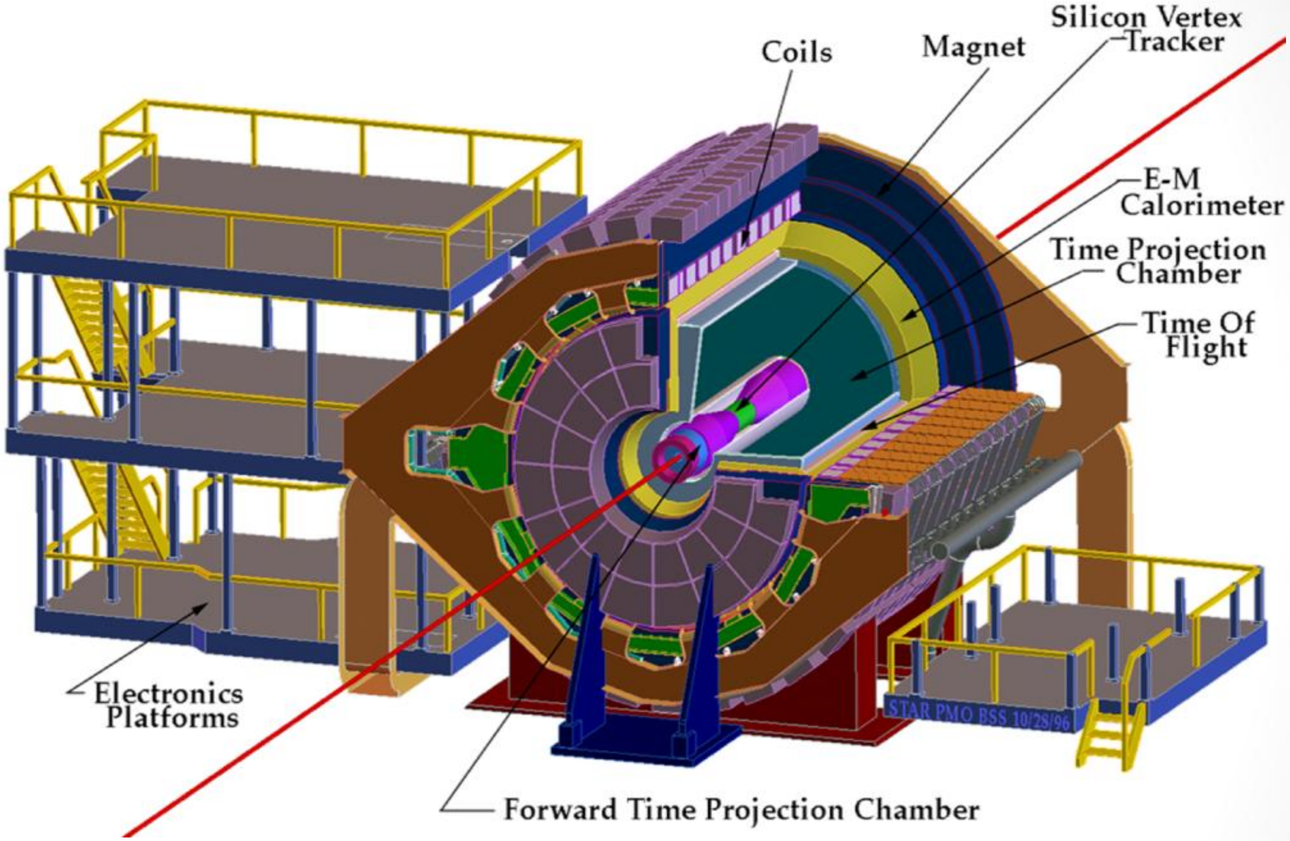
STAR framework and HLT

Jérôme LAURET

BROOKHAVEN NATIONAL LABORATORY

Outline

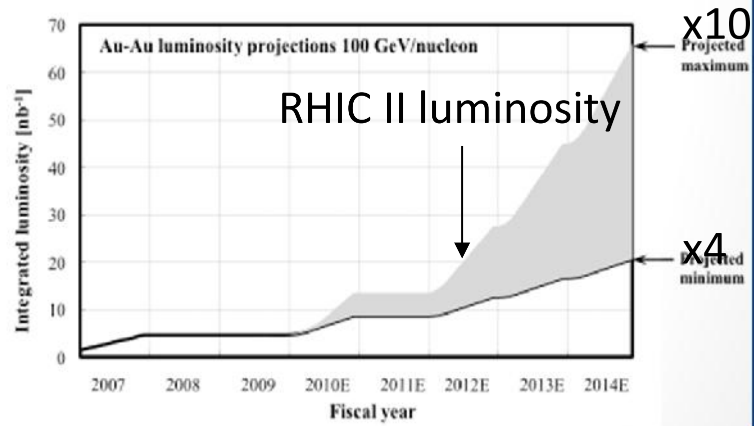
- STAR brief introduction
  - A very live experiments with changing needs and many challenges
- STAR framework
  - General ideas, reminders, components
  - Extrapolated to “the needed” for a collaborative project
- Recent key projects in STAR – offline reconstruction
- STAR HLT project
- Summary



STAR IN BRIEF

STAR – a A+A / p+p NP experiment (for now)

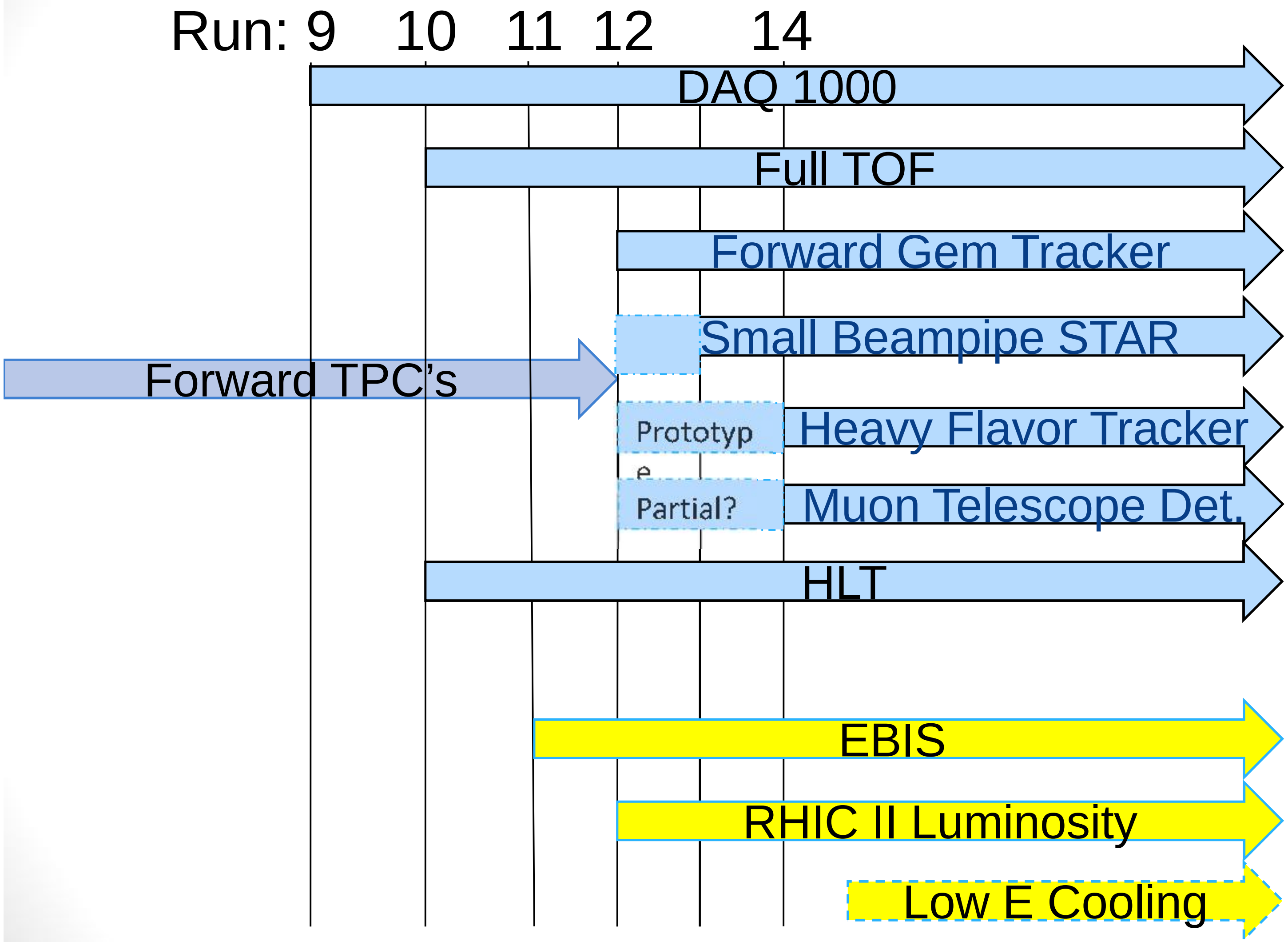
- STAR - **S**olenoidal **T**racker **A**t **R**HIC
  - Located at the Brookhaven National Laboratories (BNL/USA)
  - Heavy ion and polarized proton machine
    - *Polarized* protons sqrt(s_{NN}) = 50-500 GeV
    - Nuclei from d to Au (U), sqrt(s_{NN}) = 20-200 GeV
    - Beam energy scan: as low as 5.0 GeV
- An ever changing detector for precision physics DAQ upgrades:
  - Staged DAQ upgrade @ STAR (x100 in 2004, x1000 in 2008, 10k and beyond now)
  - MTD, HFT, FGT
  - HLT = High Level Trigger
  - HFT – precision decays



$26\ nb^{-1} * A^2 =$
$1\ fb^{-1}$ pp-equivalent

- Versatility implies a morphing framework
  - Had to deal with pileup, distortions, …
  - Pileup rejection vertex finder, …
  - Need to extend NOW for eRHIC / eSTAR era (framework should support R&D and be ready for data production)

STAR Timelines

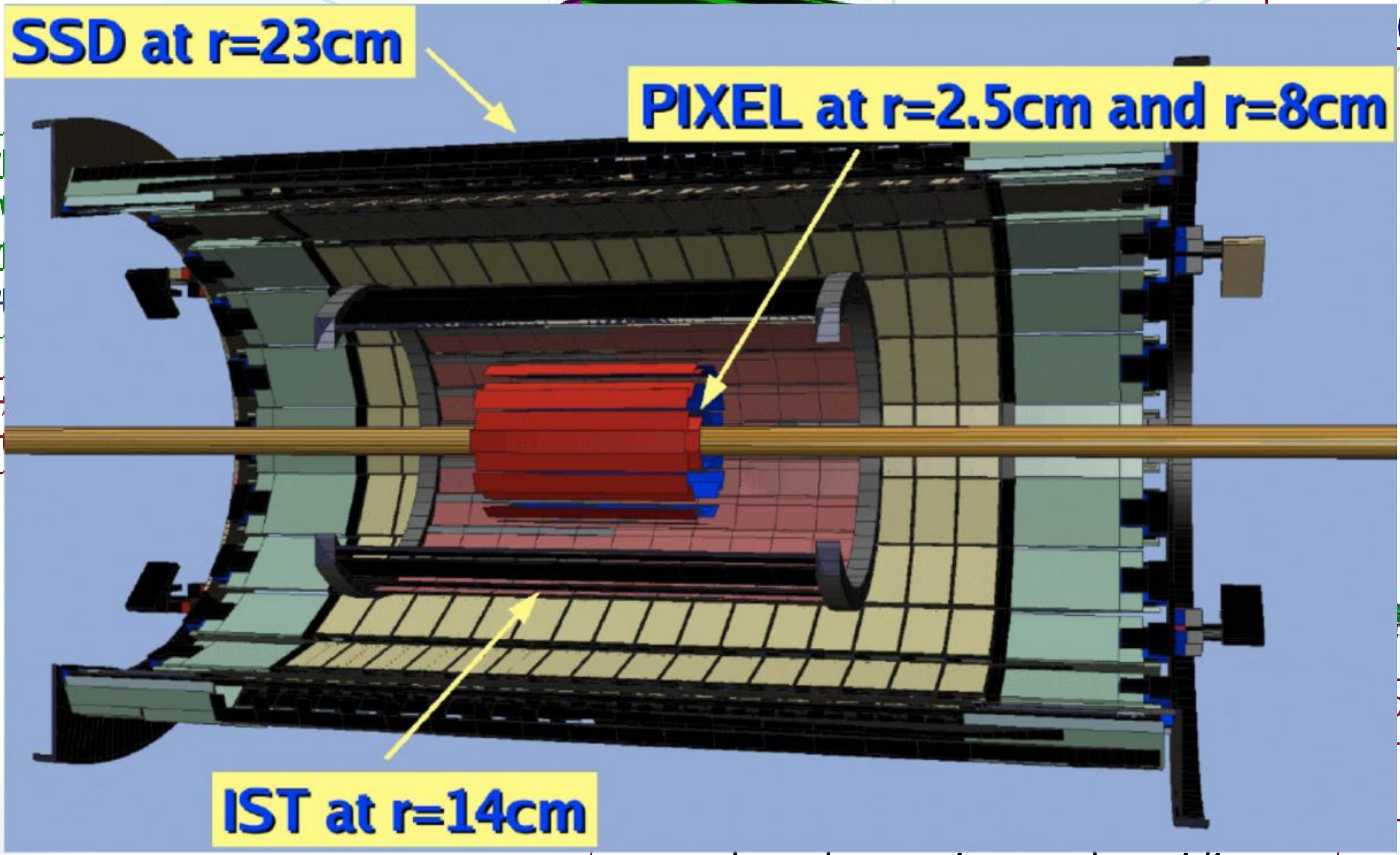


STAR: A Correlation Machine

Particle ID: TOF

Electromagnetic Calorimetry: BEMC+EEMC+FMS
(-1 ≤ η ≤ 4)

Tracking: TPC



...rward Gem ...acker ...011)

over a broad range in pseudorapidity

(STAR, “rough” summary)

- Very vibrant program, rich set of research fields ever expanded by our scientists ingenuity
  - BES, Anti-Hypernuclei - not in the initial RHIC long term plan
  - Large data samples brings new topics, new opportunities
- Wave of upgrades and fast DAQ will require precision tracking, forward tracking, fast tracking (HLT or Hybrid), vertex with pile-up considerations, low multiplicity vertex, …
- STAR (framework) even supports eSTAR and possibly (eRHIC) …
  - Here again, VF important, asymmetrical event in e+A first
  - More challenges ahead
- Fore more information, see the GSI workshop talk or Yuri Fisyak’s presentation on tracking @ STAR

STAR FRAMEWORK

General ideas and notions

- Previous talks at this workshop (and others) can be found on this topic - see for example
  - STAR Framework, Software architecture
  - root4star: a ROOT based framework for user analysis and data mining

- What this talk will focus on / repeat is …
  - What makes a framework successful and resilient to aging?
  - What do we need as standards and rules and why?
  - How is this relevant for this workshop …

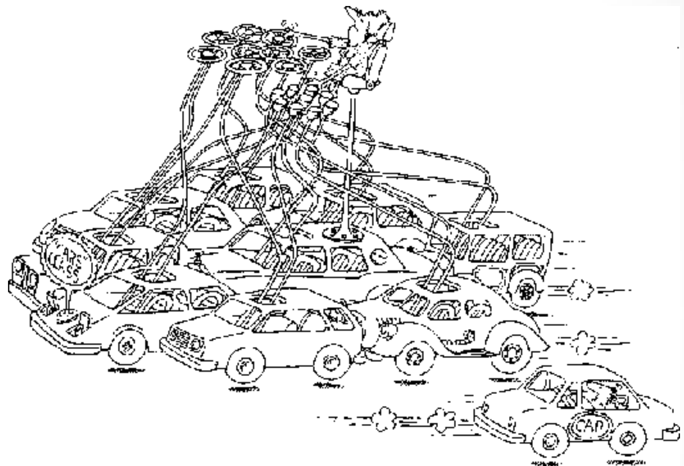
General reminder – root4star

- STAR had a few phases of framework
  - staff - framework was not that easy to extend and was opaque
  - Early adoption of ROOT ~ 2000+ - root4star = ROOT + STAR specific + Geant + …
  - ROOT + few libraries (will) suffice / for now, a few static linking in "root4star"
- root4star - A single purpose framework for
  - Simulation (Geant3, Geant4) – VMC
  - Data mining (reconstruction, DB interface)
  - User analysis
  - Other (GUI, EventDisplay, common IO layer)
- Built-in OO model, C++, ROOT class, utilities

Single framework seem to be a monolithic approach but provides easier maintenance (small team) and flexibility (re-usable components) in the long run.

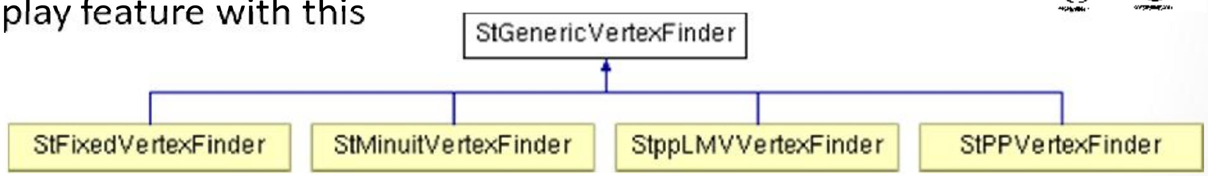
Basic concepts and components

- Steering component + datasetss: centralized common tasks & define data interface
  - Allows the construction of hierarchical organizations of components and data



- Makers – base class StMaker
  - Consumer / provider model – base element of a chain
  - Special ones - ex: VertexFinding,…
    - Generic OO base class with well documented interface
    - Chain indicates which impl. is used
  - Full plug-and-play feature with this approach

KFParticles?



- Chains (May be sequential or parallel)
  - An instance of a “chain”, a “steering” component
  - All dependencies are sorted out for you - user declares “my maker will need db, tpc clusters and geometry”

Easy and implicit layout – plug-and-play “Makers” or “Generic” virtual classes.
NO NEED TO KNOW the dependencies from the users’ stand point
Should be easy to add/remove/replace components on a need-basis

Not an easy path …

- Initial efforts to be done
  - Makers: Template provided – user can start from example
  - Model is based on TDataSet – input/output and interface well defined – **common data structure a MUST for any collaborative project** – initial effort pays off

- But NOT ENOUGH - code is subject to rules
  - Code is version control (doh!) - **changes easily seen by all**
  - Modules or "Makers" are arranged into a directory structure. Directories structure relates to components and compilation/linking rule (naming standards)
    - StRoot – ROOT related classes. Below StEvent, StXXXMaker (XXX represents a 3 letter acronym for a sub-system), if the maker contains "Db" will likely link to a database, …
  - Coding standards made an applies to "some" extent
    - Code (peer) review impose coding rules & portability
    - Rules intended for Fast debugging : Code crash because a variable of name XX is undefined: is this user code or core-code? Or ROOT code?

- Another MUST - documentation
  - Documentation and comments should accompany ANY code. It is the ONLY way to achieve resilience to time
  - Documentation may be modernized – doxygen-like comments can be added. But DO NOT add a method without a minimal explanation
  - DO NOT assume you are eternal and DO NOT assume you, yourself, will remember what you did … 5 years down the road

Version control + data structure coding & Standards + documentations: the pillar to a good lifecycle

(framework conclusion)

- What makes a framework successful and resilient to aging?
  - Central repository, all informed of changes, changes themselves accessible by area with a few top maintainer (core, db, detector sub-system, …)
  - Its initial design & architecture – this takes time but pays off.
    - Keys: achieving plug-and-play, steering components allowing extension “under the hood”, interfaces and data structure definitions are important
    - **Common data structure a MUST for any collaborative project**
  - Its easiness of use, extend, morph, … its modularity (OO model in our case)
  - Single framework and re-usable / common components makes it easy to maintain and extend (everyone is wrong or everyone has it right) – it is one of many ways

- Why do we need rules?
  - Coding standards allows many to work together with a basic expectations of interface, conventions, …
  - Easier for developers and maintainer to debug (know in advance if class members of user code, base class, method name, what it is about, …)

- **How is this relevant for this workshop …**
  - **For the current project to move forward, we will HAVE TO discuss rules, conventions, data structures, documentation, changes control, …**
  - Not all are in place – already an issue (CA integration in STAR / finding)

STAR

STAR, A LIVE FRAMEWORK

Brief STAR tracking history …

- STAR tracking history went through two cycles
  - TPT → Sti
    - Sti has gave us ~ 5-6 years of quality data production
    - Final review also identifies minor issues at the time – were NOT a show stopper at the time
  - Vertex finding EGR/LMV → Minuit / ppLMV → Minuit / PPVF
    - Vertex finding changing as pileup increases – for now, x2, later x4 to x10
- “A” problem – STAR is a HI and p+p centric vision
  - Current VF very focused on single task (with much emotional attachment to something known to produce results – PPVF is a Spin PWG “trusted” VF)
  - Current VF are not considering e+A / eSTAR realities (a few particles, one direction)
  - Current framework MUST allow easier integration of R&D for eSTAR, eRHIC

Smooth transition with data/physics production and R&D not easy. Can rebuild a framework from scratch or see if morphing is possible

Current problems to address

- Simulation package
  - Geant-3 is past “obsolete” – we need to transition, Geant-4 out there and usable for ~ 1.5 year (no typo!), new Geant project started this year at CERN
  - FLUKA? … dead from start … but who knows
  - VMC project - propagators, swimming, … need to be abstract
- Geometries
  - Multiple Geometries to maintained while workforce is “thin” – StiXXX and Geant. A new geometry model should be abstract
  - Sti not “integrated” – track model has helix model, …
  - Sti geometries and tracking have had restrictions: plane and volumes perpendicular to the beam axis cannot be handled as now (r, Φ ordering), constant field along z, no radial
    - Forward detectors design not compatible with original Sti
- Going beyond
  - Trim away FORtran and 64 bits un-maintainable components
  - Move to ROOT based only framework
  - HLT a reality and so are GPU/M-core, Vectors – CA methods positive
  - **STAR moving to more upgrades – STAR → eSTAR → eRHIC**
  - Physics going forward, high precision, shorter decay length, …

- VMC tracking
- Abstract geometry model
- VMC simulations
- New algorithms, seed finder, vertex finders …
- Embedding framework reshape
- …

Ongoing projects

- Tracking component review in October 2011
- AgML – an abstract geometry model
  - Introduction of Sti introduced the need to maintain two different geometry models (one for reconstruction, one for simulation) increasing workforce load at a time when STAR is both active and ambitious in its future program as well as running thin on detector sub-system code developer.
  - The two separate geometries have consequences on embedding and simulation hence, our ability to bring efficiency corrections to the next level of accuracy.
  - Material budgets were found to be ill-accounted in reconstruction (dead-material were not properly modeled in the Sti framework).
  - The use of a common geometry model would have removed this issue
  - Single geometries easier for R&D (write once, get both for free)
- Stv – VMC tracker
  - Geometries made of planes and volumes perpendicular to the beam cannot be treated due to a technical choice (detector elements are organized in planes // to the beam, sub-systems assumed to be composed of elements replicated in Phi).
  - This precludes tracking in detectors such as the FGT. Our goal was to create an extended set of functionalities providing a truly complete integrated tracking approach, allowing the inclusion of hit information from other detectors (a key goal the inclusion of detector hits placed in the forward direction)
  - The use Monte-Carlo based propagators would allow better access to Eloss, better predictors and track swimming allowing for tracking in non constant B field (this is also not possible in Sti)
- CA – Cellular Automaton seed finder
  - The sudy of the CBM/Alice Cellular Automaton algorithm for seed finding was launched in collaboration with our GSI colleagues. Multi-core aware, the simple algorithm is thought to provide speed gains over the seed finding. Further work could spurse from this evaluation (online HLT) if successful.

Outcome

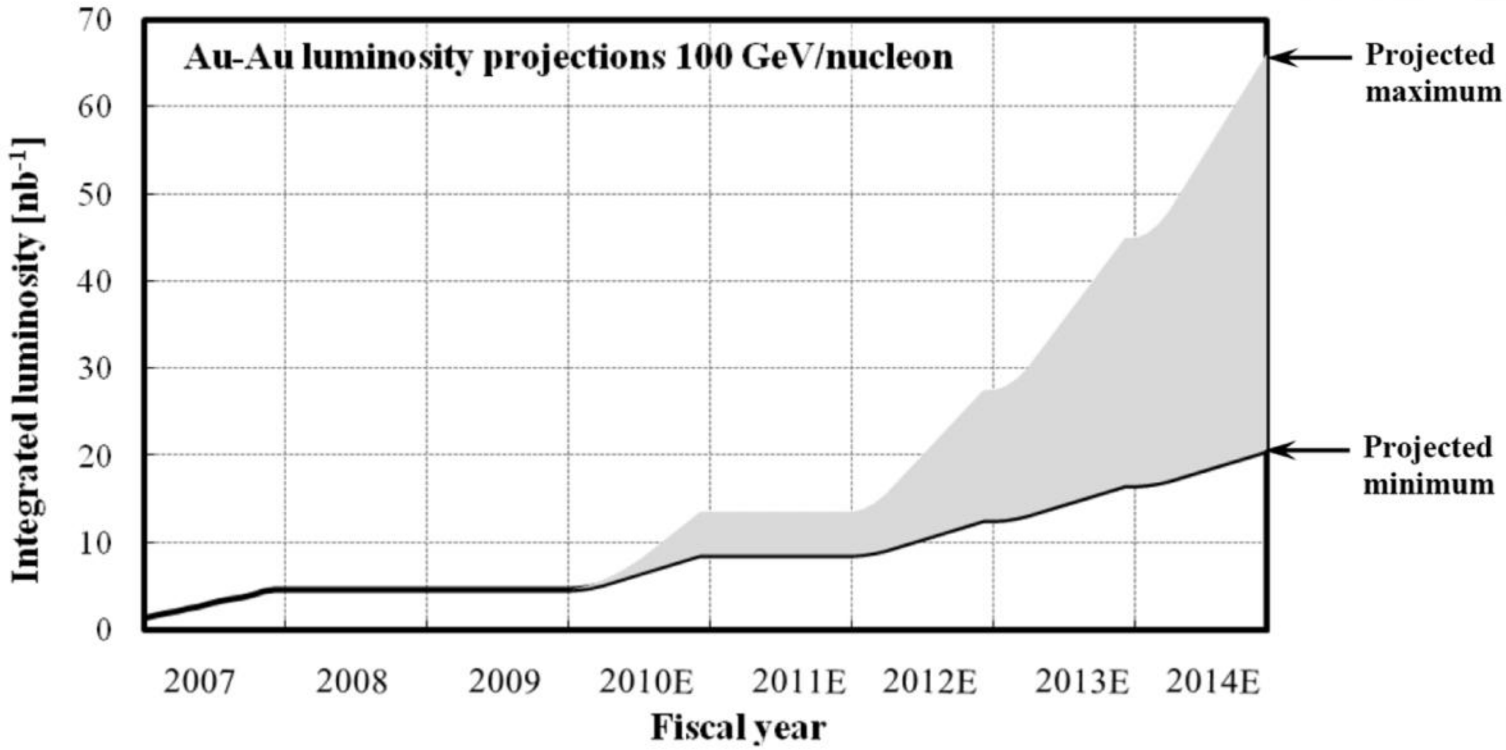
- AgML and CA accepted
  - Some adjustments to be made
  - At this stage, AgML opened issues and findings have been all addressed
  - CA will be revisited and integrated
  - Development strongly suggested to pursue and move toward the HLT
- Stv encouraging – not approved yet as not performing as well as Sti yet
  - New review June … To be Continued …
- In the interim – ongoing
  - Easier Plug-and-play event generator on the way
  - Consolidation of the event model
  - Consolidation of the embedding framework
  - New vertex finding

So far so good – morphing is ON and without distribution of the Physics

Aihong Tang

STAR HIGH LEVEL ONLINE TRACKING TRIGGER

Motivation



- Over the next a few years RHIC is expected to increase its delivered luminosity up to 8x1027cm-1s-1 for AuAu collisions at 200 GeV and 6x1031cm-1s-1 (1.5x1032cm-2s-1) for p+p collisions at 200 (500) GeV.
- To cope with the high collision rate, STAR has upgraded the DAQ system.

Motivation

- The improved data taking capability imposes a challenge for STAR on:
  1. computing resource in terms of CPU time and tape storage (cost impact)
  2. for analyzers, struggle with large data volume and bear with long analysis cycle.

- By implementing a HLT it will be possibly to reduce the amount of data written to tape by selecting desired events while still maintaining a high sampling rate to fully utilize the delivered luminosity for a wide range of triggers …

Physics Motivation

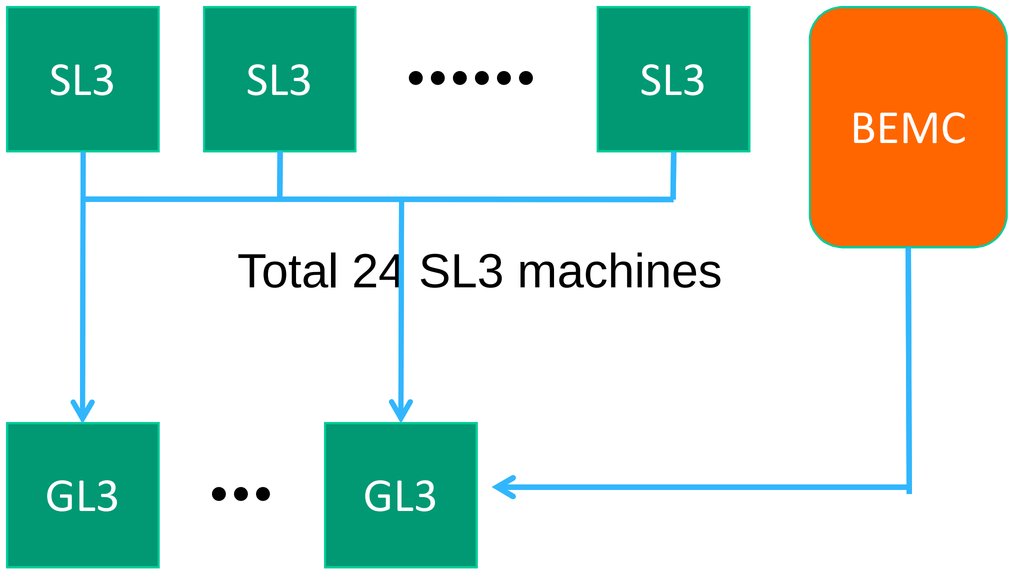
- Heavy flavor measurement
- EM probe
- High pt probe
- Search for exotics

Vision: **create A platform or "toolbox" for implementing interesting physics ideas.**

Related History

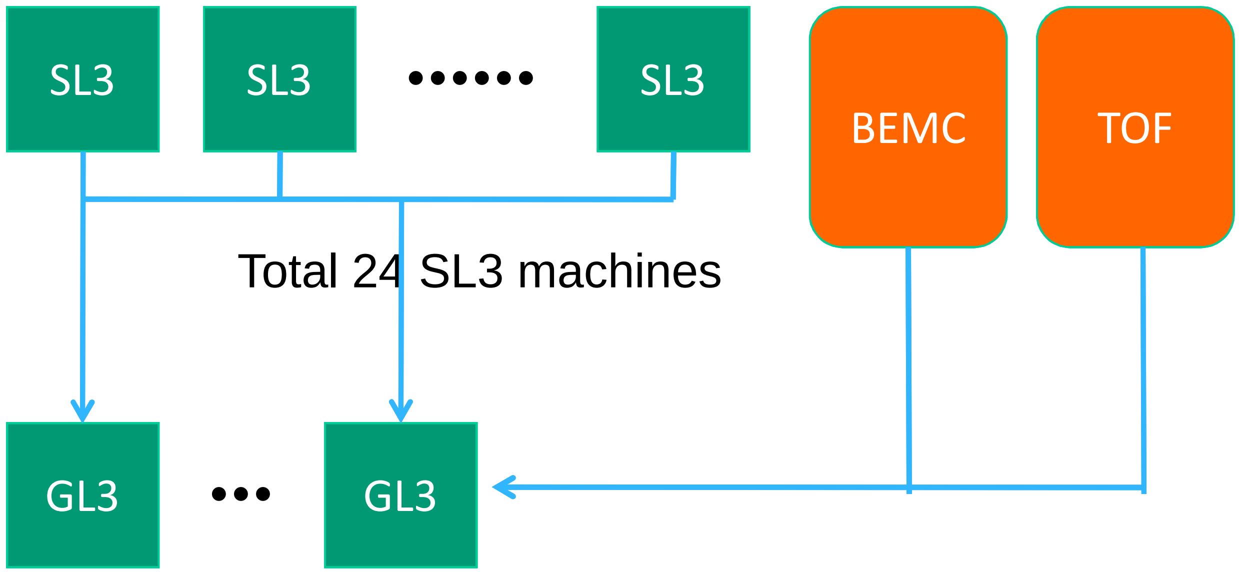
- STAR's old Level-3 system had been in limited function, phased out since then ~2002
- Propose of HLT at 2007 DAQ 1k workshop.
- Proof of principle in 2008.
- Prototype in 2009 with real data taking. DAQ 1k installed in 2009.
- In function in 2010.

HLT-2009



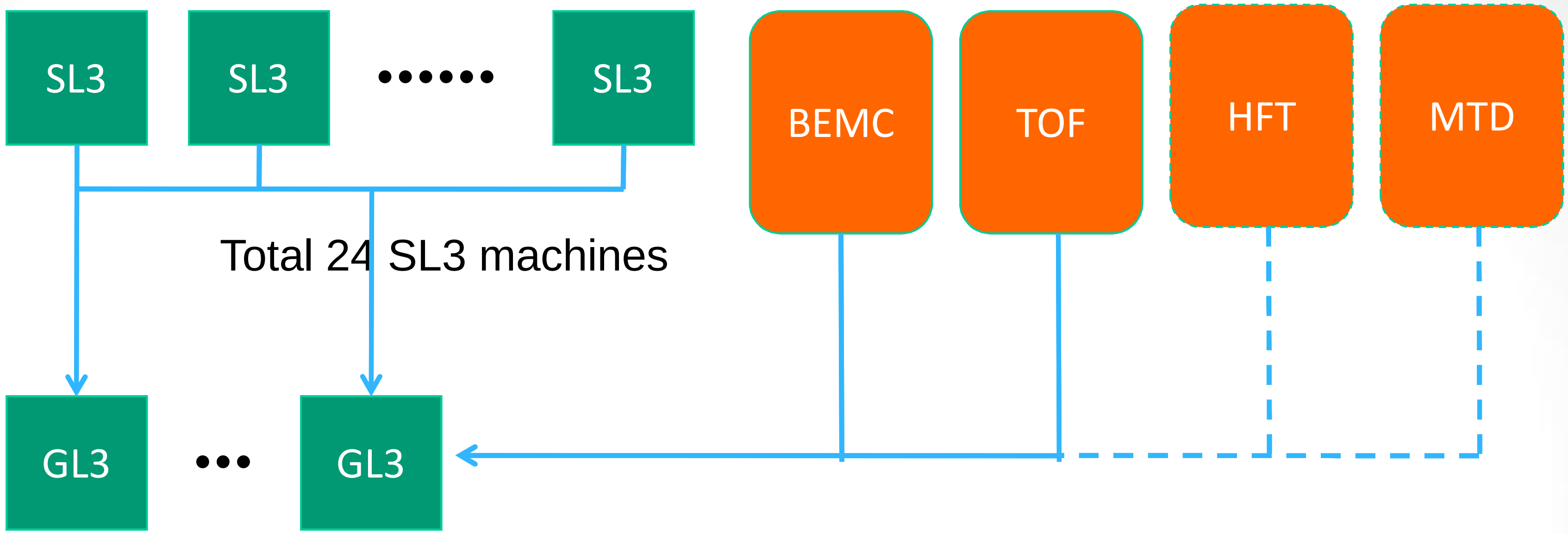
•Sector tracking (SL3) in DAQ machines (24 in total, each for a TPC sector).
•Information from subsystems (SL3 and others) are sent to Global L3 machines (GL3) where an event is assembled and a trigger decision is made.

HLT-2010



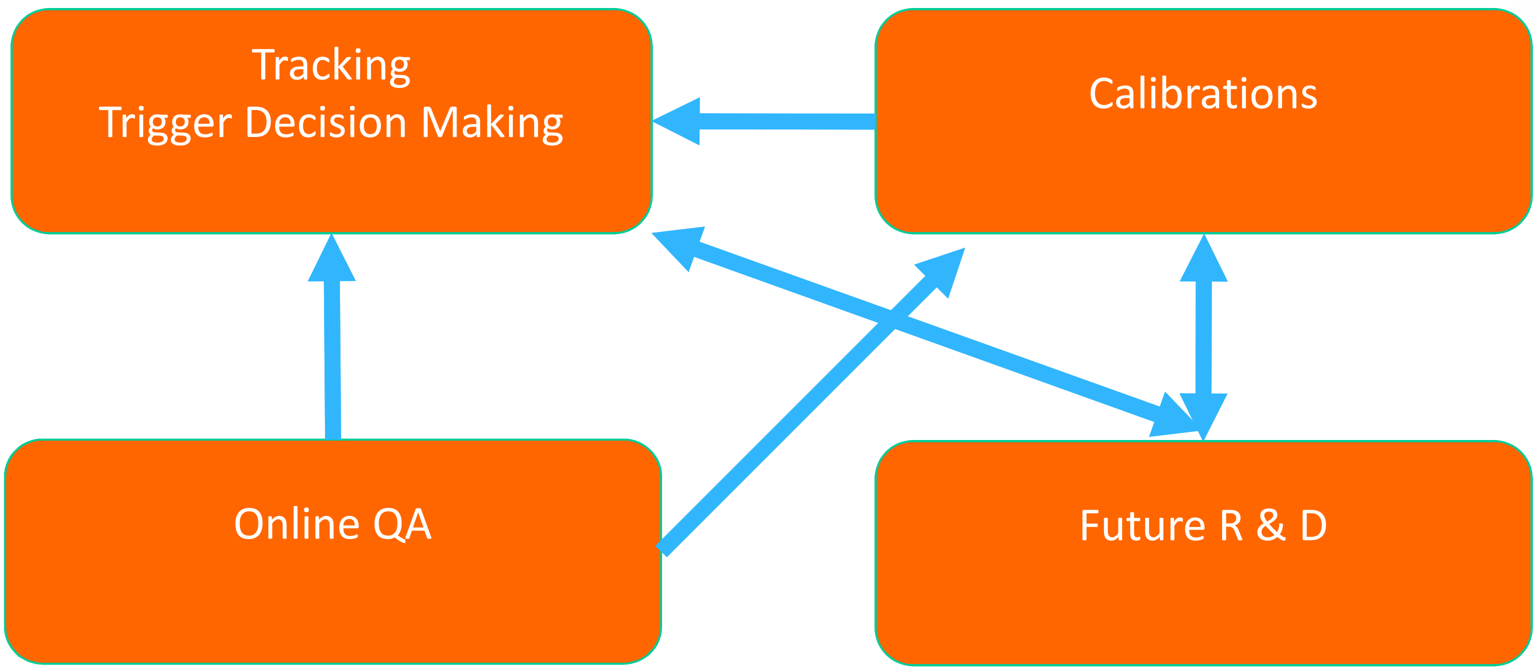
•Sector tracking (SL3) in DAQ machines (24 in total, each for a TPC sector).
•Information from subsystems (SL3 and others) are sent to Global L3 machines (GL3) where an event is assembled and a trigger decision is made.

HLT-2014



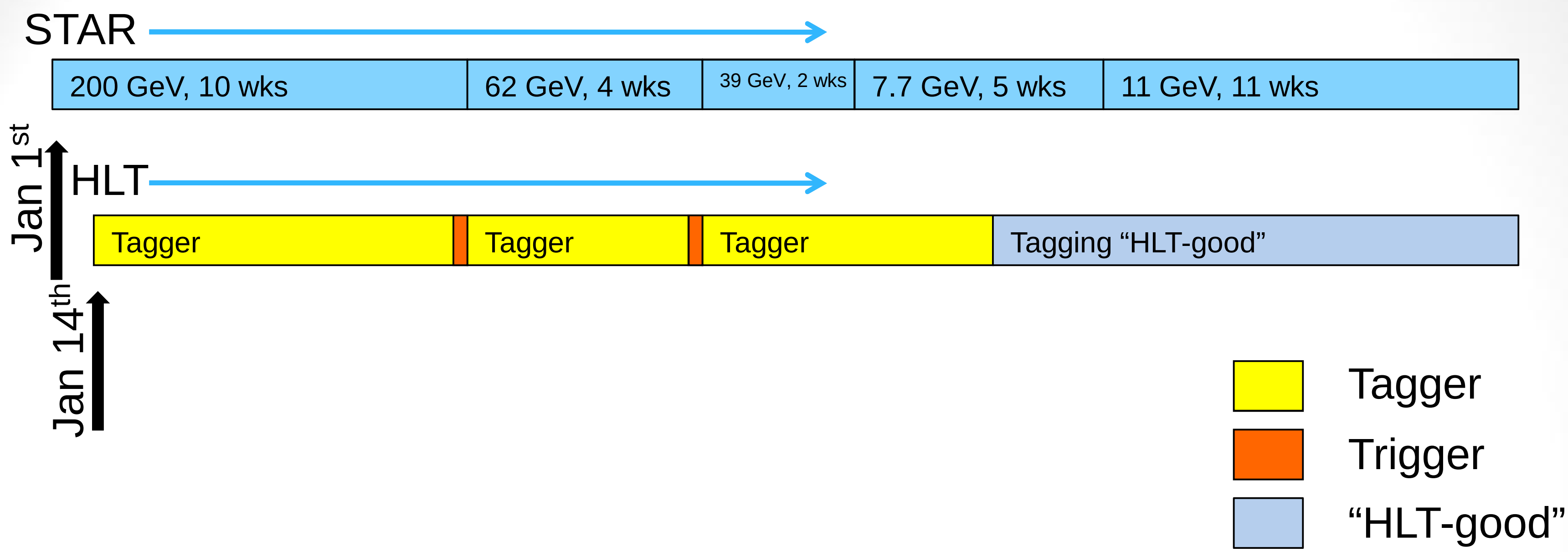
- Sector tracking (SL3) in DAQ machines (24 in total, each for a TPC sector).
- Information from subsystems (SL3 and others) are sent to Global L3 machines (GL3) where an event is assembled and a trigger decision is made.

HLT division of tasks



- Future R&D are input for calibration accuracy and tracking / trigger decisions
- Online QA provides feedback to tracking and calibration values/accuracy
- Calibrations essential to the whole process – trigger decision ↔ physics goals are intertwined

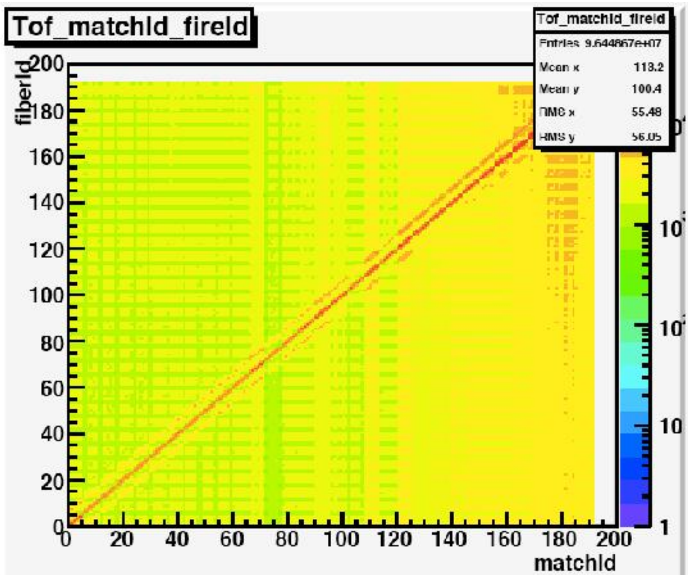
HLT-2010

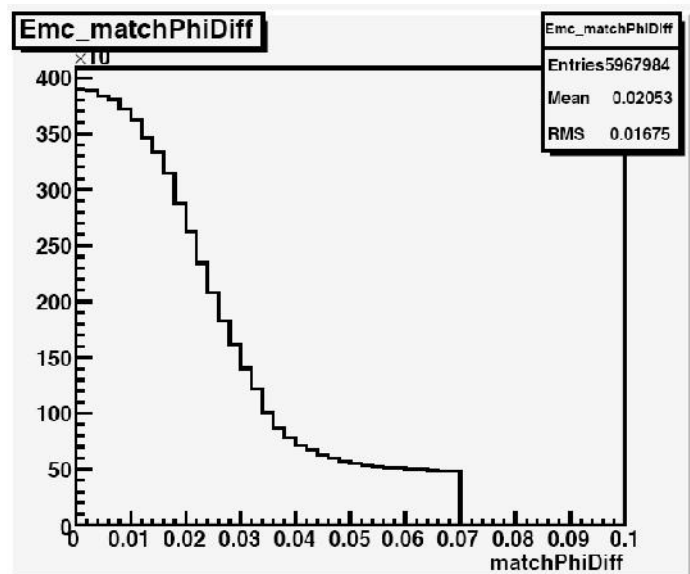


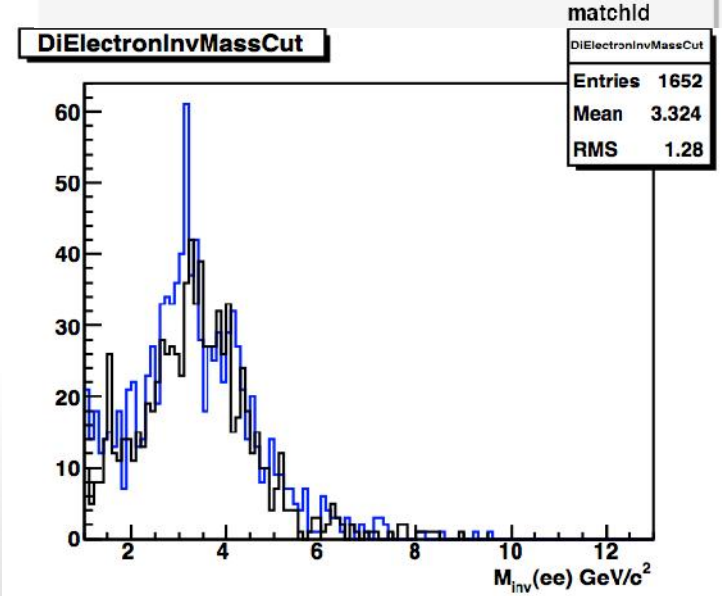
- Jan 9th. First TPX and TOF calibration ready.
- Jan 14th. HLT is up and running
- Jan 15th. L2 crashed. HLT running with TPX and TOF only for some period
- Feb. 05. HLT is decoupled from L2. Instead, HLT receives BTOWs from DAQ team

HLT tagger – flag added to the event to indicate a candidate
HLT trigger – event sent to a special "stream"
HLT-good – not sent to the "express" stream, not saved but used as monitor

Online monitoring



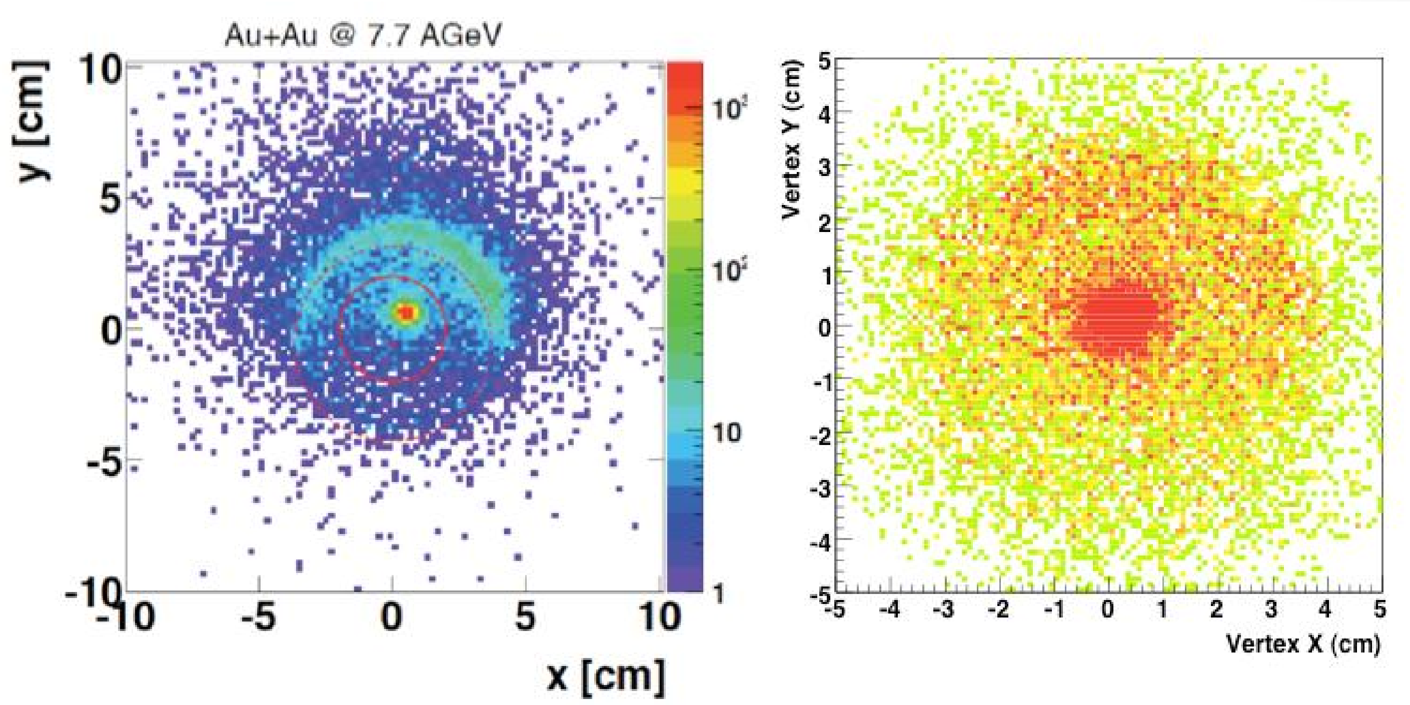




Watch J/ψ peak grow online.

Early discovery for possible run condition changes

HLT and RHIC Beam Energy Scan



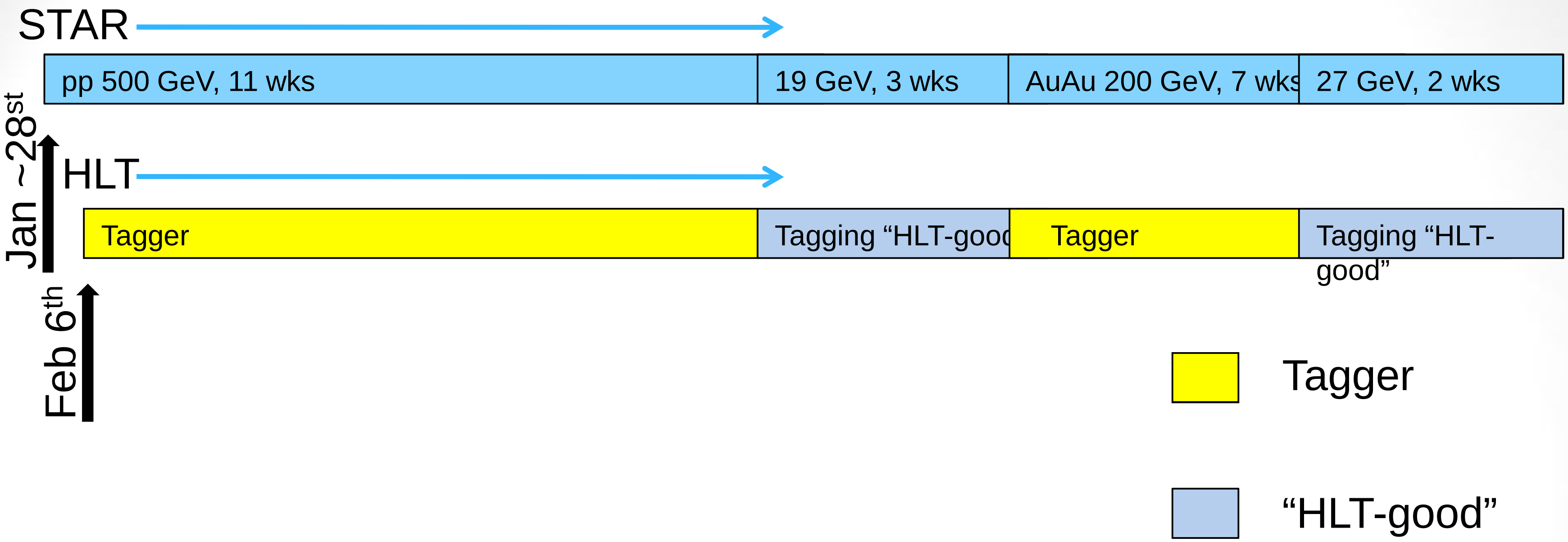
Off-line

On-line

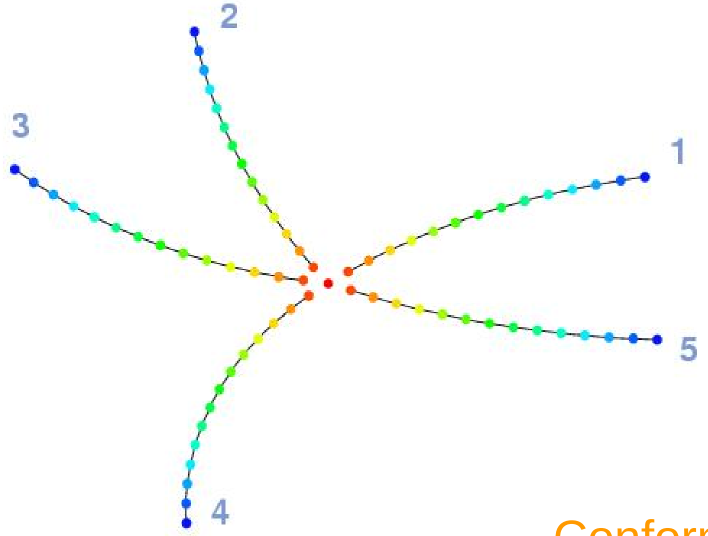
“HLT-good” purity and efficiency w.r.t. offline are both ~95%.

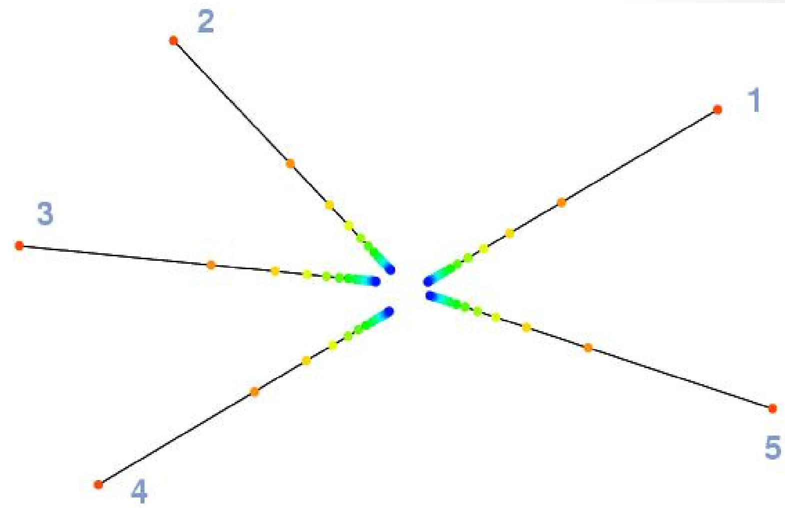
Feedback to CAD – beam tuning.

HLT-2011


STAR
pp 500 GeV, 11 wks
19 GeV, 3 wks
AuAu 200 GeV, 7 wks
27 GeV, 2 wks
HLT
Tagger
Tagging “HLT-good”
Tagger
Tagging “HLT-good”
Jan ~28st
Feb 6th
Tagger
“HLT-good”

Tracker





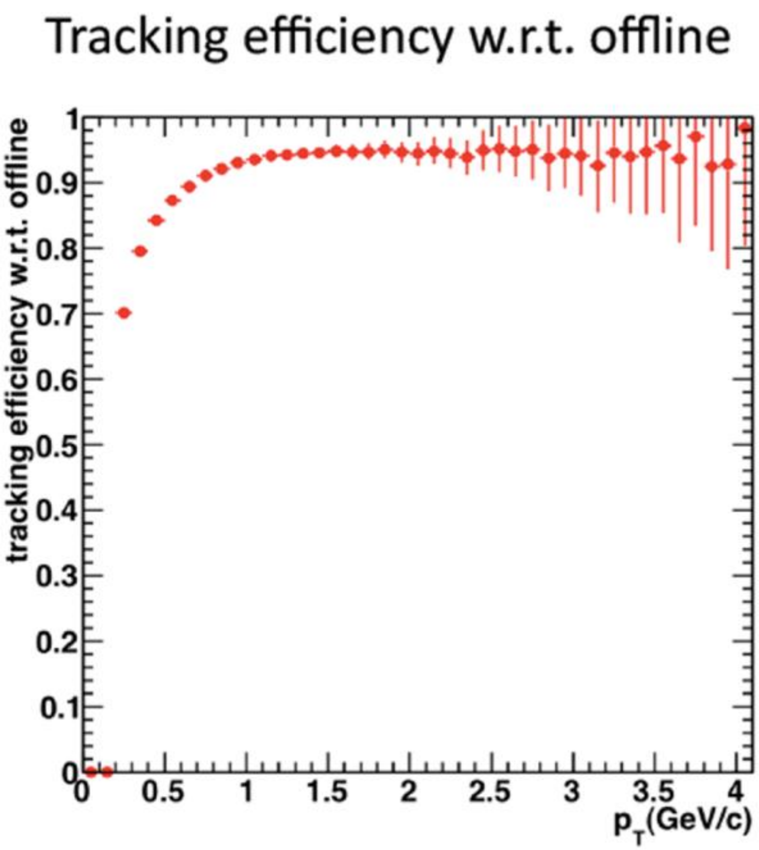
Conformal Transformation

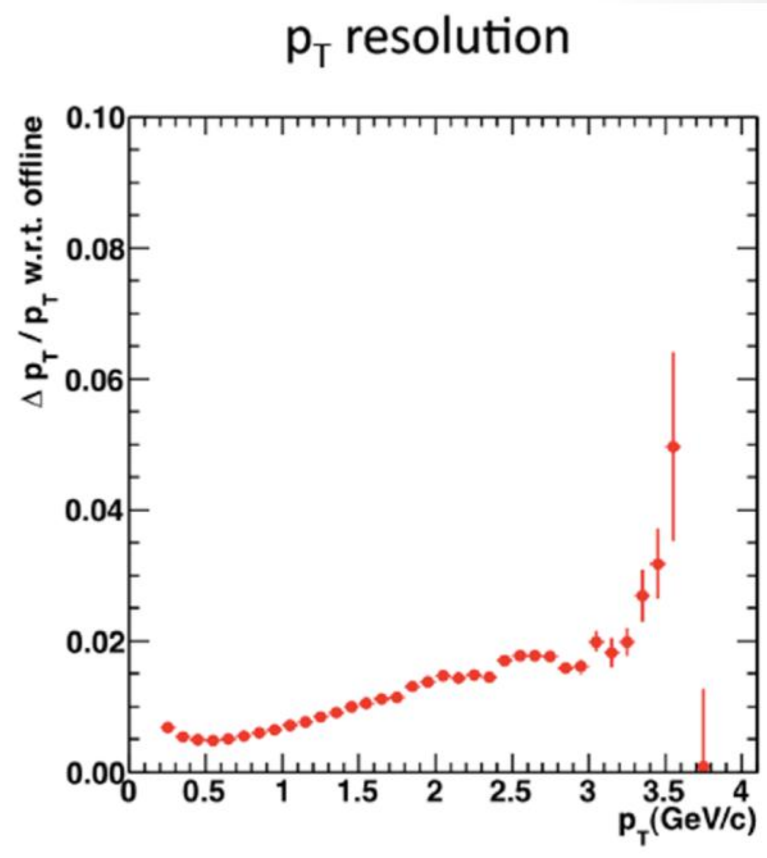
$$x_i' = \frac{x_i - x_0}{R_i^2}, y_i' = \frac{y_i - y_0}{R_i^2}$$

where $R_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$, and (x_0,y_0) is the primary vertex

Fitting lines instead of fitting curves.
Seeding from TPC out layers.
Final fit with Helix model in real space.
Fast tracker with acceptable accuracy.

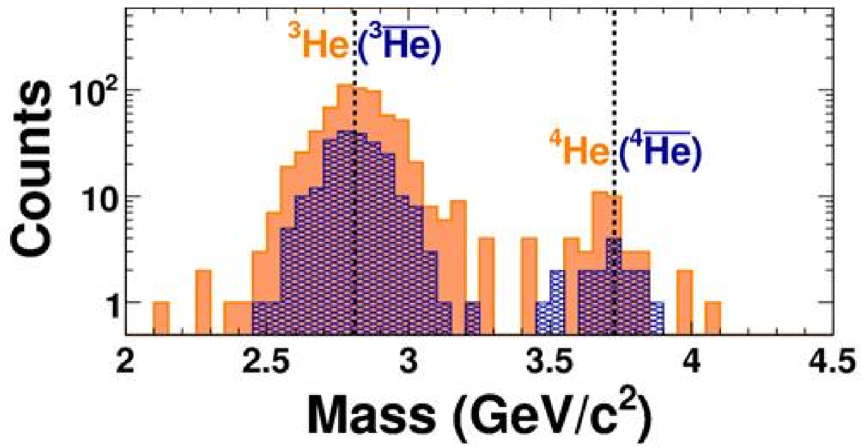
Online-offline Association



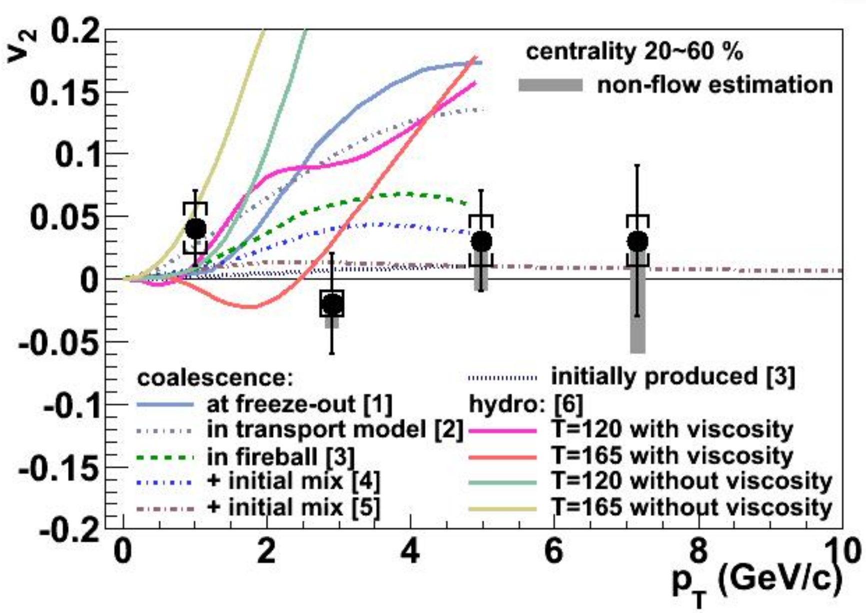


Assuming offline is 100% correct, efficiency of HLT with respect to offline …

Physical Results from run 2010

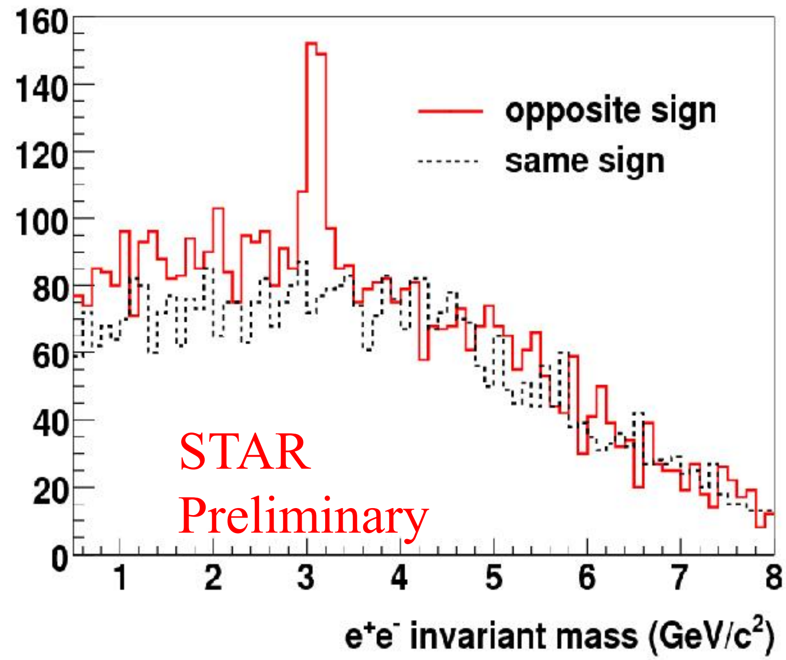


nature 473, 353-356 (May 2011)



J/ψ v_2 shown at QM2011

Di-electron from HLT online QA



- The first J/ψ signal seen at p_t > 5 GeV/c in AuAu 200 GeV collisions

Trigger Efficiency from 39 GeV data

Charge -2

Rigidity (GeV/c)	0.5-1.0	1.0-1.5	1.5-2.0
St_hlt	543 +/- 23	414 +/- 20	225 +/- 15
St_physics	615 +/- 25	420 +/- 20	227 +/- 15
Trigger efficiency	88 +/- 5 %	99 +/- 7 %	99 +/- 9 %

Di-electron

	Number of J/ψ	Number of e^+e^- pairs from γ conversion
St_physics	13 +/- 7.9	129 +/- 11.7
St_hlt	7 +/- 5.6	91 +/- 9.8

St_physcs == offline, st_hlt is the selected stream

Speed Performance (AuAu 200 GeV)

Year	2011	2012	2013	2014	2015	2016
Peak L ($10^{26}cm^{-2}s^{-2}$)	50	50	52	80	88	94
#TPX hits (minbias, central)	36.7k, 70.7k	36.7k, 70.7k	37k, 71k	40k, 75.6k	41.2k, 76.9k	42k, 77.9k
✔ Rate that HLT can handle (minbias, central)	2.0 kHz, 1.0 kHz	2.0 kHz, 1.0 kHz	2.0 kHz, 1.0 kHz	1.8 kHz, 970 Hz	1.8 kHz, 960 Hz	1.7 kHz, 950 Hz

Assuming half CPU cores of DAQ machine can be used by HLT, we expect that HLT can handle ~1k Hz for Au+Au collisions in RHIC-II era, however we have to keep in mind HLT is sharing CPUs with DAQ cluster finding code.

Speed Performance (pp 200 GeV)

Year	2012	2013	2014	2015	2016
Peak L ($10^{30}cm^{-2}s^{-2}$)	52	71	81	97	100
#TPX hits (minbias)	12k	16k	18k	21.5k	2.2k
✔ Rate that HLT can handle	6.1 kHz	4.6 kHz	4.0 kHz	3.4 kHz	3.3 kHz

Speed Performance (pp 500 GeV)

Year	2012	2013	2014	2015	2016
Peak L $(10^{30} cm^{-2} s^{-2})$	200	281	360	556	929
#TPX hits (minbias)	115k	174k	231k	374k	646k
Rate that HLT can handle	640 Hz	425 Hz	320 Hz	200 Hz	110 Hz

Problematic for handling pp 500 GeV collisions.
DAQ rate > 1000 Hz (peak at 2 KHz)

Future Developments

- R & D for including HFT hits. GPU for secondary vertex finding

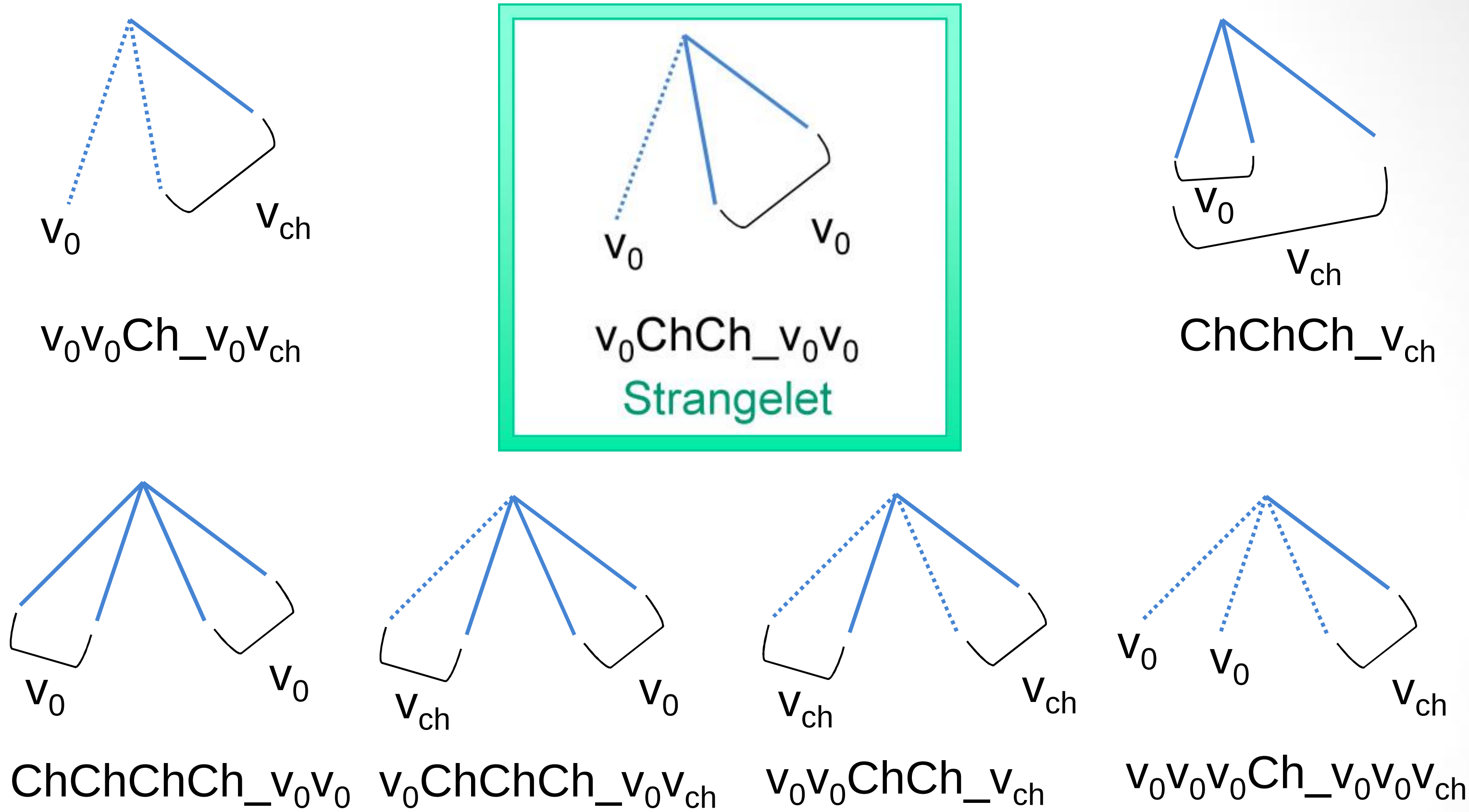
- **Explore and possible adopt CA tracker**

- Improve the MC environment for the HLT

- Offline support (HLT infor. in MuDst etc.)

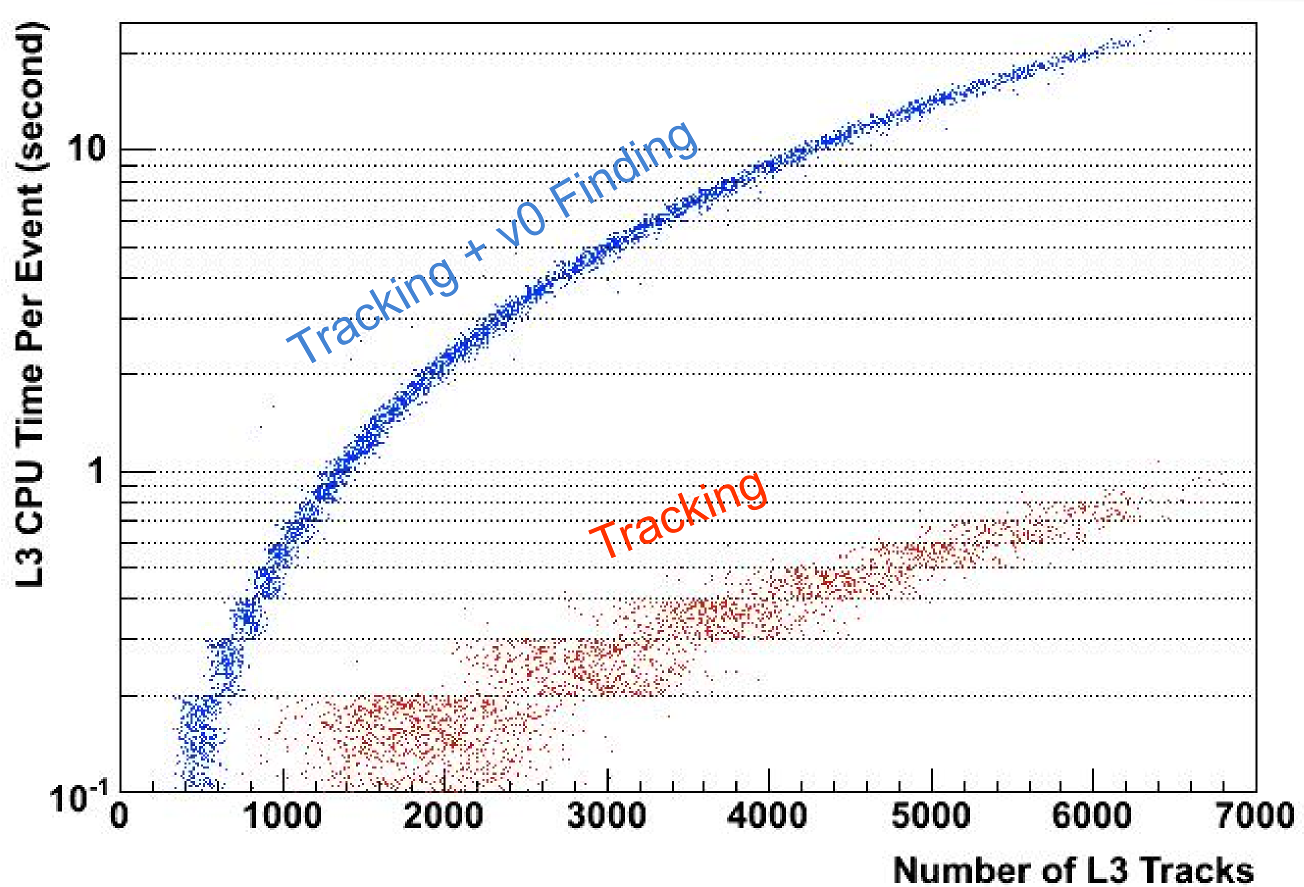
- Consolidate the online calibration

Trigger on Secondary vertice : Search for strangelets and other exotics



Good potential for new discoveries (Strangelets, di-Ω etc.)
In the future we will upgrade GL3 to trigger on secondary vertice.

Secondary Vertex Finder



v_0 reconstruction is CPU intensive ($\sim M^2$).

Secondary Vertex Finder with GPU

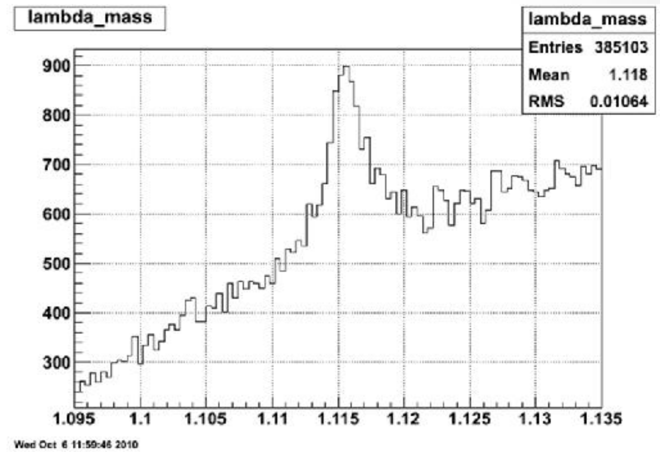
Test result:

GTX280 VS 2.8CPU

	CPU	GPU(GeForce GTX 280)
clock	2.80GHz	1.3GHz
Time cost	93us/pair	1.3us/pair

GPU is 60 times faster than single CPU core considering data transmission.

GPU is 120 times faster than single CPU core for nude calculation.



Lambda reconstructed by GPU (real data, HLT tracks)

GPU significantly accelerate v_0 reconstruction.

Exploring adopting CA tracker

Ivan Kisel, GSI — STAR Tracking Review Meeting, Berkeley, November 14, 2011

FOR INFORMATION ONLY

Potential Speed-Up

Efficiency and ratios, %		
	Aug 2010	Dec 2010
Ref Set	96.7	96.6
All Set	88.6	88.6
Clone	9.9	10.6
Ghost	29.1	12.6
Tracks/ev	660	659
Time/ev, ms	178	47

Aug 2010 - STAR version
Dec 2010 - GSI local version:
- code optimization
- no need to extrapolate
- time optimization

Reconstructable track:
Number of MC hits ≥ 10

All set: $p \geq 0.05$ GeV/c
Reference set: $p \geq 1$ GeV/c
Ghost: purity < 90%

Residuals and resolutions		
	Aug 2010	Dec 2010
y, mm	0.50	0.48
z, mm	0.96	0.92
sin φ, 10^{-3}	4.7	4.5
dz/ds, 10^{-3}	6.1	5.6
p_t, %	2.6	2.2

$\mathrm{tg}\,\varphi = dy/dx$
$ds^2 = dx^2 + dy^2$

lxir039.gsi.de
2 Intel Xeon X5550
8 cores per CPU, HT, 2.7 GHz
(only a single core performance)

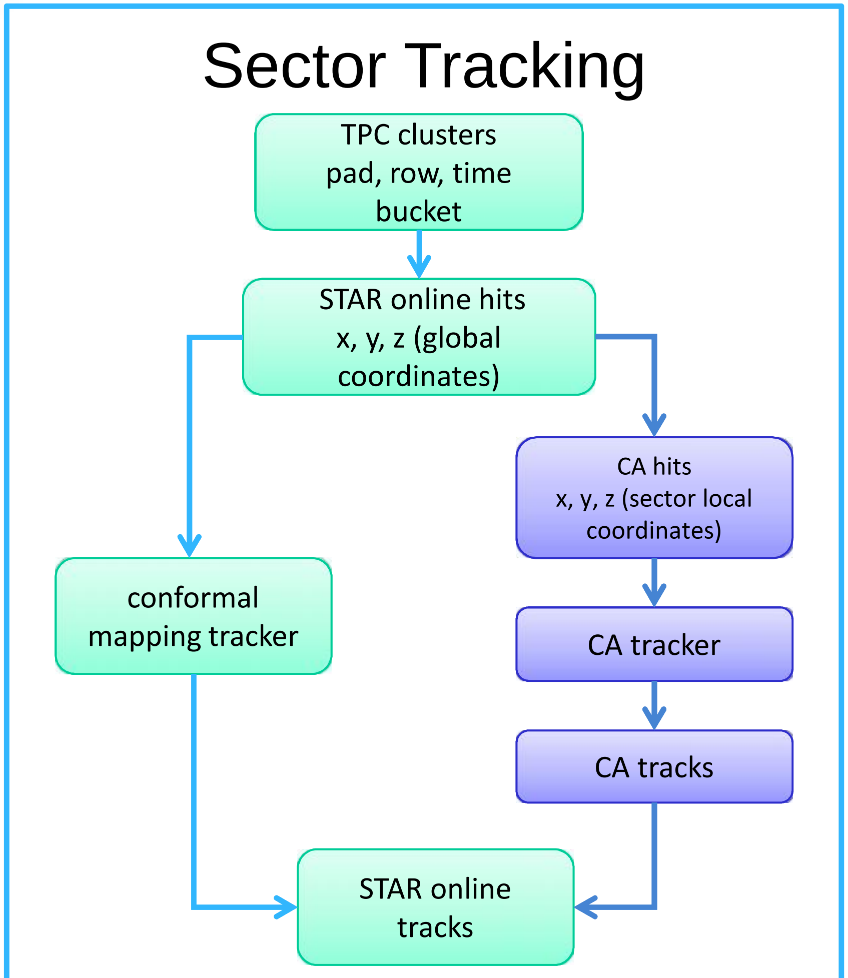
Au+Au 200 AGeV; 100 MC events
(CA in the stand-alone mode)

The STAR TPC CA track finder takes 47 ms/event (a requirement for HLT is 50 ms/event).
Both versions of the CA algorithm are SIMDized. A non-optimized ArBB version exists.

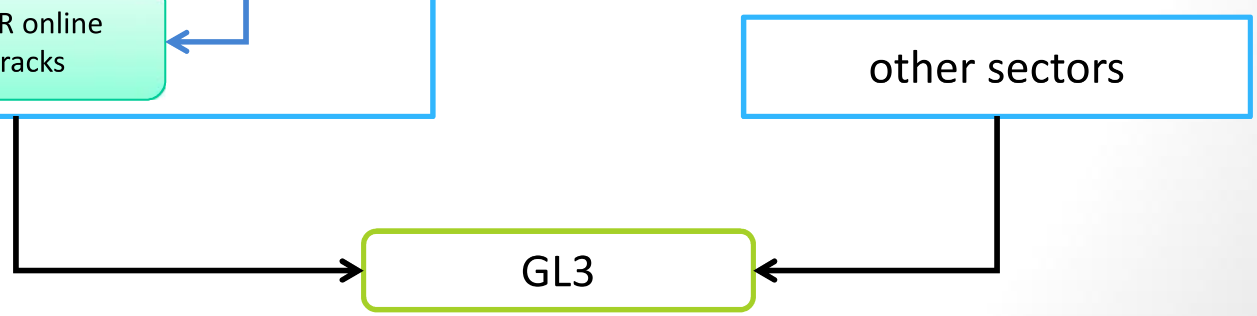
10

Compare to current STAR HLT tracker:
Same speed, better efficiency, easy for future parallelization.

Exploring adopting CA tracker



- Treat core tracking routine as a black-box.
- CA tracker is vectorized, if CPU supports SIMD, i.e. SSE, instructions.
- CA tracker can run in multi-thread mode, if Intel Thread Building Blocks is available.
- CA tracker has a GPU version
- We run a whole CA tracker for each sector, but feed only hits from a single sector.



Working in progress.

(HLT summary)

- STAR HLT has successfully selected events of interests and sent them to express streams.
- It is demonstrated that we can deliver important physics fast with the HLT
- Future development is presented. In particular we are working with experts to implement CA tracker in STAR HLT.

SUMMARIES

Summary

- STAR – an experiment with a challenging Physics program with many changes
  - Physics objectives, luminosity, …
  - R&D, …
- STAR framework
  - Has survived 12 years and we intend to make it morph and survive much longer
    - Much work is to be done – but changes are happening while physics continues, R&D can proceed, and STAR → eSTAR → eRHIC
  - First wave of new component review – thumb up (with findings to address) of 2 out of 3 now being integrated without disruptions
  - CA activities strongly suggested to move to HLT
- STAR HLT a success so far
  - More challenges are ahead due to luminosity increase, fast DAQ rates but slower HLT
  - CA speed up numbers tend to indicate it would do for STAR outer years
- HLT / tracking workshops
  - Have been of a net benefits to STAR – CA generally approved, perhaps vertexing
  - STAR could / would help in discussions of a path forward to
    - Make the package and algorithm portable, plug-and-play (to the extend possible)
    - Principles for long term survival important (not content)
    - We need to discuss code preservation and control, documentation, interface, data structures, coding standards and other "mundane" issues, yet core to preservation