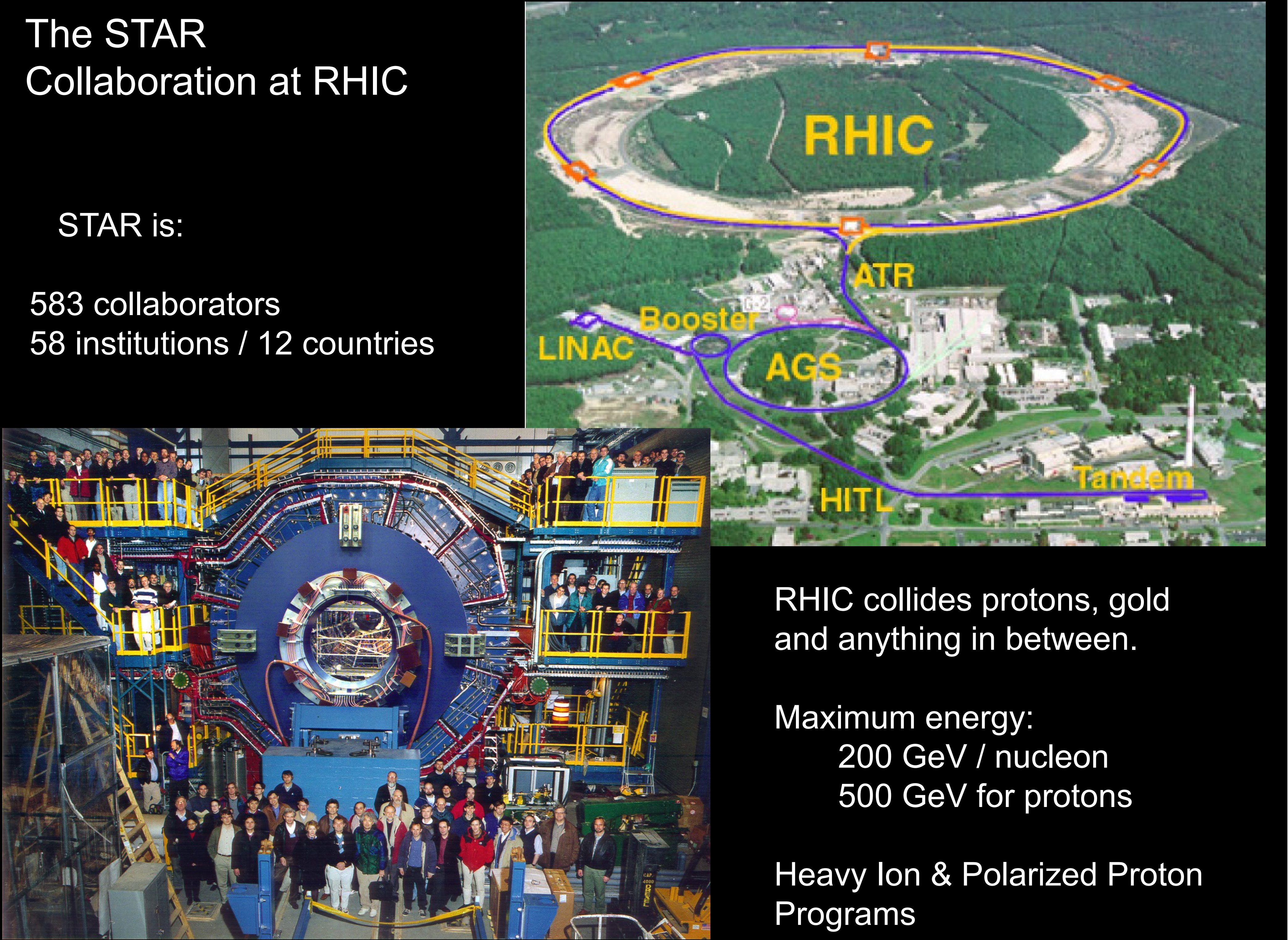
DAQ Receiver Crates
2000-2008

DAQ Receiver CPUs
2008-present

The STAR Collaboration at RHIC

STAR is:

583 collaborators
58 institutions / 12 countries



RHIC collides protons, gold and anything in between.

Maximum energy:
200 GeV / nucleon
500 GeV for protons

Heavy Ion & Polarized Proton Programs

This will be an overview of many projects, so of course a lot of people were involved
Some of the critical ones are listed here.

STAR DAQ Group (BNL):
- Jeff Landgraf
- Tonko Ljubicic
- (Bob Scheetz)

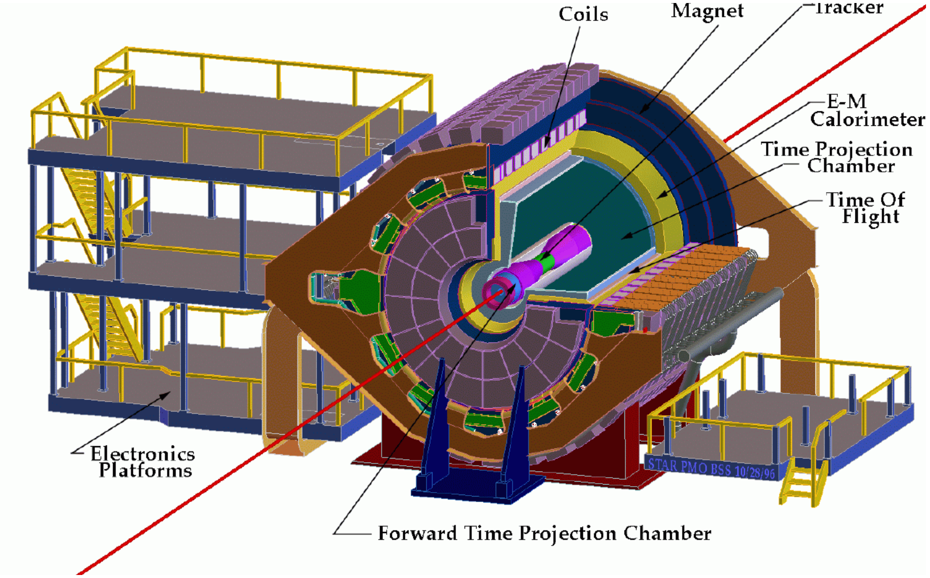
Various Front End Electronics:
- Gerard Visser
- Joe Schambach

STAR Trigger Group (LBL):
- Hank Crawford
- Eleanor Judd
- Jack Engelage
- Chris Perkins

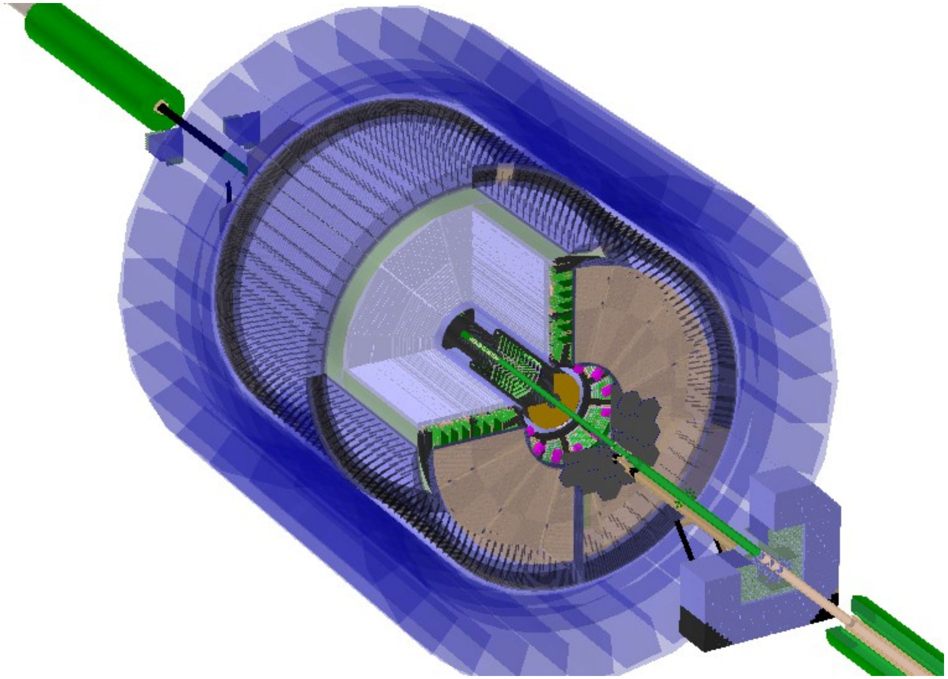
STAR Electronics Support Group (BNL):
- Bill Christie
- Bob Soja
- John Scheblein
- Alexei Lebedev
- John Hammond
- Phil Kuczewski
- Alex Tkatchev
- Tim Camarada

HLT Trigger:
- Aihong Tang (BNL)
- Hongwei Ke (CCNU & BNL)
- Yi Guo (USTC & BNL)
- Ivan Kisel (Frankfurt)
- Igor Kulakov (Frankfurt)
- Qiye Shou (SINAP & BNL)
- Zhangqiao Zhang (SINAP & BNL)
- Maksym Zyzak (Frankfurt)

The Evolution of STAR as a Whole



2000



2013

Detectors Removed:

- RICH
- SVT
- FTPC
- CTB
- PMD
- SSD

Detectors Added or Extended

- VPD
- TOF
- BBC
- FPD
- FMS
- EEMC
- BEMC
- ESMD
- BSMD
- SSD
- PXL
- FGT
- PMD
- MTD
- Roman Pots

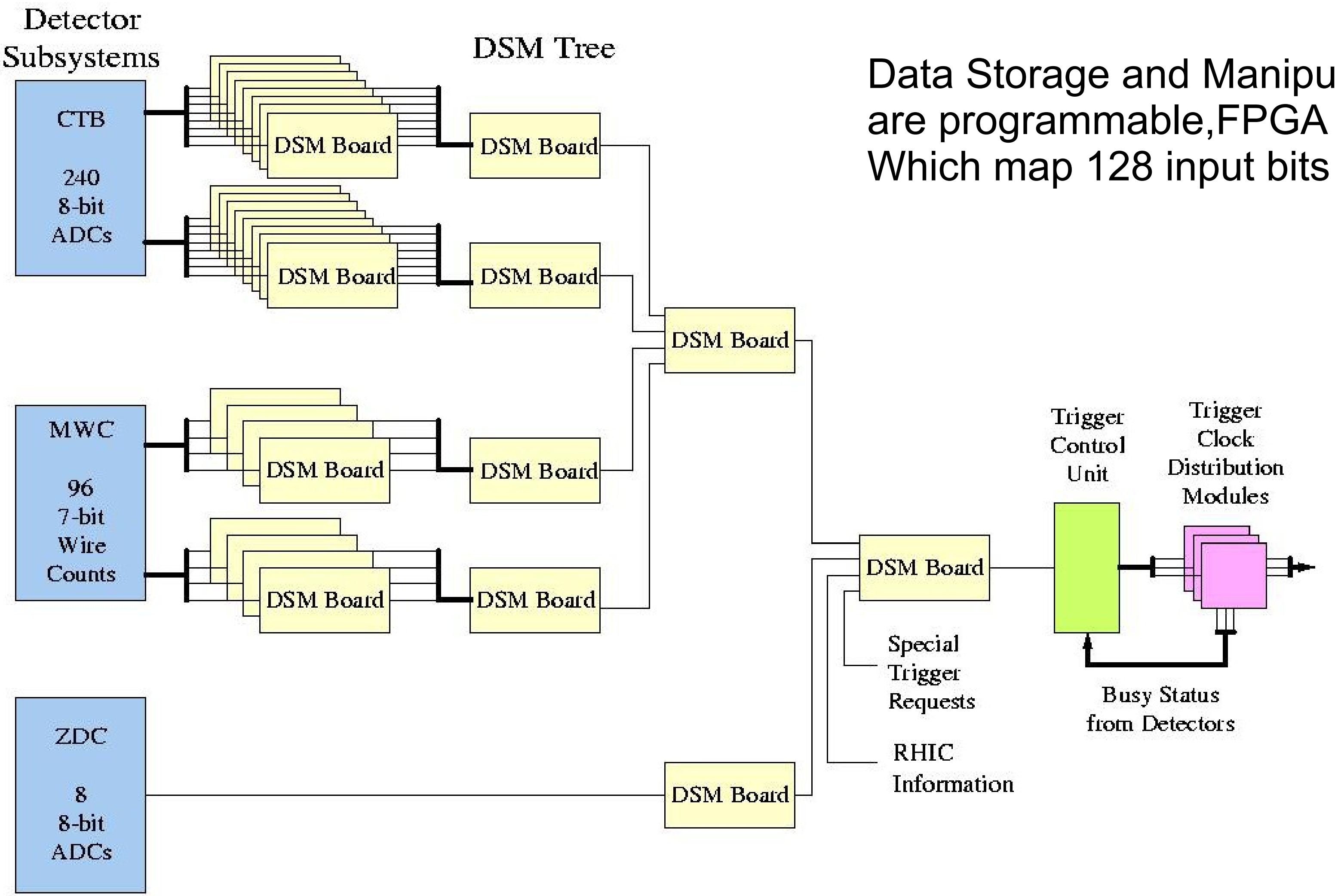
I'm going to give an overview of the STAR Trigger/DAQ as it was at the start of physics operations in 2000, and highlight some of the major upgrades through the present time.

Although the point of upgrades is always to extend the physics reach of the detector, I am going to largely ignore the direct detector upgrades and instead focus on the trigger/DAQ logic and design. In some cases the impact on physics is simple, such as a reduction in detector deadtime or an increase in available bandwidth or event rate. In other cases, the effect of the trigger/DAQ upgrades is far more profound.

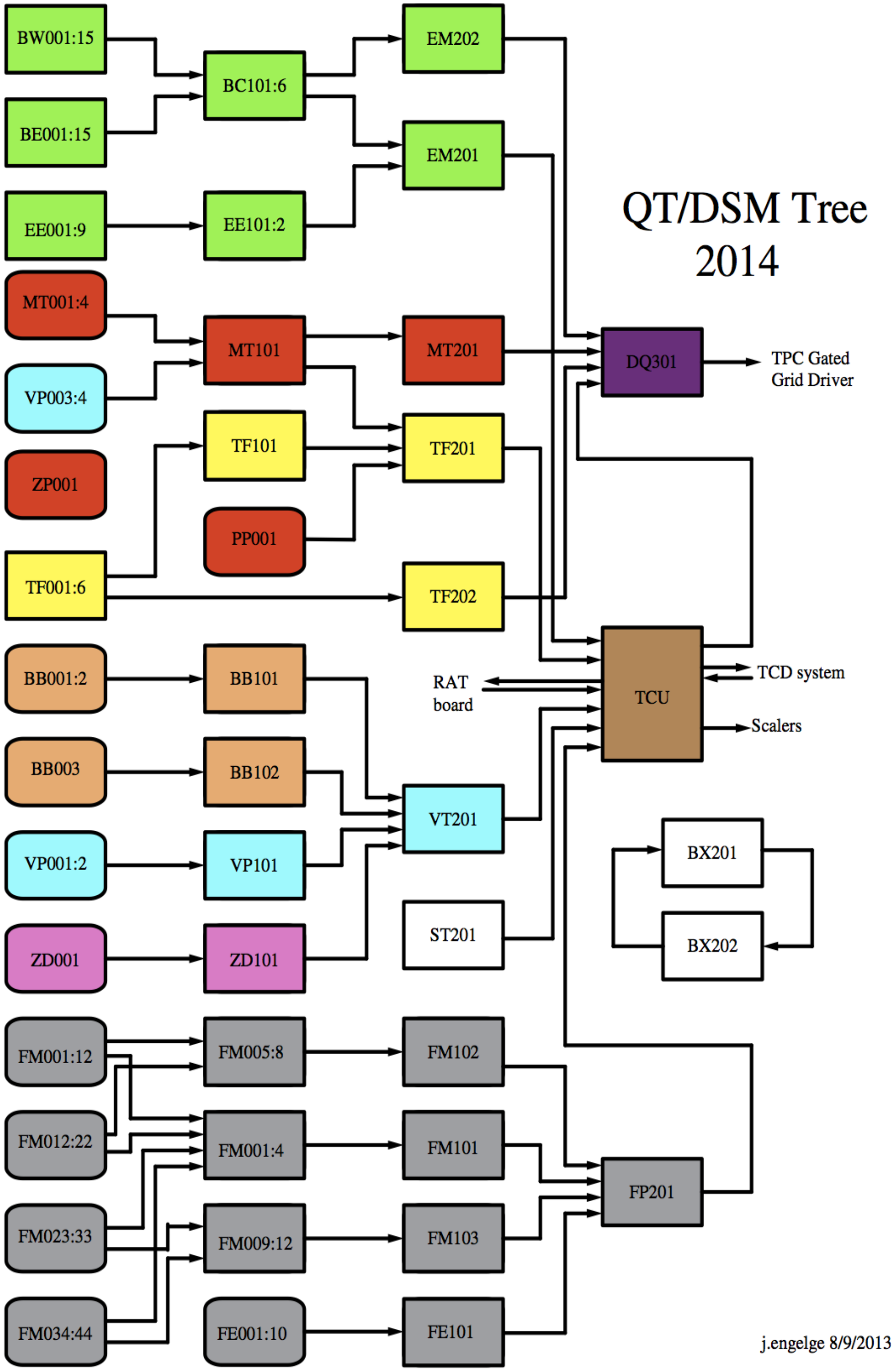
My strategy will be to go through various parts of the system in turn:

- The L0 trigger
- The DAQ system
- The High Level Triggers (L1, L2, L3 and L4 triggers)

L0 Trigger as of 2000:



Data Storage and Manipulation Boards are programmable,FPGA based boards Which map 128 input bits to 32 output bits

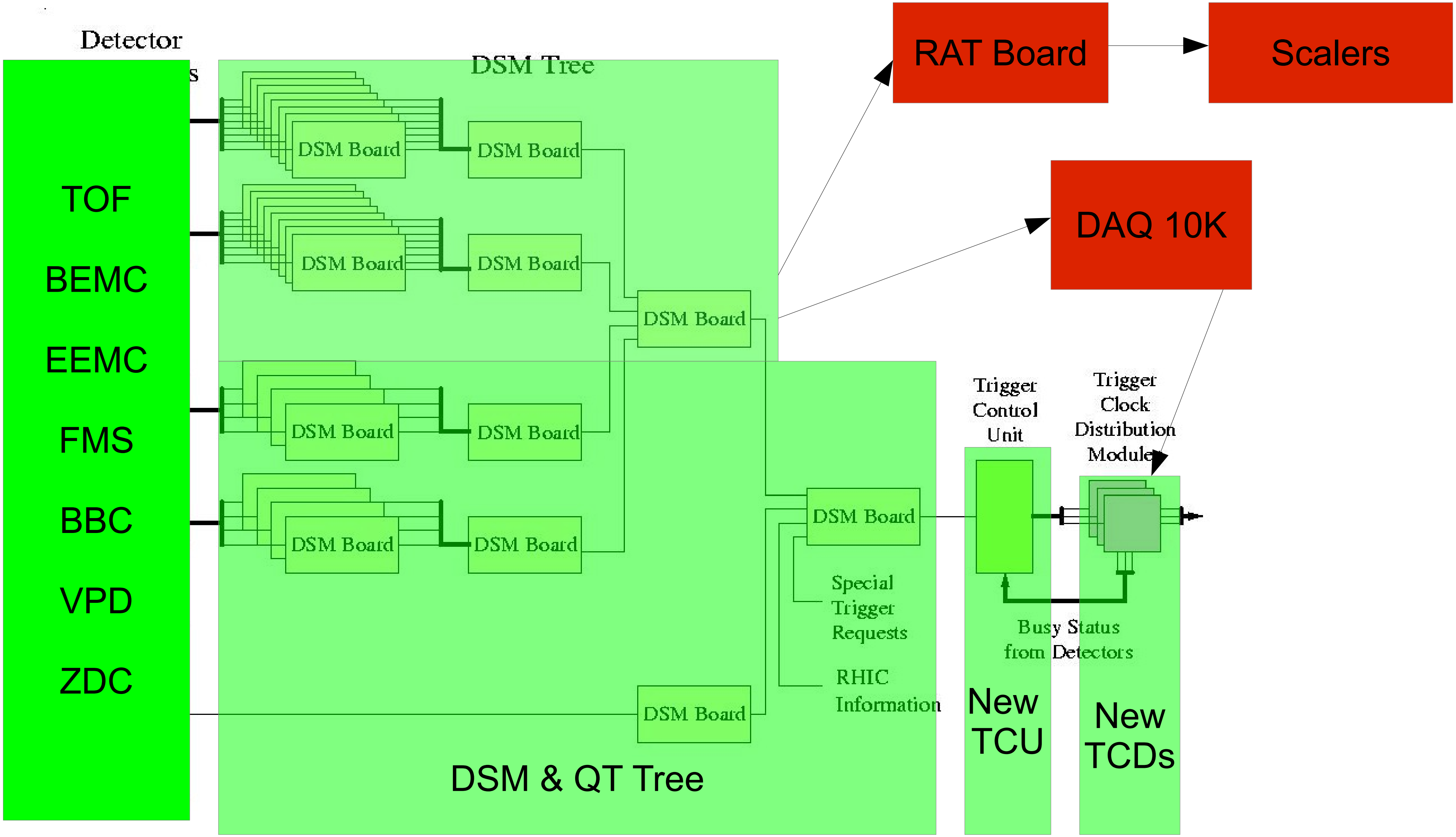


Everything has changed!

- * New/Different Detectors
- * More DSMs with more complex logic
- * QT boards created
  - \- Add digitization
  - \- Add time discrimination
- * Upgraded TCU (4 upgrades)
- * Upgraded TCD
- * Installed/expanded scaler systems
- * Installed RAT boards
- * Upgraded clock distribution
- * Switched/developed new network

Note that each of these DSM/QT boards have different programming.

Some new concepts, but the scheme has not changed!



Detectors

TOF

BEMC

EEMC

FMS

BBC

VPD

ZDC

DSM Tree

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

DSM Board

Special Trigger Requests

RHIC Information

DSM & QT Tree

RAT Board

Scalers

DAQ 10K

Trigger Control Unit

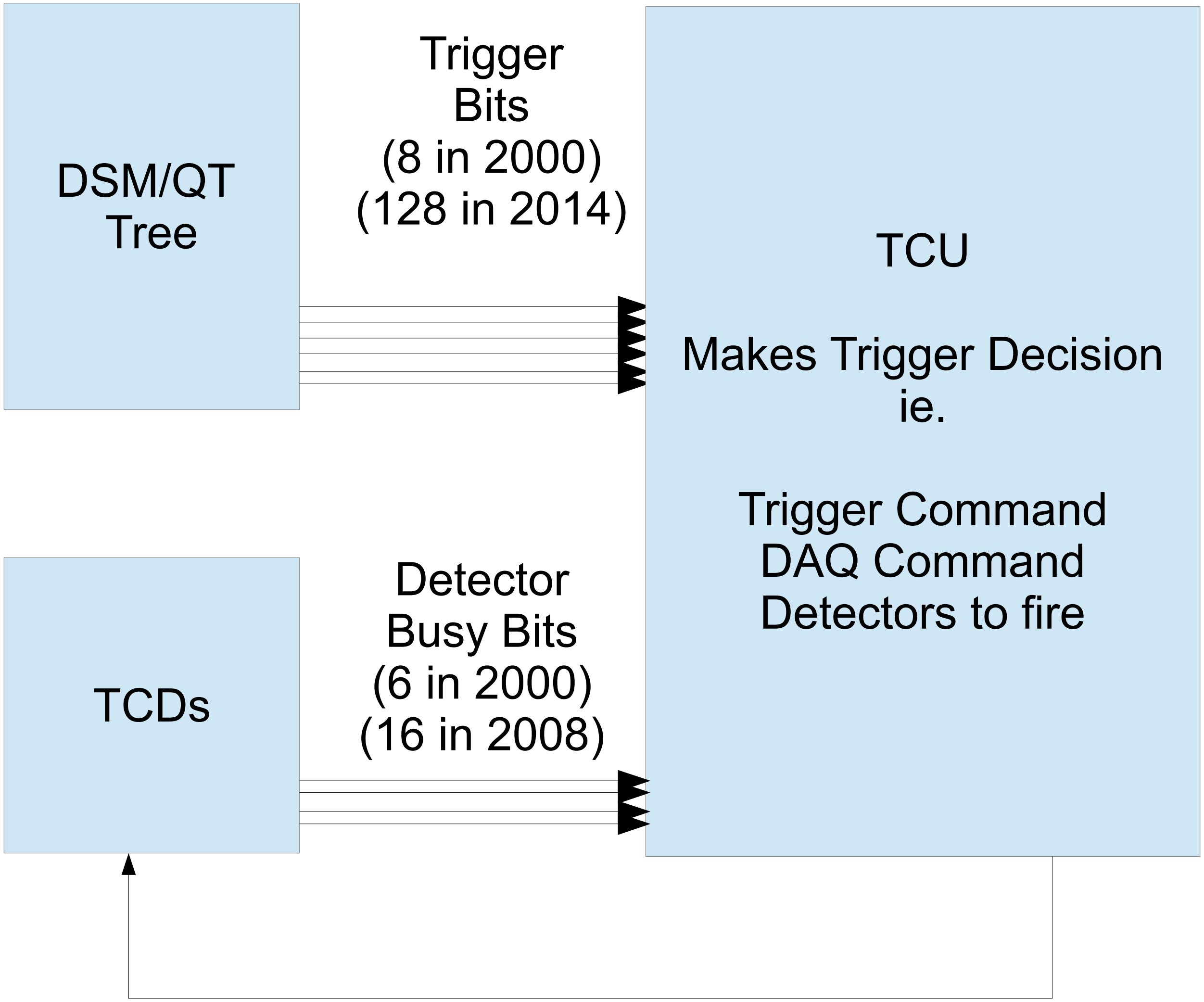
Trigger Clock Distribution Modules

Busy Status from Detectors

New TCU

New TCDs

What is this scheme? Making the trigger decision:



DSM/QT Tree

Trigger Bits
(8 in 2000)
(128 in 2014)

TCU

Makes Trigger Decision
ie.

Trigger Command
DAQ Command
Detectors to fire

TCDs

Detector Busy Bits
(6 in 2000)
(16 in 2008)

Simplifying Feature:

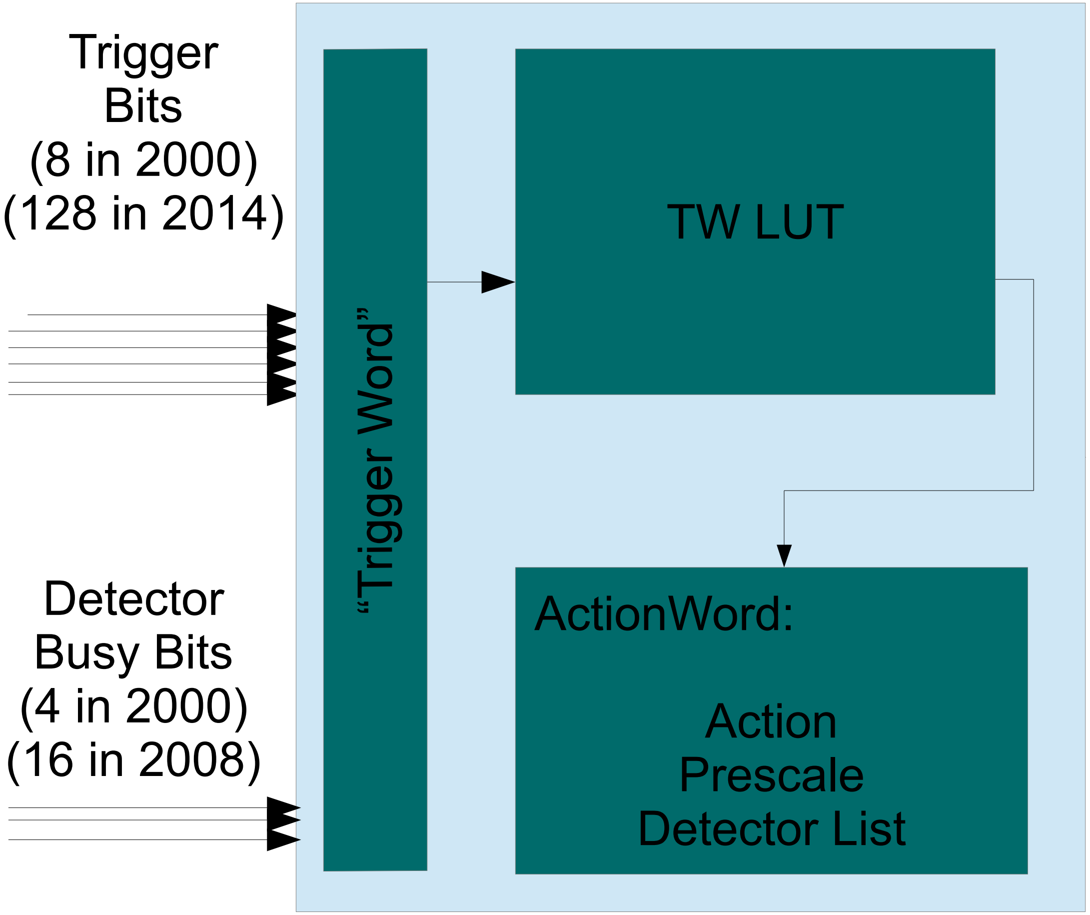
* Despite the complexity of the L0 electronics & detectors the only information the trigger depends on is A very small number of bits

Challenges:

* The meaning of these bits is not static.   It can change run to run..

* Any change, in the programming of any DSM has the potential to change the meaning of the trigger bits.

2000 – 2008 TCU Logic:



Formation of the Trigger:

1. The trigger and detector busy bits are treated as an integer called the Trigger Word
2. The TW is used as an index into a lookup table
3. This lookup table entry Describes the action to be Taken including:
    - Trigger decision
    - Prescales
    - Detectors to fire
    - physics/laser/pulser
    - special DAQ handling

Features:

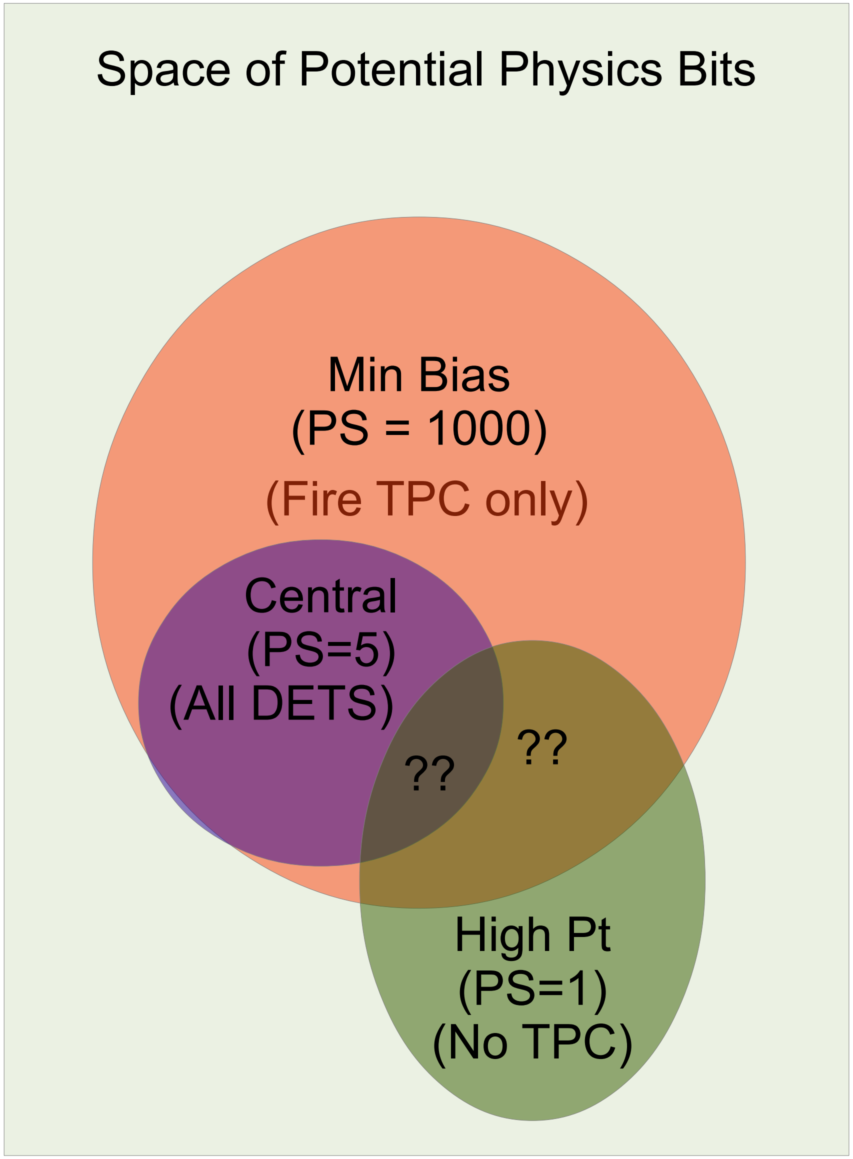
* Multiple Triggers possible
(up to 64K with 16 bits?)
* Very flexible:
"Infinitely programmable"

Early Challenges!

* The DSM programming changed run to run
  - Different versions of the DSM FPGA code for each implementation of trigger
  - Different versions of the DSM FPGA code even for threshold changes
  - Difficult to track meaning of input bits
  - Difficult to track history of trigger definitions

* How to program the TCU?
  - For 2000-2001 the TCU LUT was programmed by writing a "C" program to produce the table for each configuration change.
  - Difficult & Obtuse & Error prone
  - Untrackably dependent on DSM tree

* How to implement multiple triggers?
  - The Trigger group considered triggers to be Defined by the "trigger word" But the trigger word did not map well to physics triggers.
  - What to do with trigger overlap regions?



The early attempts at running with multiple triggers caused chaos. Despite the potential for STAR to be able to multiple triggers simultaneously, STAR was a single trigger detector.

The Solution was the “Run Control”

1. The Java User Interface:
   - One click shift crew operation
   - Easy interface to control and view DAQ/trigger component status
   - Editor to define trigger configurations

2. The Run Control “Handler”

3. The TIER1 file
   - An interface for “Releases” of DSM code. We made the agreement that DSM code would always be released as bundles, and this was enforced by using those bundles in the configuration code of the trigger nodes.
     – The compiled DSM codes for all components
     – Dictionaries for TCU Bits, DSM registers, & L1/L2/L3 algorithms

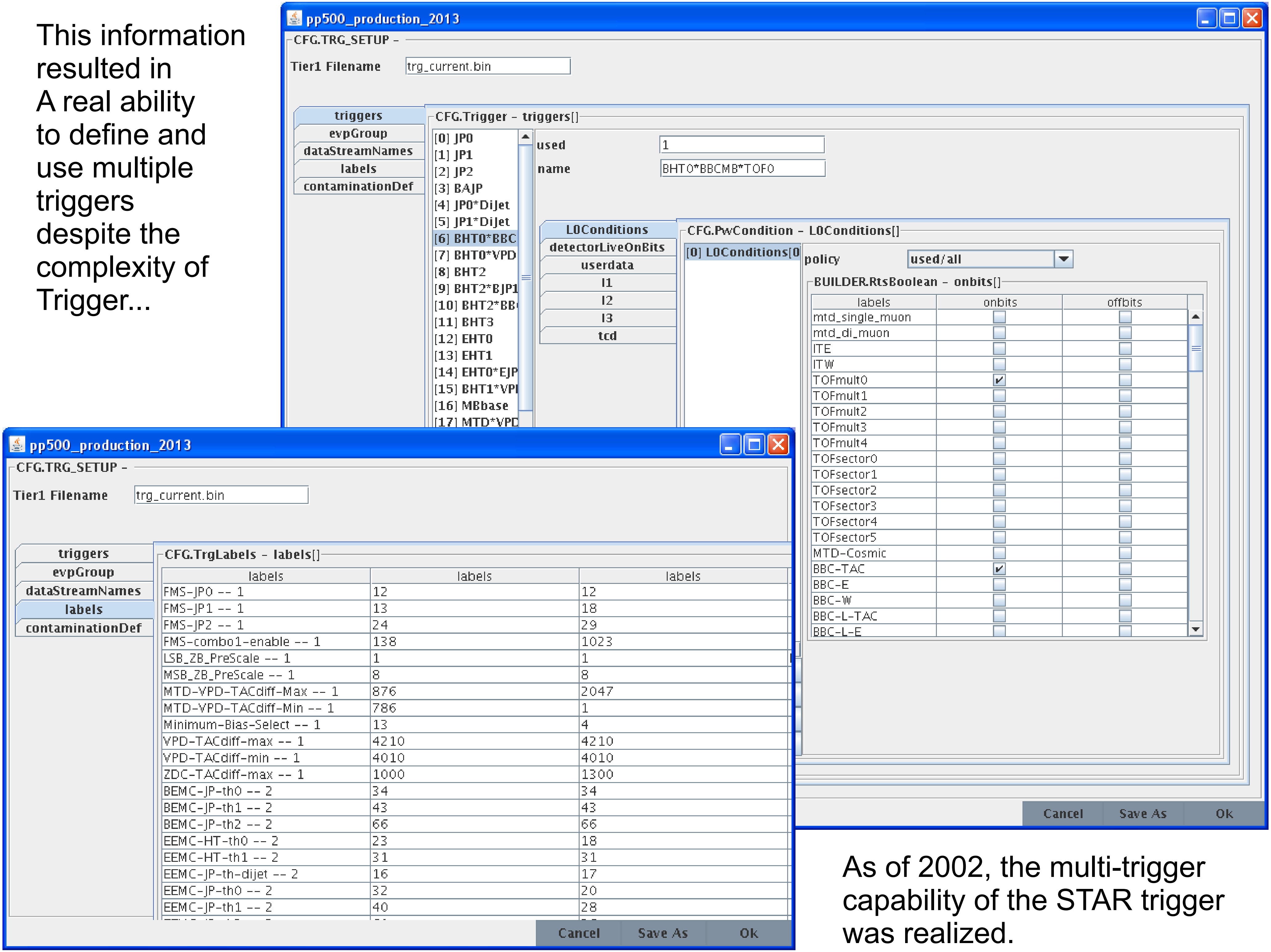
4. A simple way of defining trigger conditions based on masks of the TCU input bits and Detector busy bits

5. Code to create the TCU lookup tables and deal with overlap regions
   - Use the minimum prescale for overlapping triggers
   - Use the maximal detector set for overlapping triggers
   - Flag errors if incompatible actions requested

6. Rescaling code for the L1 CPU that retroactively applies the remaining part of the prescale for each overlapping trigger and marks the event.

7. DB's for tracking the dictionaries and trigger definitions for each run

This information resulted in
A real ability to define and use multiple triggers despite the complexity of Trigger...



CFG.TRG_SETUP –

Tier1 Filename: trg_current.bin

CFG.Trigger – triggers[]

- [0] JP0
- [1] JP1
- [2] JP2
- [3] BAJP
- [4] JP0*DiJet
- [5] JP1*DiJet
- [6] BHT0*BBC
- [7] BHT0*VPD
- [8] BHT2
- [9] BHT2*BJP1
- [10] BHT2*BB
- [11] BHT3
- [12] EHT0
- [13] EHT1
- [14] EHT0*EJP
- [15] BHT1*VP
- [16] MBbase
- [17] MTD*VPD

used: 1
name: BHT0*BBCMB*TOF0

CFG.PwCondition – L0Conditions[]

[0] L0Conditions[0

policy: used/all

BUILDER.RtsBoolean – onbits[]

labels	onbits	offbits
mtd_single_muon	☐	☐
mtd_di_muon	☐	☐
ITE	☐	☐
ITW	☐	☐
TOFmult0	☑	☐
TOFmult1	☐	☐
TOFmult2	☐	☐
TOFmult3	☐	☐
TOFmult4	☐	☐
TOFsector0	☐	☐
TOFsector1	☐	☐
TOFsector2	☐	☐
TOFsector3	☐	☐
TOFsector4	☐	☐
TOFsector5	☐	☐
MTD-Cosmic	☐	☐
BBC-TAC	☑	☐
BBC-E	☐	☐
BBC-W	☐	☐
BBC-L-TAC	☐	☐
BBC-L-E	☐	☐

CFG.TRG_SETUP –

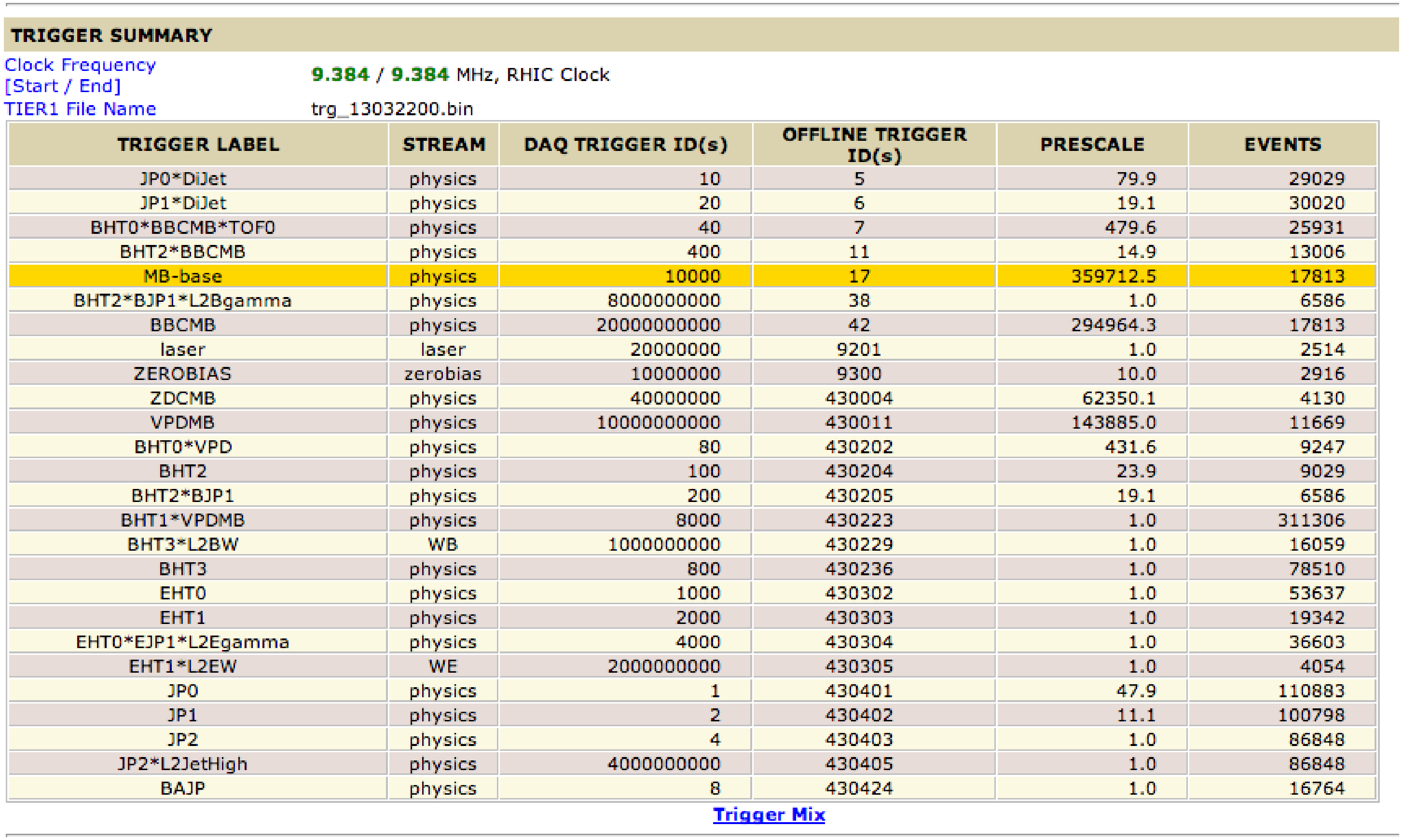
Tier1 Filename: trg_current.bin

CFG.TrgLabels – labels[]

labels	labels	labels
FMS-JP0 -- 1	12	12
FMS-JP1 -- 1	13	18
FMS-JP2 -- 1	24	29
FMS-combo1-enable -- 1	138	1023
LSB_ZB_PreScale -- 1	1	1
MSB_ZB_PreScale -- 1	8	8
MTD-VPD-TACdiff-Max -- 1	876	2047
MTD-VPD-TACdiff-Min -- 1	786	1
Minimum-Bias-Select -- 1	13	4
VPD-TACdiff-max -- 1	4210	4210
VPD-TACdiff-min -- 1	4010	4010
ZDC-TACdiff-max -- 1	1000	1300
BEMC-JP-th0 -- 2	34	34
BEMC-JP-th1 -- 2	43	43
BEMC-JP-th2 -- 2	66	66
EEMC-HT-th0 -- 2	23	18
EEMC-HT-th1 -- 2	31	31
EEMC-JP-th-dijet -- 2	16	17
EEMC-JP-th0 -- 2	32	20
EEMC-JP-th1 -- 2	40	28

As of 2002, the multi-trigger capability of the STAR trigger was realized.

Run Log:



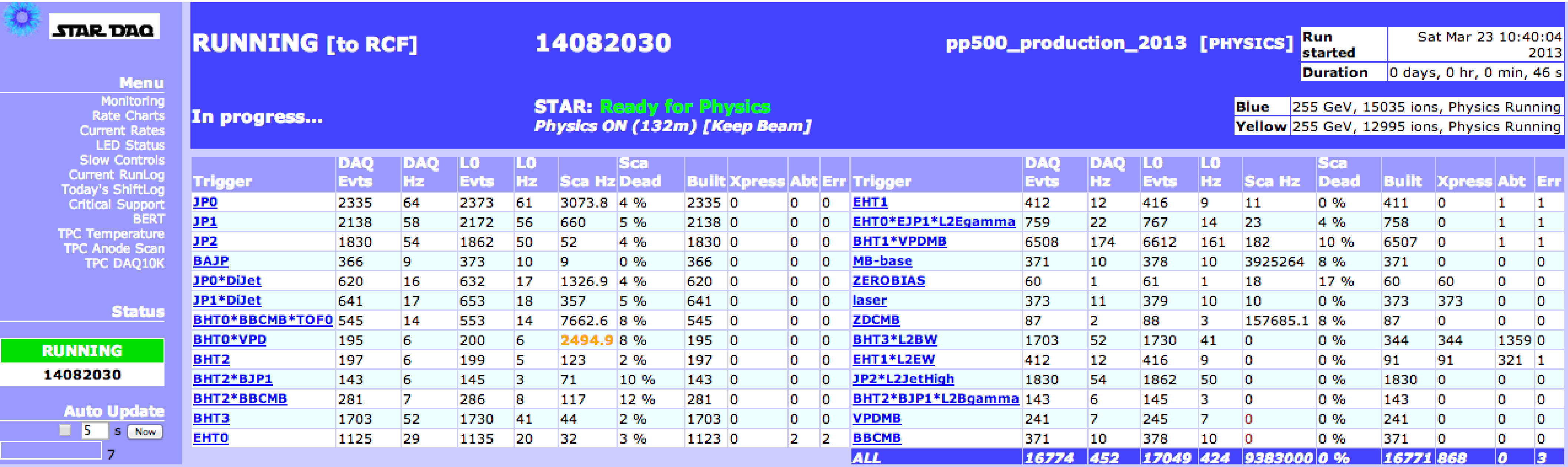
TRIGGER SUMMARY

Clock Frequency [Start / End]	**9.384** / **9.384** MHz, RHIC Clock
TIER1 File Name	trg_13032200.bin

TRIGGER LABEL	STREAM	DAQ TRIGGER ID(s)	OFFLINE TRIGGER ID(s)	PRESCALE	EVENTS
JP0*DiJet	physics	10	5	79.9	29029
JP1*DiJet	physics	20	6	19.1	30020
BHT0*BBCMB*TOF0	physics	40	7	479.6	25931
BHT2*BBCMB	physics	400	11	14.9	13006
MB-base	physics	10000	17	359712.5	17813
BHT2*BJP1*L2Bgamma	physics	8000000000	38	1.0	6586
BBCMB	physics	20000000000	42	294964.3	17813
laser	laser	20000000	9201	1.0	2514
ZEROBIAS	zerobias	10000000	9300	10.0	2916
ZDCMB	physics	40000000	430004	62350.1	4130
VPDMB	physics	10000000000	430011	143885.0	11669
BHT0*VPD	physics	80	430202	431.6	9247
BHT2	physics	100	430204	23.9	9029
BHT2*BJP1	physics	200	430205	19.1	6586
BHT1*VPDMB	physics	8000	430223	1.0	311306
BHT3*L2BW	WB	1000000000	430229	1.0	16059
BHT3	physics	800	430236	1.0	78510
EHT0	physics	1000	430302	1.0	53637
EHT1	physics	2000	430303	1.0	19342
EHT0*EJP1*L2Egamma	physics	4000	430304	1.0	36603
EHT1*L2EW	WE	2000000000	430305	1.0	4054
JP0	physics	1	430401	47.9	110883
JP1	physics	2	430402	11.1	100798
JP2	physics	4	430403	1.0	86848
JP2*L2JetHigh	physics	4000000000	430405	1.0	86848
BAJP	physics	8	430424	1.0	16764

Trigger Mix

DAQ Realtime Monitoring:



RUNNING [to RCF] **14082030** **pp500_production_2013 [PHYSICS]**

Run started	Sat Mar 23 10:40:04 2013
Duration	0 days, 0 hr, 0 min, 46 s

In progress...

STAR: Ready for Physics
Physics ON (132m) [Keep Beam]

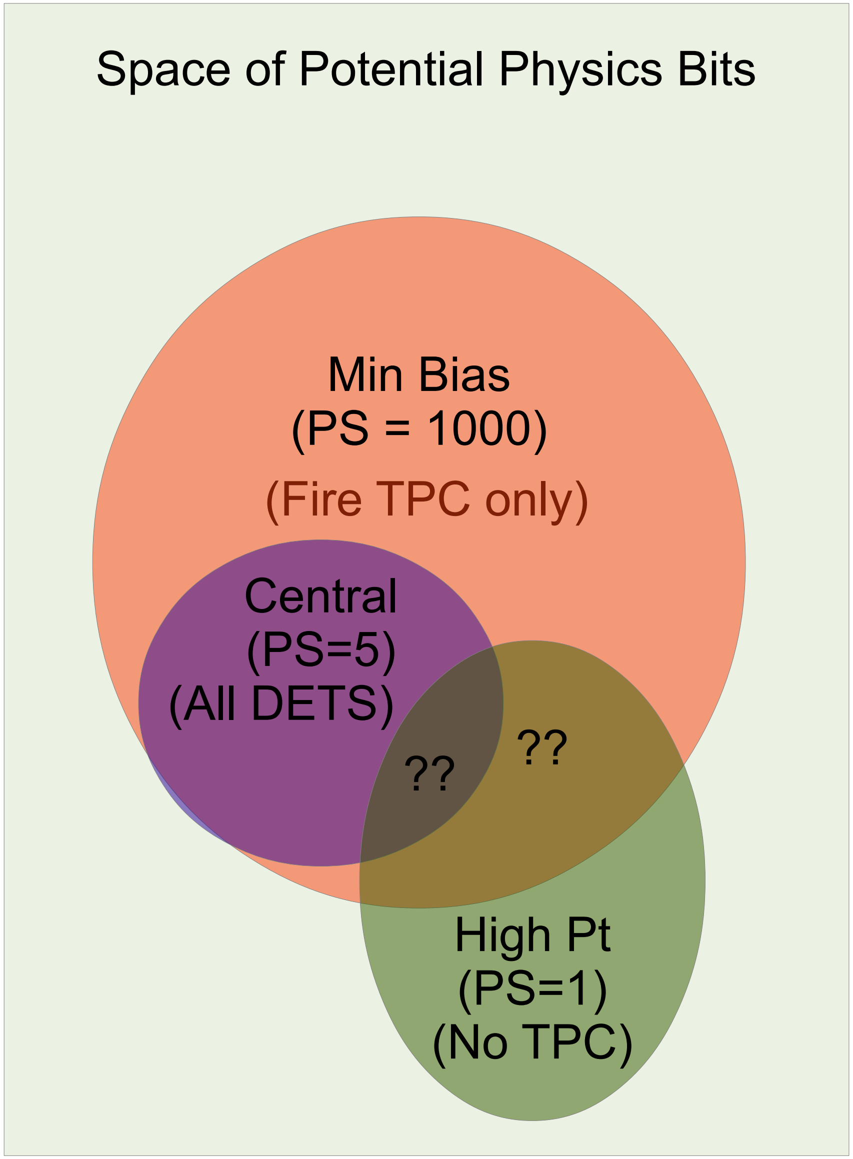
Blue	255 GeV, 15035 ions, Physics Running
Yellow	255 GeV, 12995 ions, Physics Running

Trigger	DAQ Evts	DAQ Hz	L0 Evts	L0 Hz	Sca Hz	Sca Dead	Built	Xpress	Abt	Err
JP0	2335	64	2373	61	3073.8	4 %	2335	0	0	0
JP1	2138	58	2172	56	660	5 %	2138	0	0	0
JP2	1830	54	1862	50	52	4 %	1830	0	0	0
BAJP	366	9	373	10	9	0 %	366	0	0	0
JP0*DiJet	620	16	632	17	1326.9	4 %	620	0	0	0
JP1*DiJet	641	17	653	18	357	5 %	641	0	0	0
BHT0*BBCMB*TOF0	545	14	553	14	7662.6	8 %	545	0	0	0
BHT0*VPD	195	6	200	6	2494.9	8 %	195	0	0	0
BHT2	197	6	199	5	123	2 %	197	0	0	0
BHT2*BJP1	143	6	145	3	71	10 %	143	0	0	0
BHT2*BBCMB	281	7	286	8	117	12 %	281	0	0	0
BHT3	1703	52	1730	41	44	2 %	1703	0	0	0
EHT0	1125	29	1135	20	32	3 %	1123	0	2	2

Trigger	DAQ Evts	DAQ Hz	L0 Evts	L0 Hz	Sca Hz	Sca Dead	Built	Xpress	Abt	Err
EHT1	412	12	416	9	11	0 %	411	0	1	1
EHT0*EJP1*L2Egamma	759	22	767	14	23	4 %	758	0	1	1
BHT1*VPDMB	6508	174	6612	161	182	10 %	6507	0	1	1
MB-base	371	10	378	10	3925264	8 %	371	0	0	0
ZEROBIAS	60	1	61	1	18	17 %	60	60	0	0
laser	373	11	379	10	10	0 %	373	373	0	0
ZDCMB	87	2	88	3	157685.1	8 %	87	0	0	0
BHT3*L2BW	1703	52	1730	41	0	0 %	344	344	1359	0
EHT1*L2EW	412	12	416	9	0	0 %	91	91	321	1
JP2*L2JetHigh	1830	54	1862	50	0	0 %	1830	0	0	0
BHT2*BJP1*L2Bgamma	143	6	145	3	0	0 %	143	0	0	0
VPDMB	241	7	245	7	0	0 %	241	0	0	0
BBCMB	371	10	378	10	0	0 %	371	0	0	0
ALL	***16774***	***452***	***17049***	***424***	***9383000***	***0 %***	***16771***	***868***	***0***	***3***

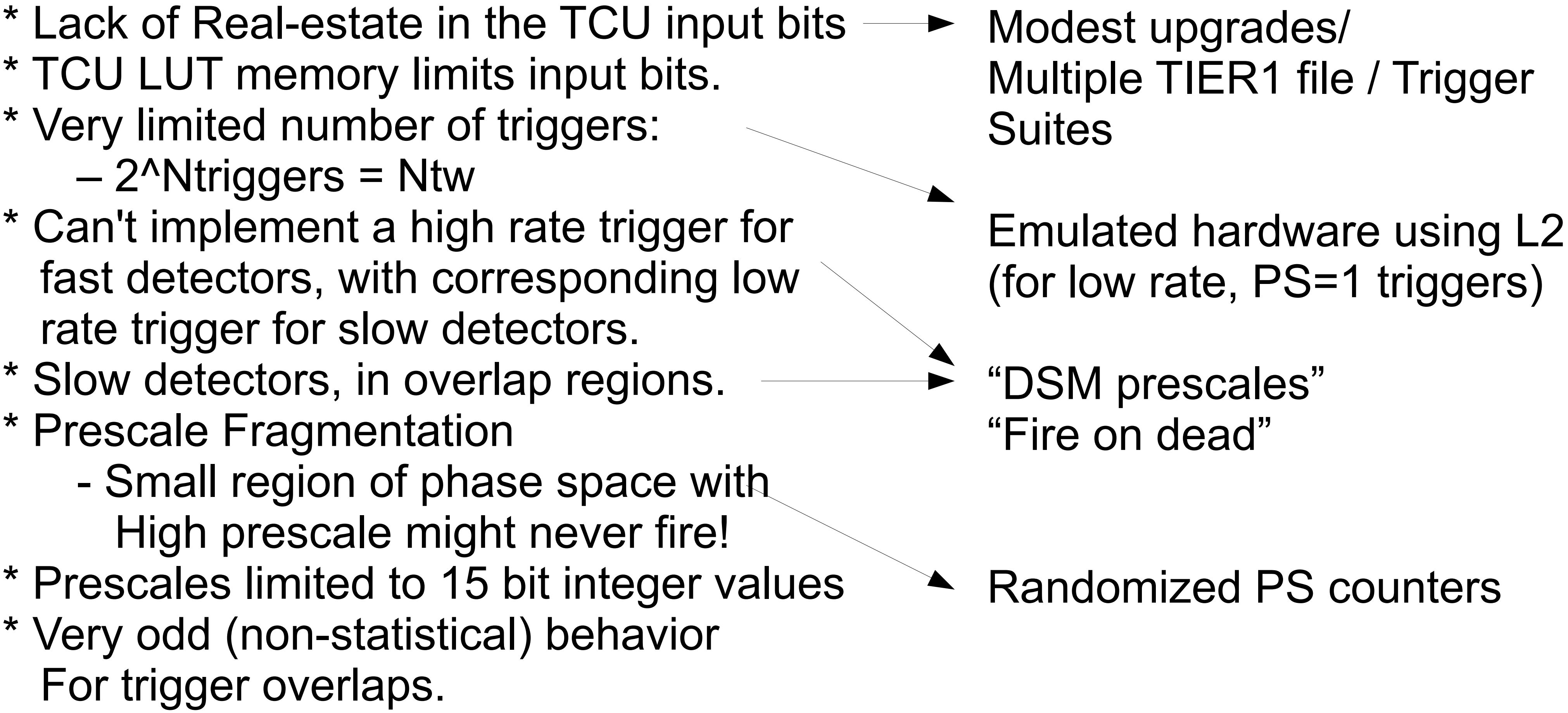
Remaining Challenges Related To the TCU!

* Lack of Real-estate in the TCU input bits
* TCU LUT memory limits input bits.
* Very limited number of triggers:
  – 2^Ntriggers = Ntw
* Can't implement a high rate trigger for fast detectors, with corresponding low rate trigger for slow detectors.
* Slow detectors, in overlap regions..
* Prescale Fragmentation
  - Small region of phase space with High prescale might never fire!
* Prescales limited to 15 bit integer values
* Very odd (non statistical) behavior For trigger overlaps



Remaining Challenges Related To the TCU!

Creative solutions!



- Lack of Real-estate in the TCU input bits
- TCU LUT memory limits input bits.
- Very limited number of triggers:
  – 2^Ntriggers = Ntw
- Can't implement a high rate trigger for fast detectors, with corresponding low rate trigger for slow detectors.
- Slow detectors, in overlap regions.
- Prescale Fragmentation
  - Small region of phase space with High prescale might never fire!
- Prescales limited to 15 bit integer values
- Very odd (non-statistical) behavior For trigger overlaps.

Modest upgrades/
Multiple TIER1 file / Trigger Suites

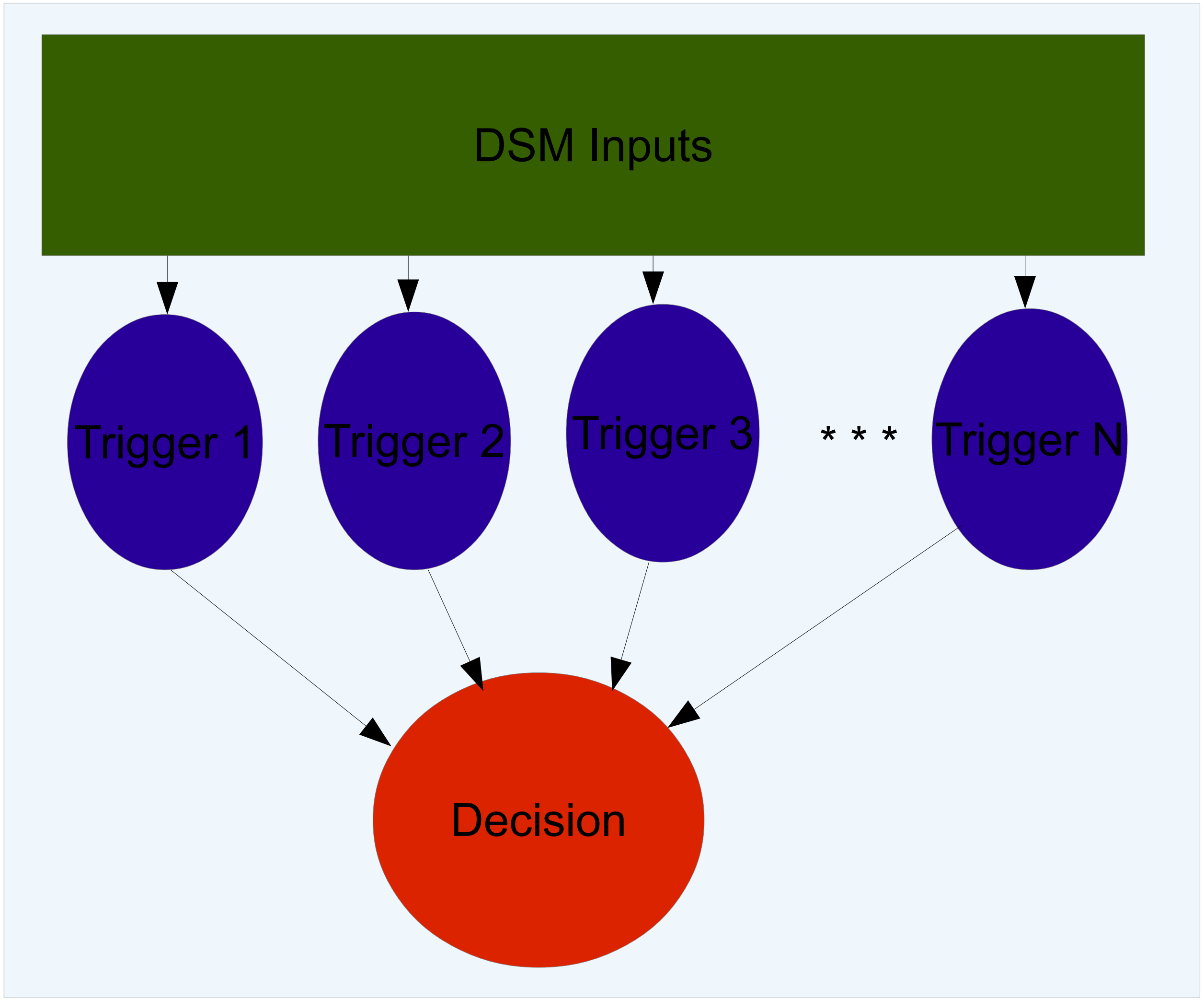
Emulated hardware using L2
(for low rate, PS=1 triggers)

“DSM prescales”
“Fire on dead”

Randomized PS counters

From 2002-2008 we were successful in running a very complex Physics program, but only using complex hacks!

The 2009 TCU:



For 32 independent triggers:

1. The DSM input bit masks are applied.
2. The detector busy requests are applied.
3. The prescale is applied

The event is fired if any trigger has been satisfied. The detector request is the logical “or” of the request from all Triggers.

Additionally, counters are kept for each trigger, tracking the number of times the physics was satisfied and the number Of times the event was rejected due to detector busy.

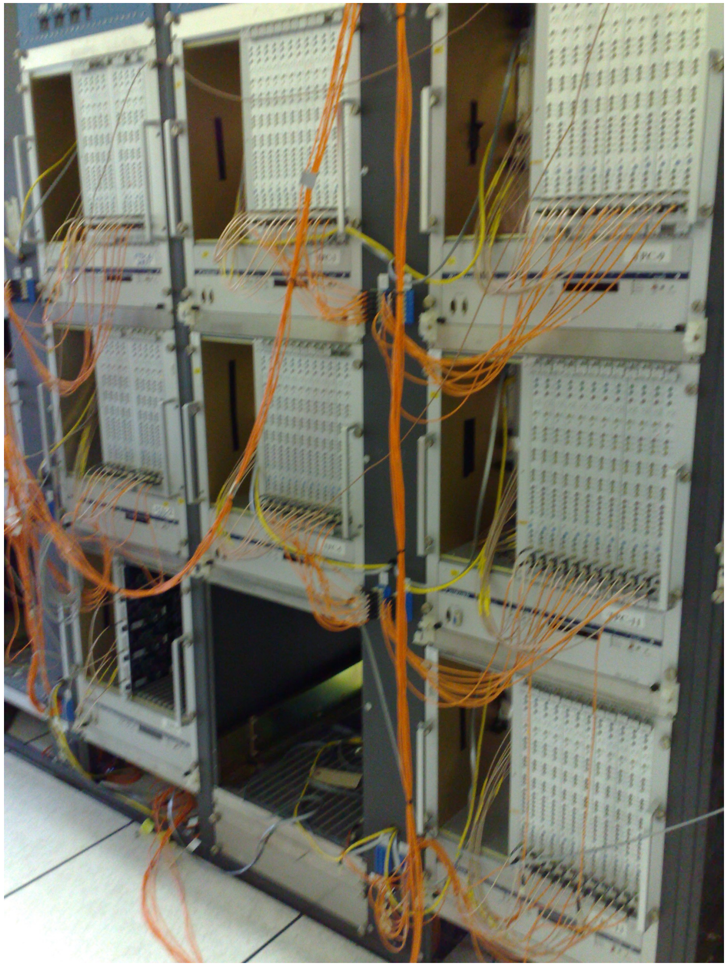
Trigger Summary:

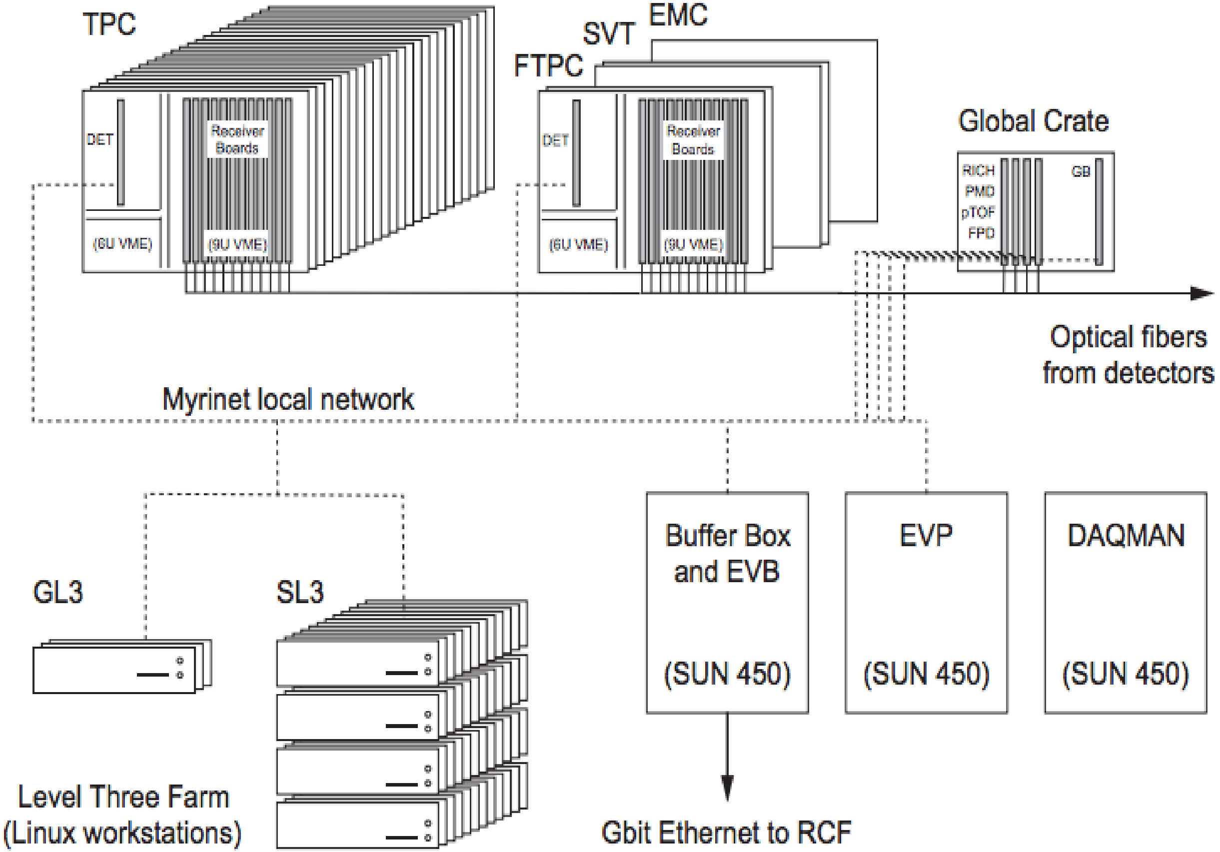
In 2013 we installed the final (I hope) upgraded TCU.

* 128 Input bits from the DSM tree
* Up to 64 completely independent parallel triggers
* Floating point prescales capable of scaling the full RHIC clock.
* Counters for each trigger, scaling the physics rate and deadtime

The result is that STAR is able to set up in a single configuration file the entire physics program for each run period, which allows our shift crew's to take data very efficiently.

DAQ (2000-2007)





Architecture:

- Gigalink fibers from detectors to RB
- VME Receiver Boards
- Myrinet Event Building Network
- Solaris 450 Event Builder
- Linux L3 trigger farm

Performance Specifications:

- 100Hz Maximum rate into DAQ (@ 100% Dead)
- 1Hz design event building (6-8hz)
- 20-30 MB/sec Event Building
- 10 MB/sec to RCF

2000-2007 Sector Crates:

MVME
CPU
(2606)

Myrinet
Interface
(to EVBs)

Busy Distribution Board

12 Receiver Boards

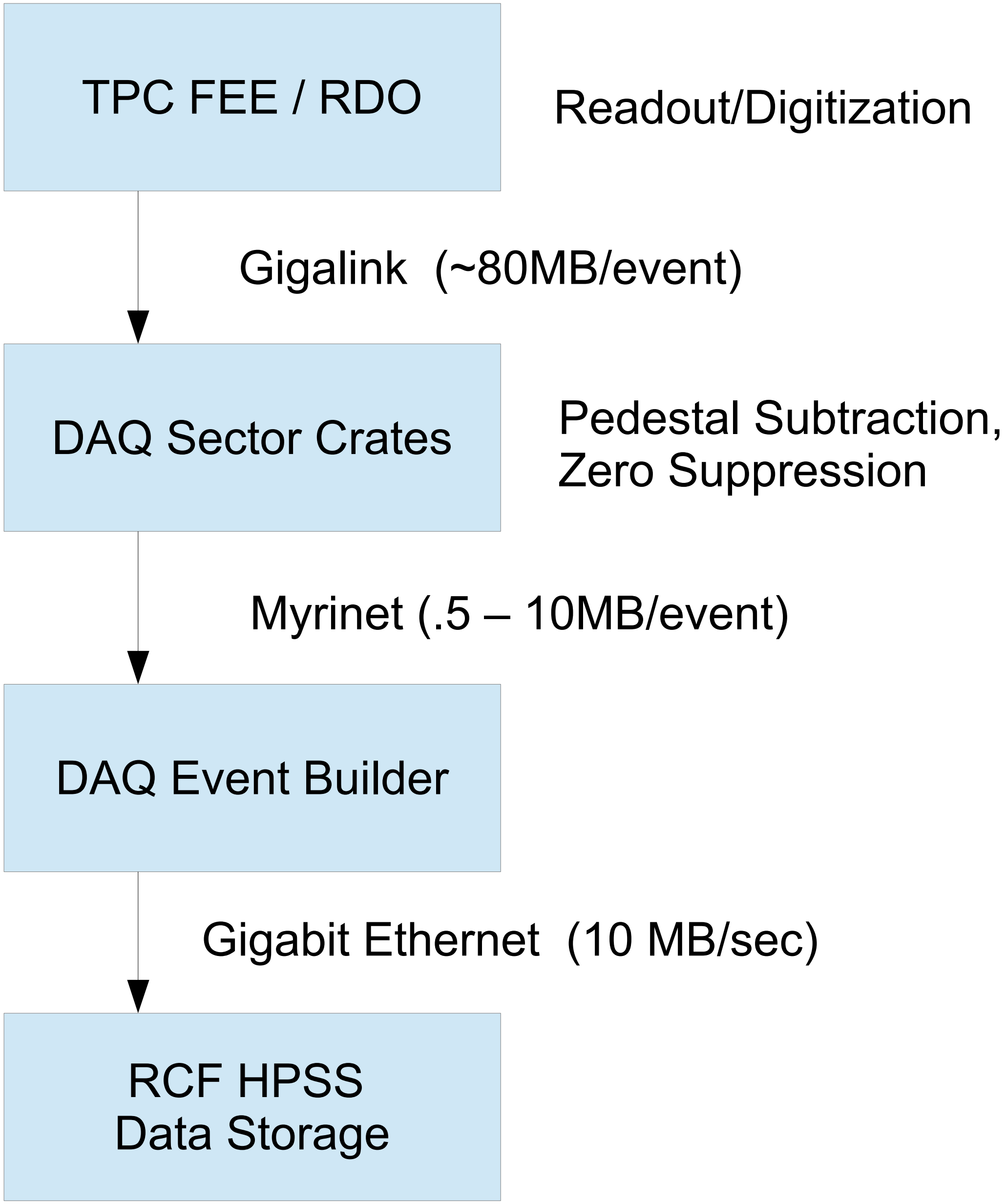
PMC Interface
(to Mezzanine's)

Fiber Interface
(to detectors)

VME Interface

The Mezzanine cards did most of the processing using 6 STAR ASIC's and an i960 CPU
- * pedestal subtraction
- * 1-D Hit finding (In the time direction)
- * zero suppression

DAQ Processing (2000-2002)



TPC FEE / RDO

Readout/Digitization

Gigalink (~80MB/event)

DAQ Sector Crates

Pedestal Subtraction,
Zero Suppression

Myrinet (.5 – 10MB/event)

DAQ Event Builder

Gigabit Ethernet (10 MB/sec)

RCF HPSS
Data Storage

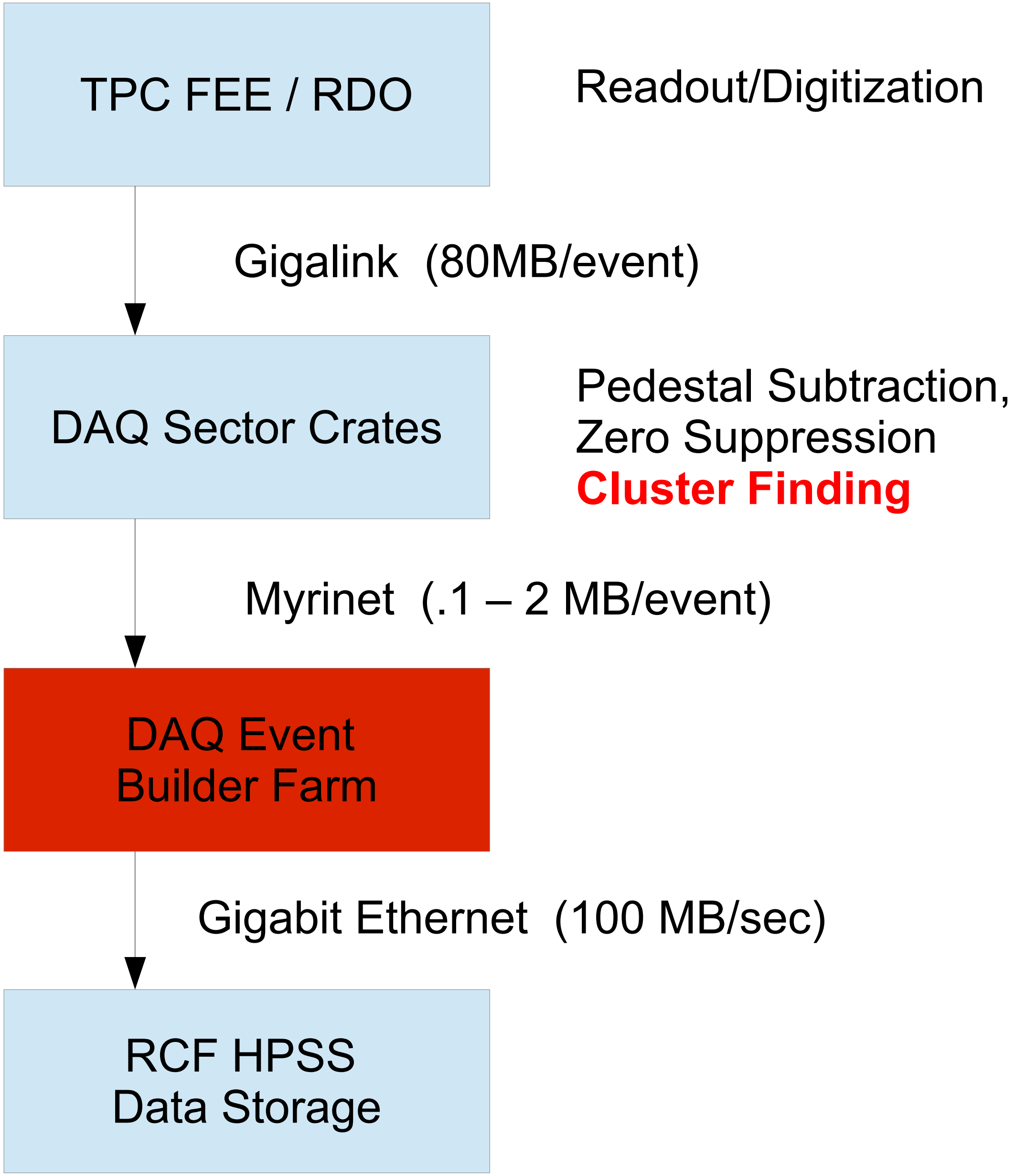
DAQ 100 (2002)

We had extra processing
Power in the i960 CPU's

* Added 2-D cluster finder in DAQ sector crates
* Added a farm of event builders

Resulting in:

* Additional factor of 5 reduction in data volume
* The DAQ was able to keep up with the full rate of TPC digitization / fiber transfer
* Reduction of analysis resources (At the time cluster finding took 80% of the analysis Time)



TPC FEE / RDO

Readout/Digitization

Gigalink (80MB/event)

DAQ Sector Crates

Pedestal Subtraction,
Zero Suppression
Cluster Finding

Myrinet (.1 – 2 MB/event)

DAQ Event
Builder Farm

Gigabit Ethernet (100 MB/sec)

RCF HPSS
Data Storage

The bulk of the effort for this upgrade was in demonstrating that the online cluster finder Was good enough to replace the far more sophisticated offline cluster finder.

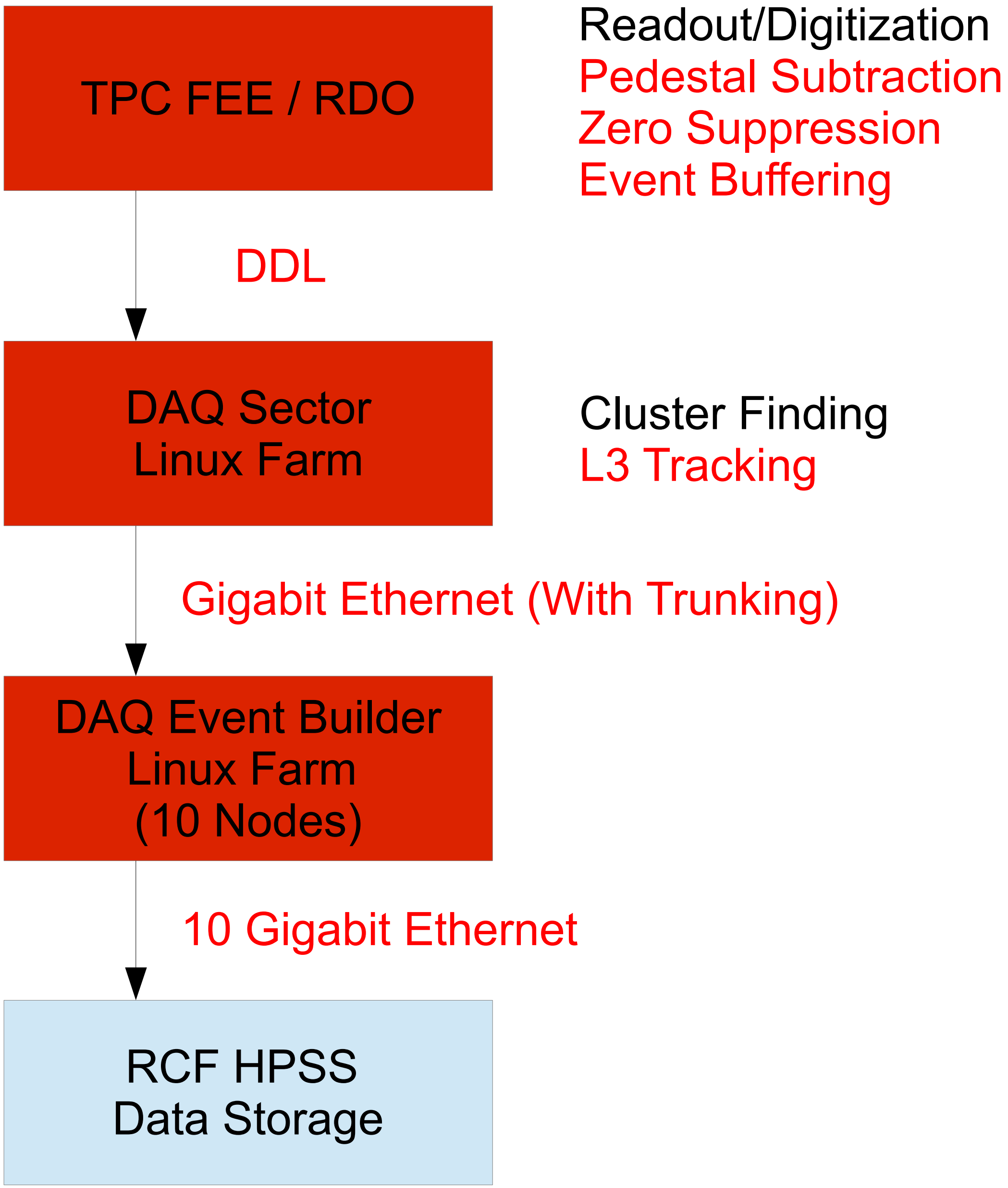
DAQ 1000 (2008)

The only way to increase the performance of STAR was to upgrade the TPC readout electronics

* We used technologies from ALICE
  - The Altro Chip
  - The PASA Chip
  - The DDL fiber interconnect (PCI cards for linux)

* Full re-spin of the TPC front end Electronics (The RDO, and FEEs)

* Replaced nearly the entire DAQ System (related to TPC)



TPC FEE / RDO

Readout/Digitization
Pedestal Subtraction
Zero Suppression
Event Buffering

DDL

DAQ Sector
Linux Farm

Cluster Finding
L3 Tracking

Gigabit Ethernet (With Trunking)

DAQ Event Builder
Linux Farm
(10 Nodes)

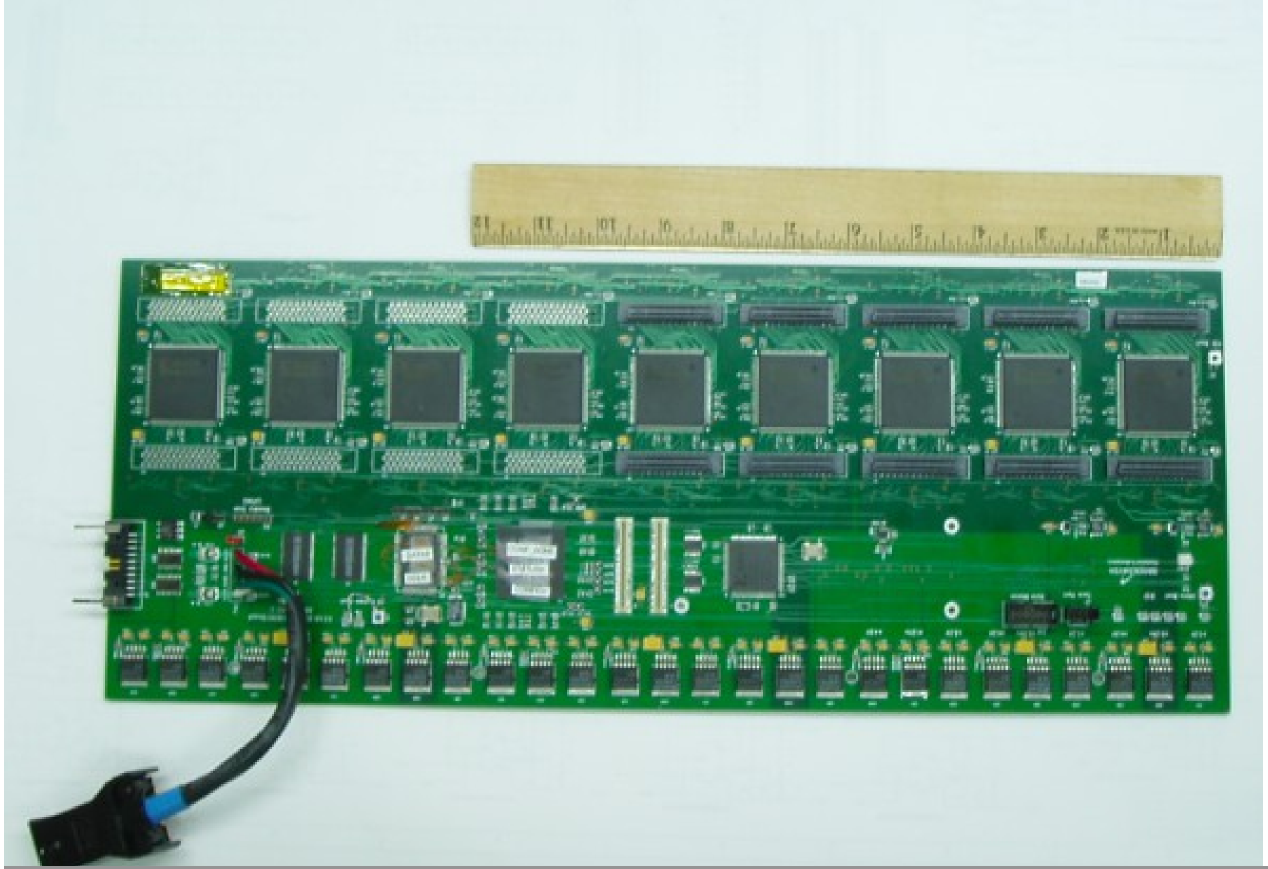
10 Gigabit Ethernet

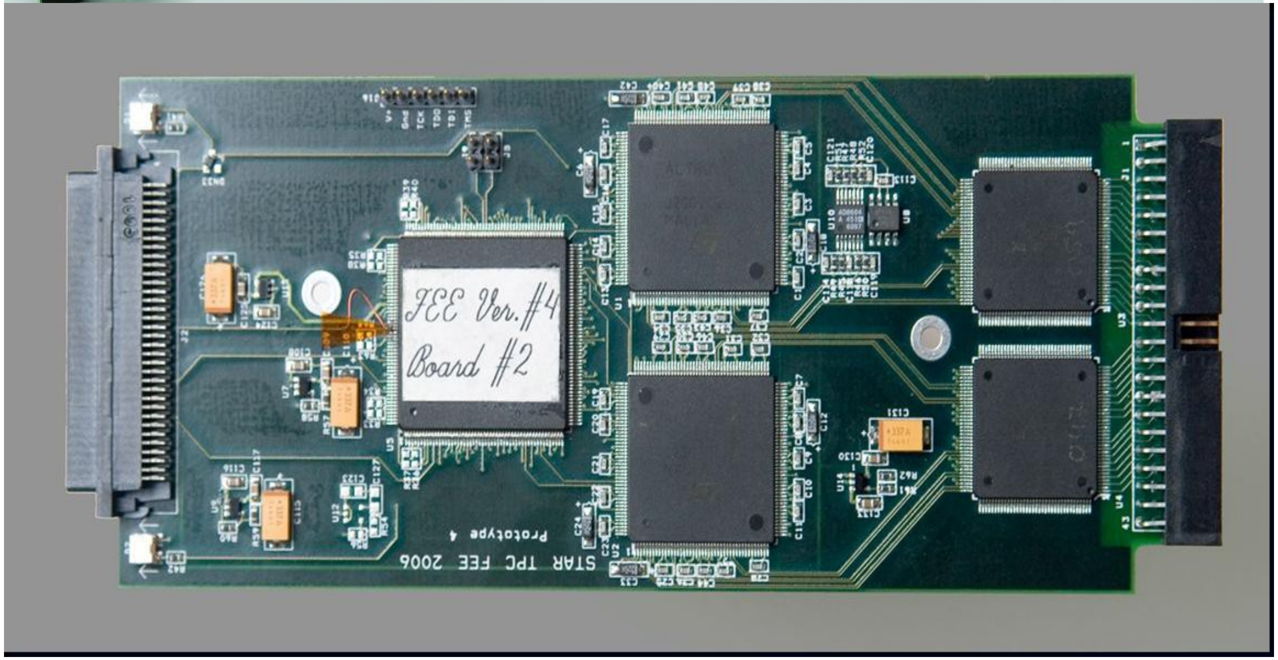
RCF HPSS
Data Storage

DAQ 1000 Hardware:

In addition to the performance Advantages, many new features

- Lower Power Consumption (Heat Load)
- Smaller (Freeing space)
- DDL is bidirectional. Can load all of the FPGA codes on both RDO and FEE at will:
  - Add features
  - Fix bugs
  - Auto-Fix radiation upsets!
- FEE's Buffer 4 events, so "zero deadtime" (40usec)
- Can access the Trigger commands history. (fire/abort
- Compatible with original TPC connecters and supports)

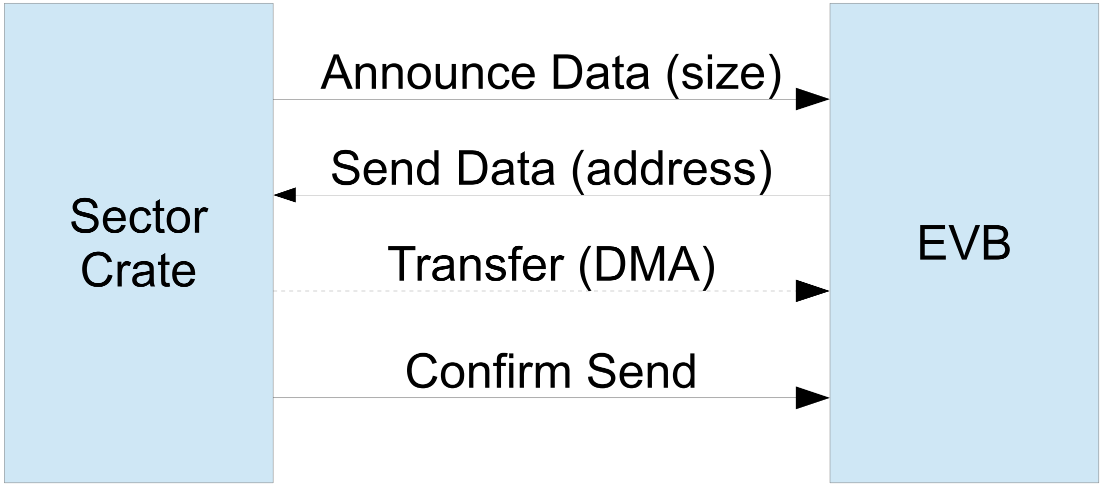




System Design Impacts of DAQ 1000

* The original DAQ used a Myrinet event building network

- High performance (similar to gigabit ethernet)
- Low latency (~30usec / message)
- Available in PMC for the MVME processors
- Avoided CPU usage to move blocks of memory by using “Direct Sends”, A DMA feature of the Myrinet network interfaces.
- However, required a complex protocol

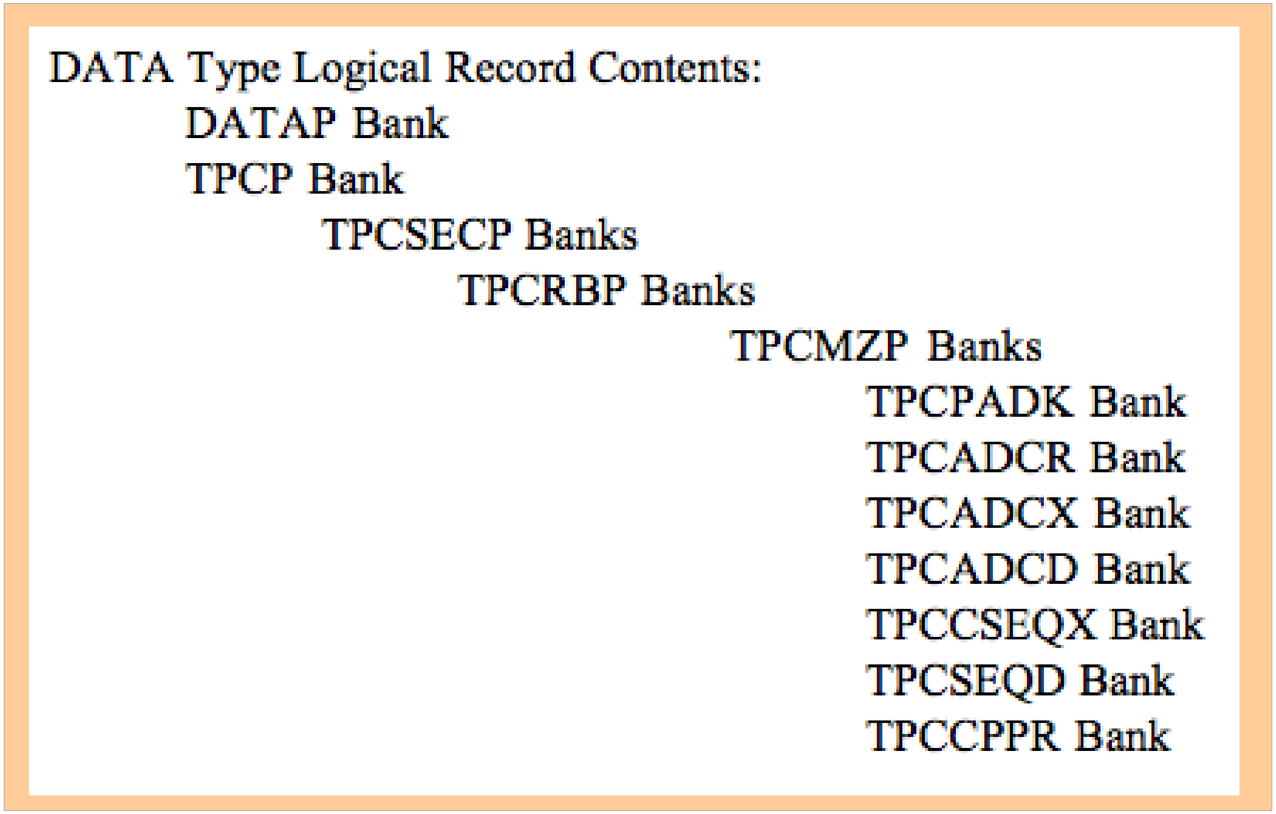


* We chose an Gigabit ethernet in for DAQ 1000

- Cheap, universally available, simple
- Faster Multi-CPU linux machines make memcpy() less expensive
- Simple “push” protocol eliminates the “chitchat”

System Design Impacts of DAQ 1000   (Continued...)

- Data Formats
  - The original DAQ data format was defined by a set of hierarchical pointer banks.



DATA Type Logical Record Contents:
- DATAP Bank
- TPCP Bank
  - TPCSECP Banks
    - TPCRBP Banks
      - TPCMZP Banks
        - TPCPADK Bank
        - TPCADCR Bank
        - TPCADCX Bank
        - TPCADCD Bank
        - TPCCSEQX Bank
        - TPCSEQD Bank
        - TPCCPPR Bank

  - The scheme was based around the logic of the DAQ sector hardware
  - The offsets each bank required full knowledge of all the components sizes to calculate.
  - Adding new detectors, or data banks required defining new structures And updating offline readers, creating maintenance issues.

System Design Impacts of DAQ 1000 (Continued...)

* Created Simple File System (SFS) Data format

Volume Spec	
Head	
File 1	File 1 Binary Data
File 2	File 2 Binary Data
...	...

Type = “FILE”		
Byte Order = 0x04030201		
File Size		
Head Sz	Attribute	Reserved
Filename ...		
...		
filename length define by Head Sz		
File Data (File Size Bytes)		
...		
File length defined by File Size		

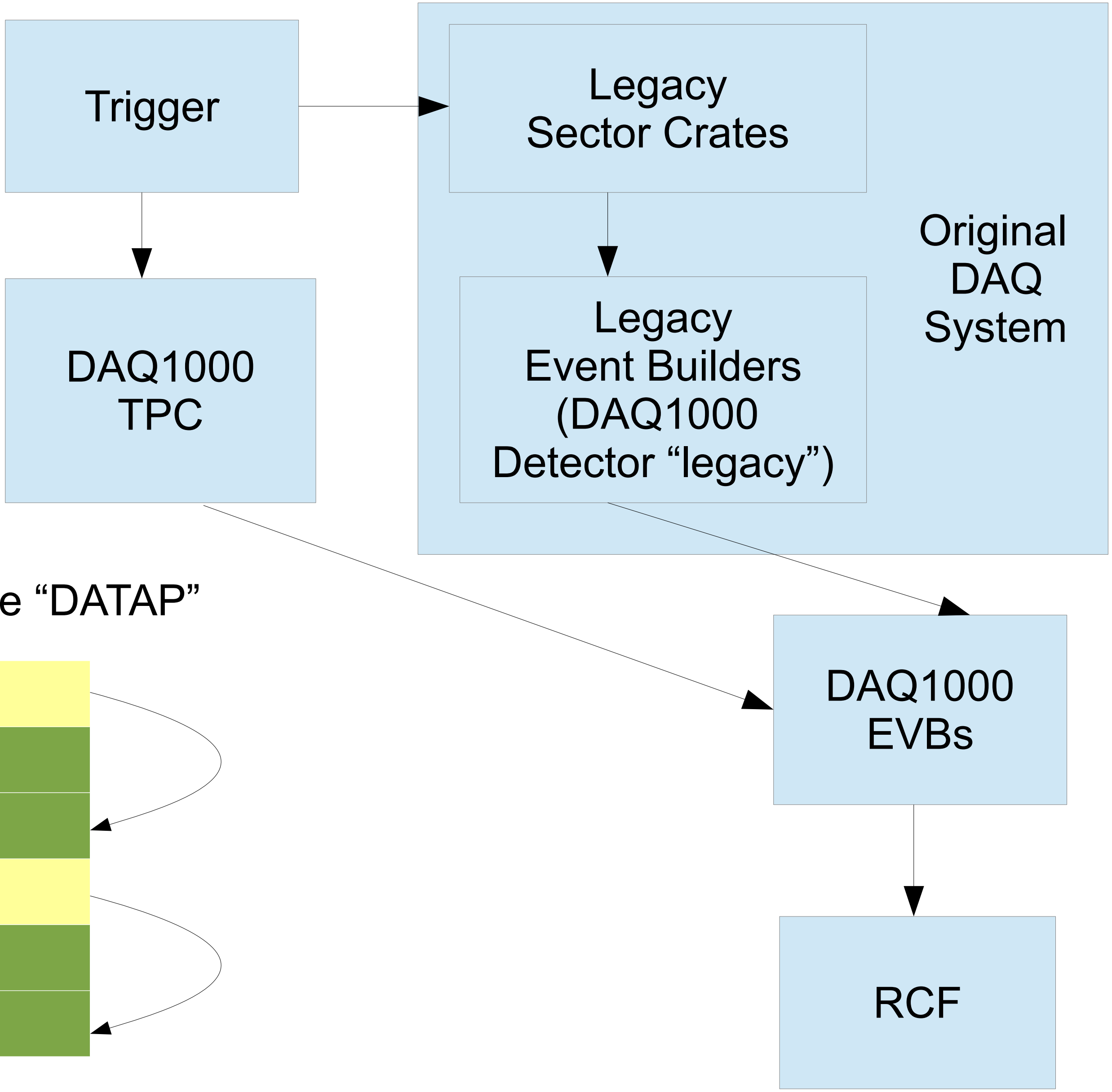
- The filename is typically just a local path
- The file is interpreted in order with A current directory implicit. The Current directory is affected by the Filename in conjunction with the Attribute field. (PUSHDIR, POPDIR, NOCD etc...)

* SFS has Many Desirable Features
  - low overhead
  - Full unix-like paths identify data banks
  - New detectors/databanks are trivially added / accessed
  - Easy data browser tools (unix commands: cd, ls, od etc...)
  - Files can be concatenated
  - Event building is simplified, because file creation is “local”

* One undesirable feature:
  - Slow random access
  - However, event data is processed by event...

System Design Impacts of DAQ 1000 (Continued...)

- * “Legacy” Detectors
  - – The TPC was upgraded In two steps
  - – The other detectors Still needed to be Supported.
  - – The legacy event Builders were Converted to virtual Detectors
- * “Legacy Detectors” in data File:
  - – SFS was modified to ignore “DATAP”

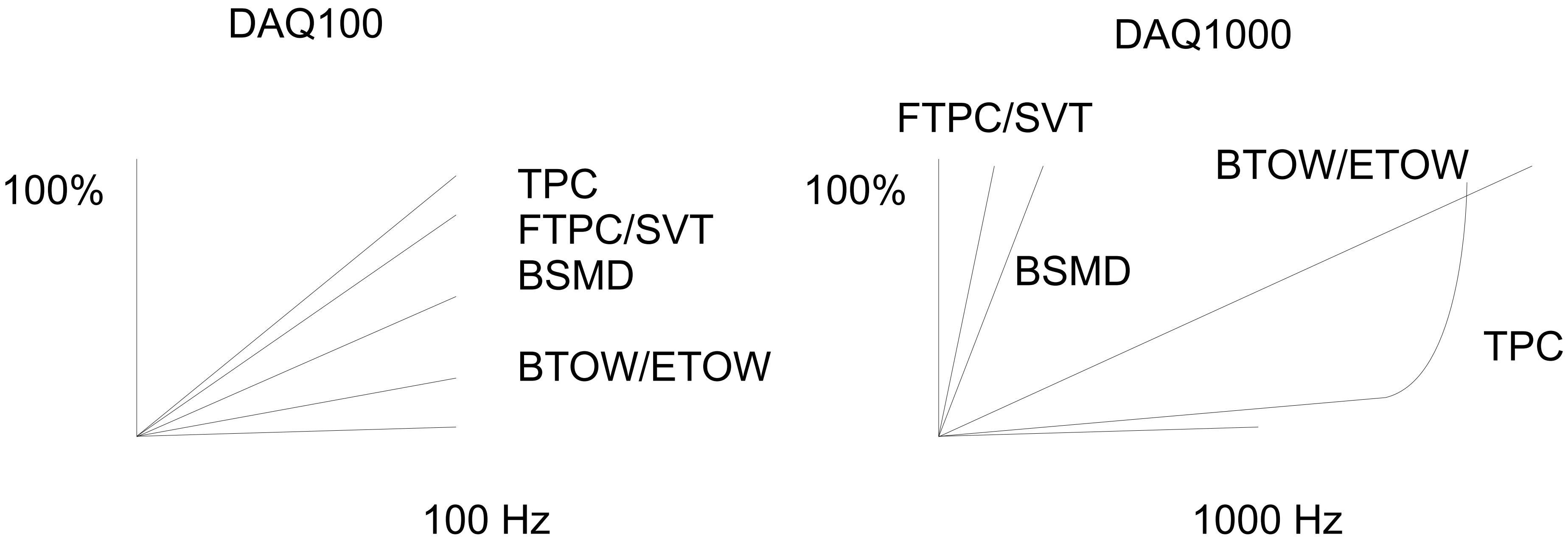


DATAP Event #1
SFS File: /#1/tpc
SFS File: /#1/legacy
DATAP Event #2
SFS File: /#2/tpc
SFS File: /#2/legacy

* The data files satisfied BOTH SFS and legacy data formats, and could be read by both!

System Design Impacts of DAQ 1000 (Continued...)

* Detector Deadtime Response:

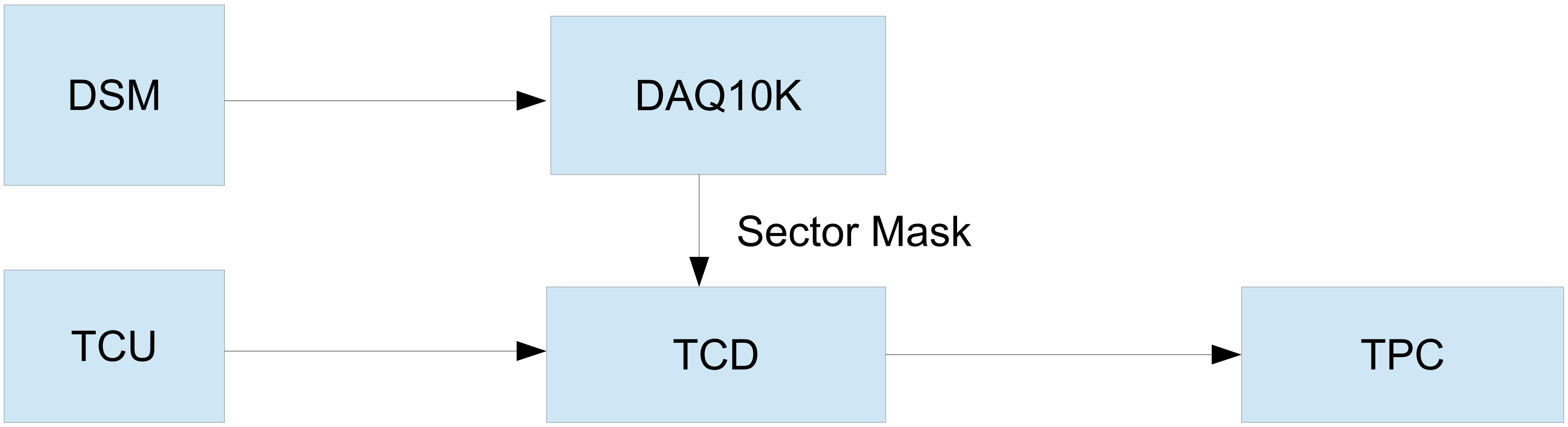


* The “slow” TPC, was now one of the fastest detectors!
    - -- This led to a major shift in our operations and trigger setups
    - – Still using old TCU
* The DDL link / Linux receiver CPU became the standard for new detectors and detector upgrades
* Detectors were all either upgraded, or retired over the next several years
    - – legacy event builders retired in 2011...

DAQ 10K

There is physics interest in processes that can be triggered using the TOF or MTD Which are only interested in limited geographical portions of the TPC.

DAQ10K allows the selective triggering of individual TPC sectors (2 sector granularity due to the routing of trigger signals)



1. The TCD is provided a TPC mask every bunch crossing, based on the TOF/MTD Information.
2. When the TCD is triggered, it checks the trigger action word to determine whether the event needs the entire TPC or only the indicated sectors.

Implemented/tested during 2012 run, but still awaiting first physics use!

DAQ Summary:

* The STAR DAQ has undergone numerous changes over the years resulting in more than a full turnover of all the hardware, and many software components.

* The capabilities have been enhanced in multiple ways:

  - Event rates (from < 10hz to > 2khz)
  - Data rates (from < 30MB/sec to > 1GB/sec)
  - Low dead-time operation
  - Enhanced reliability and replaceability due to use of commercially available equipment.
  - Enhanced flexibility for operation and upgrade

* These capabilities have been added while maintaining robust STAR operations.

High Level Triggers in STAR

* STAR was originally envisioned to have 3 software based triggers.
  - sold using idea that we could sample more luminosity than we could readout through DAQ.

Level	When	Time Budget	Information	Action
L1	TPC Drift Time	40 usec	Contents of L1 DSMs	Abort Readout
L2	TPC → DAQ transfer time	10 ms	Contents of BEMC/EEMC All DSM Data	Abort Transfer
L3	Event Assembly Time	100 ms	TPC sector tracking BEMC/EEMC	Throw Away Data

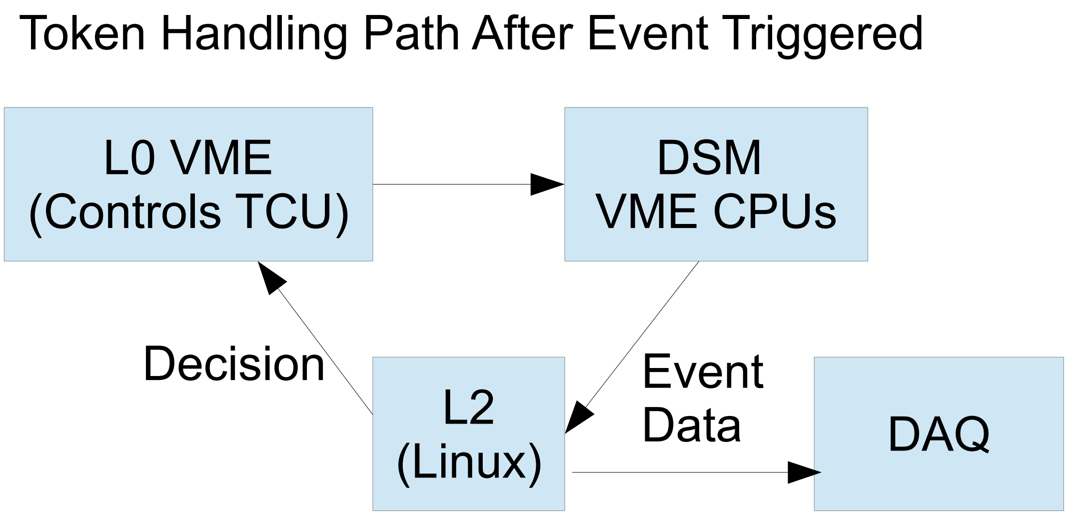
L1 Trigger:

The L1 trigger was never used as a trigger however, the mechanism proved vital to STAR

- The re-scaling process that was required to implement multiple trigger operation needed the processing of L1.

- Future protection was implemented using the L1 Mechanism:
  - A virtual pileup trigger was created
  - If this trigger fires within 40 usec after a trigger requesting future protection that trigger is aborted.
  - In the actual implementation of the detector electronics, the readout is not interrupted. It finishes and then is dropped.

L2 Trigger:


Token Handling Path After Event Triggered

L0 VME (Controls TCU)

DSM VME CPUs

L2 (Linux)

DAQ

Decision

Event Data

The L2 system has been used extensively, particularly in the polarized proton program. We've had triggers for jets, di-jets, upsilon Jpsi, Ultra-peripheral events and others. We also used it to work around TCU issues by emulating hardware triggers.

However, for the first several years, it could not be run as designed because detector front end electronics didn't handle the protocol correctly. We invented the “L2.5” abort, which did not actually abort the readout. Instead the event was thrown away by DAQ after it was read out.

The detectors did not all accept aborts until 2006.

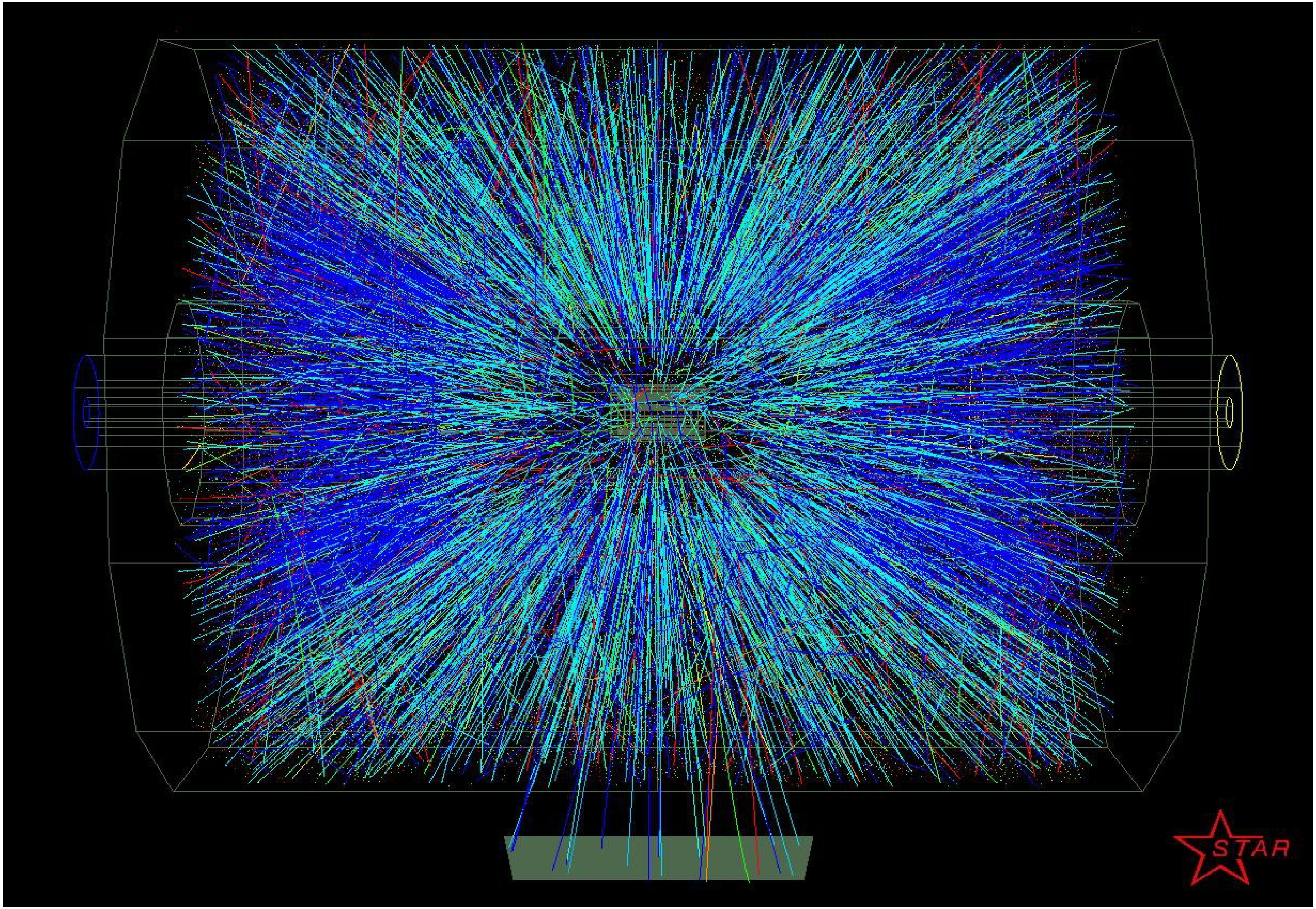
L2 Trigger: (Continued)

* After 2007, the DAQ1000 upgrade made the readout time of the TPC less than the readout time of the EMCs.
  - -- Impossible for L2 aborts to reduce the dead-time
  - – Actually, L2 aborts increase the dead-time, because the TPC stays dead while waiting to determine an events fate.
  - – So, we reverted to the L2.5 abort scheme

* In 2012, the L2 mechanism was entirely removed from the trigger system, and ported into the Event Builder farm.

* This has not impacted the real benefits of L2:
  - – Calibration and monitoring
    - -- Our formal online QA only samples a small fraction of events. L2 samples all events, and provides detailed diagnostics for the spin program
  - – Reduction of data volume on tape
  - – Marking events for “Express Stream” analysis

L3 Trigger (2000-2002)



L3 Trigger (2000-2002)

The original L3 system was a farm of ~30 Linux CPU's integrated into the Event building network of the DAQ system.

The intention was to track events and make complex trigger decisions based on the tracking. In this model the L3 system could process 100Hz, while the EVB could only write out 1Hz, so there would have been a significant advantage in sampled luminosity.

L3 was never able to process more than about 30Hz of data, and the project was killed by the DAQ100 upgrade.

Still the original L3 system had significant impact on STAR:

- The fast L3 cluster finding algorithm was adapted to make DAQ100 possible
- The tracking algorithm was ported into the online QA application, and continues to give us best crosschecks between the TPC and trigger performance
- The original creation of the express stream feature (which can be applied for any trigger) was invented according to the needs of the L3 system.
- An online event display was developed by this group, and later ported to a stand Alone application
  - Best quick early debugging tool
  - Highly useful for PR and tours and was the “face” of STAR in popular publications for years...
- First experience with network booting linux farms...

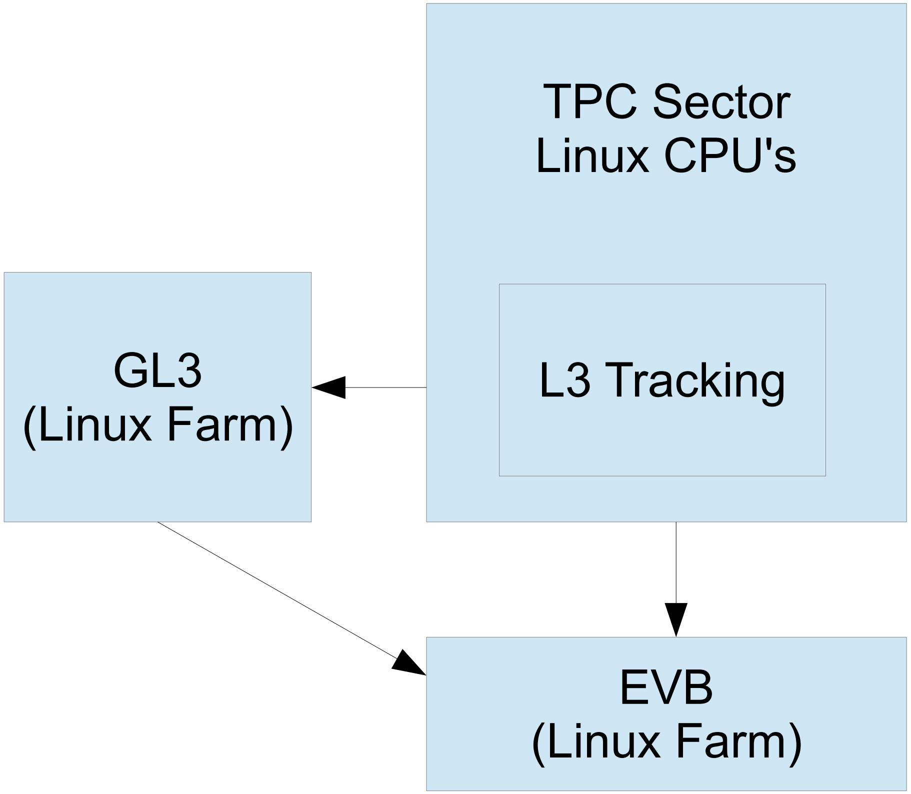
L3 Trigger (2011-2012)

Re-spin of the L3 trigger

* Tracking done inside TPC sector CPUs

* L3 decision made by a Small farm (4) Linux CPUs

* The tracking can keep up with Gold-gold collisions but not with High luminosity PP (where there is a lot of pileup)


TPC Sector
Linux CPU's

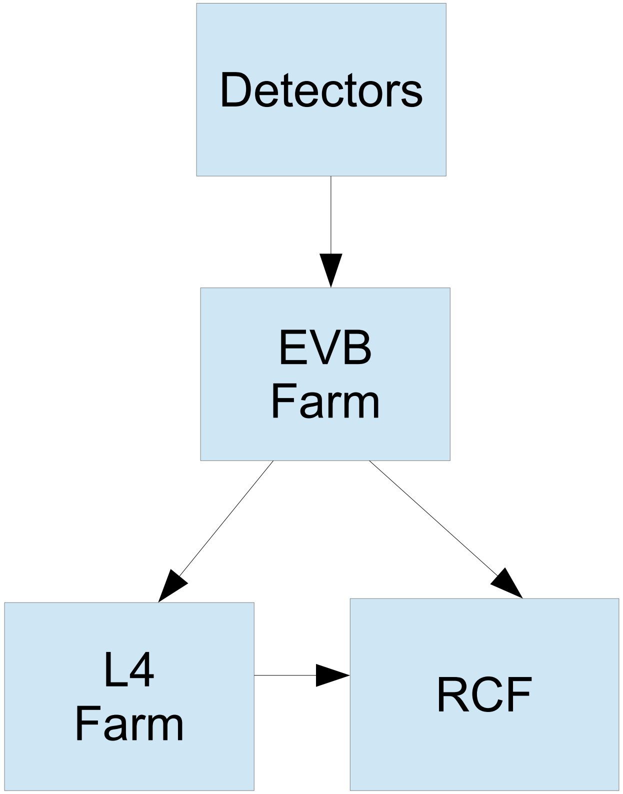
L3 Tracking

GL3
(Linux Farm)

EVB
(Linux Farm)

* Used (among other things) to produce the Anti-He discovery in a short time.

L4 trigger (2013):



The success of the re-spin of L3 prompted more support For the HLT trigger to (eventually) process all of the STAR event data.

* “L4 Architecture” advantages
  - Entire TPC available for tracking
  - All STAR detector's available for processing
  - There is no longer any requirement for event processing latency (as long as the farm is large enough to maintain the Throughput)
  - The L4 is not as coupled to the internal DAQ Protocol, so the systems can be more stable.

* Development on L4 continues
  - Tracking algorithms
  - GPU's
  - Physics algorithms
  - Plans for increasing the L3 farm size

High Level Trigger Summary:

The goal of sampling more luminosity than the DAQ is capable of reading out has never been satisfied.... However, HLT triggers have been a key element of STAR throughout it's lifetime.

* The cost advantage of saving tape storage is significant

* The time advantage of having smaller data sets (while maintaining useful information) is significant

* The time advantage of marking interesting data sets for priority analysis is significant

* The challenge of HLT triggers does good things for the collaboration:
  - attracts motivated people
  - gives them strict performance goals to beat
  - it allows them to rethink basic things like tracking which tend to get compartmentalized in the collaboration
  - it gives alternate paths for event information to be used for monitoring, QA, and data analysis.

Summary

I hope I have given an accurate picture
of the STAR DAQ/Trigger Systems and how they have
evolved over the years.