1.1. Software starts in the next page

1.2. Software

This section contains the description of the software elements required for the successful processing and analysis of the acquired raw HFT data. Since the HFT is an Upgrade Detector of an existing experiment (STAR) its software needs need to be incorporated into the existing software and computing environment of the experiment. After a brief discussion of STAR's environment, we list and describe the online, offline and simulation modules and tools that are required to be developed for the HFT needs, and we will finish with a discussion of resources and institutional software responsibilities and commitments.

1.2.1. STAR's Software Environment

The STAR software environment comprises of a set of tools (development, simulation, production and analysis environment), mainly in the form of plug-in software modules in a ROOT – based backbone interface. At the same time it provides the coding standards and the data model for new module development and integration in the top-level shell scripts. Each detector sub-system is responsible for the development of all modules necessary for its successful operation. This work is coordinated with the rest of the experiment through a designated software representative from the group. At the same time there is a software infrastructure group at Brookhaven National Lab (aka BNL core), which is maintaining critical pieces of code (e.g. tracking, calibrations, databases) and also provides help with the integration of new software in the system.

Online Environment

The Online software primarily ensures the data integrity during data acquisition via appropriate detector monitoring and sample event reconstruction. Beyond these basic but important tasks, and as computer processing capabilities improve dramatically, more and more formerly offline tasks move to the online environment. One such task is the hit finding in the STAR TPC and discussion has started on the possibility for online (pre-) tracking in the TPC. This is of particular interest to this group that plans for on chip Pixel clustering and hit finding, especially in the area of triggering.

Offline Environment

The Offline environment consists of the event reconstruction software packages. This starts with the raw data as input and through proper calibrations it proceeds with detector cluster/hit finding, integrated tracking, event vertex finding and event information writing on DSTs.

1.2.2. Online Software

[Need a couple of paragraphs from LBL/MIT and/or a pointer to an appropriate section in this document]

1.2.3. Offline Software

Hit Reconstruction

The Cluster/Hit finder is the first piece of code applied to the pedestal subtracted raw information from the IST and PIXEL detectors.

In **IST** this will be a standard search for all fired strip-lets, i.e. all strips with a pedestal subtracted ADC value above a cut/threshold value. This is a common practice in silicon strip detectors and the MIT group, which will build the detector, has extensive experience. [Gerrit should read/add/update this] Each fired strip is then first going to be transformed into a set of local/wafer coordinates based on its position number on the wafer. This will be followed by a transformation to STAR global coordinate system (detector hit information needs to be saved in global coordinates), which is usually done via a series of partial transformations (wafer to ladder, ladder to shell, shell to detector, detector to STAR).

In the PIXEL detector the first steps (Cluster/Hit finding) are planned to be incorporated on the chip's logic and are done online during data acquisition, as discussed in ??? The local to global transformation process is identical to the IST even though the partial transformations will be different in order to incorporate the specific geometry and the specific hardware-implemented alignment features of the PIXEL detector.

Tracking

The current STAR reconstruction environment provides a Kalman-filter based integrated tracker. This tool is in principle ready to accommodate and integrate the IST and PIXEL hits, with their proper error/weights, in its environment. In reality work and close collaboration with the BNL-core group will be required to tune the tracker's parameters in order to properly deal with the high precision information coming from the PIXEL layers. At the same time there is a need to develop methods to deal with the path ambiguities in the SSD and IST (extended strip 'hits'), as well as dealing with the ghosting in the PIXEL detector due to out of time events in a high luminosity environment. Dealing with the latter two problems of tracking/ghosting will require studies that use a several-passes tracking approach and/or knowledge of the triggered event vertex obtained from a first-pass (or quick) vertex finder.

Event Vertex Reconstruction

Currently STAR deploys two different event vertex finders during event reconstruction, one for heavy ion and one for proton-proton collisions. Each of them is specifically tuned to perform best in these completely different environments (high multiplicity w/out pile up in heavy ions and low multiplicity with high pile-up in p-p). In a typical heavy ion collision the TPC has to cope with events of relatively large multiplicity and virtually pile-up free whereas in p-p one needs to extract the primary vertex from a few primary tracks surrounded by a thousand of out of time (pile-up) tracks. As a direct consequence of this fact STAR uses a Minuit-based event vertex fitter (with a seed finder) in heavy ion collisions. For p-p the vertex fitting procedure is based on a chi square minimization method but, most importantly, the information from fast detectors is used to select (tag) the tracks that belong to the triggered event.

In the RHIC-II era's increased luminosity there is going to be significant pile-up not only in the TPC but also in the PIXEL layers of HFT due to the relatively large integration time of the detector (see section ??? and also Pileup discussion below). This will be the case in both p-p and heavy ion collisions. The SSD and IST are assumed to be pile-up free even for the highest rate p-p collisions. The presence of these two (as well as the other) fast detectors will require a new, revised version of the vertex finder that will combine the best features of both current finders. We do not anticipate the need for any new functionality, just the need for tuning and QA-ing the new/combined finder.

We should note here that the mid-term plans of the reconstruction infrastructure group include a Kalman filter-type vertex finder, which at the same time will do the primary track fitting. In high multiplicity events (>30 tracks or so) it is ok if one does the vertex finding/fitting and primary track fitting (i.e. fitting global tracks with the vertex as an extra point on track for tracks with DCA within 3cm from the vertex) in two separate steps. In low multiplicities it is generally better if one performs a simultaneous fit of primary tracks and event vertex. This should be worth exploring. Let us remember that the larger fraction of the secondary/decay vertices we are trying to resolve are in the range of 10-100 microns and any improvement in determining the event vertex (the most important single reference point in the event) is indispensable.

Secondary/Decay Vertex Reconstruction

The reconstruction of short-lived particles in a collider environment is an extremely challenging task. The key measurements of HFT involve the reconstruction of **D**- and **B**-mesons with typical cτ in the range of 120 – 500 microns, and Λ_C with a cτ of 60 microns. The lack of a Lorentz boost typically results in mean decay distances of about half the cτ, for decays at midrapidity of a properly pt weighted sample. For example, the mean decay distance of D0 mesons (cτ of 120 microns) at midrapidity is 60 microns. This environment demands the highest level of sophistication in the methods used to reconstruct the decay/secondary vertices.

Up to now, the STAR reconstruction code had to deal with decay vertices of strange particles, typically in the few centimeters range. For those distances, crude, fixed value cuts were sufficient. Only a recent effort to reconstruct D-mesons with the SVT, the first generation silicon vertex detector in STAR, started using decay vertex fitting techniques using the full information of a track, on a track-by-track basis (sometimes also called μVertex-ing). This work, currently still under development, should and will be the basis of the modules deployed on the HFT data. These important software modules are to be developed, as they are a key piece of the new software.

Databases – Calibration and Alignment

The accurate recording of the state and the position of the detector inside the STAR apparatus is of outmost importance as it directly impacts its performance.

[Database and Calibration to consult with other chapters before filling]

The task of Alignment is a very demanding one especially for the PIXEL detector where one would like to perform/know the positioning of the detector elements with offsets and tolerances to within a couple of microns.

The alignment of IST is not a challenging task provided good survey data has been collected of the detector's elements beforehand. The in-situ alignment will be done with software techniques (global and/or local alignment) and there is previous experience on this in the collaboration. All it is required is to bring the PIXEL hits within the (TPC+SSD+IST) track projection errors to the pixel layer, typically around 100 microns or so. Global alignment techniques usually yield results around 10 microns with a set of a few hundred thousand tracks. Rotations are also typically kept to a fraction of milli-radian.

In the PIXEL detector this task is more difficult and the designers of the detector decided early on to incorporate 'hardware' techniques in order to minimize element displacement in-situ. The pixels are designed with a 20 microns 'envelope' error, i.e. maximum allowed displacement in-situ. To achieve this various sophisticated methods have been developed, e.g. interlocking, easily replaceable shells of extreme precision on-bench survey data. Details on the method and the specific hardware implementation can be found in section ???.

Despite this excellent 'hardware pre-alignment' software methods will have to be deployed in order to both check and fine-tune the in-situ information of the detector elements. There are two categories of software alignment techniques, the so-called *Global* and *Local* alignment.

The *Global* alignment uses TPC track information on a statistical basis in order to obtain systematic silicon detector rotations and shifts. Typically a 'rigid body' model is been applied (i.e.\ ignoring possible ladder twists, sagging effects and wafer non-planarities) and a misalignment model is introduced. Then a Taylor expansion with respect to misalignment parameters (3-D shifts and 3-D rotations) is performed looking for deviations of measured hit position from predicted primary track position on a measurement (wafer) plane. The track prediction comes from the detector(s) used as reference, e.g.\ initially the TPC alone, and later the combined TPC+SSD (+IST) tracking. In the next step, from the hit deviations distribution, a misalignment parameter has been calculated as a slope with a straight line fit. A global least-squares fit is also simultaneously performed on all available information. The method is applied iteratively until the fitted parameters reach stability. This global method was first applied to TPC+SSD+SVT data in STAR and it is well understood. It will serve us for the IST alignment but it will need modifications for the PIXEL detector. This is because the Pixel elements (wafers) on a ladder will have deviations from the 'flat plane' hypothesis.

In a *Local* or *Self*-alignment method one aims at the most precise *relative* placement of the detector elements. In this procedure only high precision hit information is used coming exclusively from the detector under local alignment. A successful method was developed and tested on simulations by the BNL-core group and this should be further developed into a working module for HFT.

1.2.4. Simulation Framework

The current simulation framework in STAR is GEANT 3.0 based with custom script extensions to facilitate detector geometry implementation and event generation. It also includes event generators like HIJING, PYTHIA, phasespace etc. that are interfaced to geant. This framework is soon to be abandoned for a ROOT-based geometry and tracking package (VMC, virtual monte carlo). Nevertheless, we are still using it and we will continue to do so in the immediate future. The tasks, and therefore software modules one needs to develop here are: a) the detector geometry definition, b) the detector response packages (fast and slow simulators), c) track embedding in real/raw events, d) a hit pileup handler, e) the Association Maker and structures for evaluation purposes, and f) Analysis code (performance, physics etc) capable of handling and evaluating the resulting information. Our group will have to contribute modules and effort in all these categories.

Detector Geometry Definition

This task is to include in the simulated apparatus of the experiment the latest and most accurate/realistic (actual) geometry of HFT (IST and PIXEL), since this is the only way to ensure reliability of the resulting efficiency numbers. This is also the place where the active areas of the detector, the hit information and the global positioning matrices of the detector are defined.

Detector Response Simulators

The detector response simulation packages in STAR reside outside the Geant framework. They are actually invoked at the event reconstruction step. Typically there are two or three categories of response simulators: a) *Fast simulators*, which smear the hit position coordinates and assign hit uncertainties based on parameterized analytical functions, b) *Slow simulators,* which mix raw and simulated hits at the ADC level (the latter usually obtained from sampling of parameterized response functions, and c) *Very Slow simulators*, which track individual electrons through the detector body; from their generation to the readout. This is usually very time consuming and one utilizes this method only in small scale productions in order to determine the functions used in the first two methods.

A Slow Simulator for the IST detector

[Need a couple of paragraphs from Gerrit]

A Slow Simulator for the PIXEL detector

The detailed simulation for STAR Heavy Flavor Track PIXEL silicon detector consists of 4 steps. First, use the information of a charged particle passing through the PIXEL as inputs. The information contains the particle momentum, incident direction, path length in the PIXEL, and the sum of electron-hole pairs it generates. The total number of electrons generated from charged track passing through the silicon sensor is calculated using Bichsel distribution[1]. Second, build the geometry of the detector: one chip of 640 x 640 PIXEL array. One PIXEL is 30um x 50um x 30um, consist of four different layers from top to bottom: Readout electronics layer, diode layer, epitaxial layer and substrate layer. Third, simulate the transportation of electrons generated in the PIXEL^2: diffusion, recombination and reflection at interfaces between different layers. A Gaussian equation is used to describe the diffusion as a random walk process. The electron recombination rate is dependent on the different doping density of different layers. Finally, calculate the distribution of electrons collected in the PIXEL array as output signal. The left panel of Figure 1 presents the simulated pixel cluster shape from 1GeV charged pion incident at 45 degree angle. The right panel of the figure shows the comparison of the deposited number of electron profile from data [2] and simulation. The two results agree with each other very well. The major problem for this slow simulator is the speed. It takes about 20 minutes to simulate a single charged track and it comes mainly from simulating the electron diffusion process. This is too slow to be used in future large scale simulation studies.

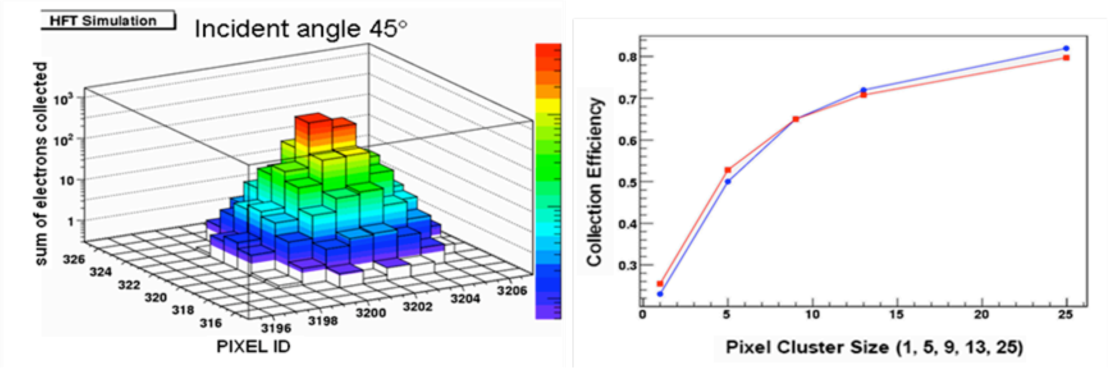

Figure 1: (Left) distribution of number of deposited electrons on pixels from a single charged pion with 45 degree incident angle from the slow simulator; (right) profile of the fraction of deposited number of electrons from simulation (blue) and data [2] (red).

To significantly improve the speed while keeping good accuracy, we developed a simplified method. Instead of simulating diffusion process step by step for each single electron, we calculate the probability distribution function for each electron in a specific space location to be collected by different pixels. Since any electron generated in the PIXEL is independent from each other, by randomly sampling this probability distribution function, we can decide which pixel collected this electron or if the electron recombined before being collected. Following above steps, we collect the pixel IDs that absorb all electrons along a charged track and add them up to obtain the number of electrons deposited in each pixel. To implement this method, we built a fine 3-D grid in a single pixel and calculated the probability distribution function for electron produced from all grid points using the slow simulator. For any one electron from incident charged track, we directly use the probability distribution function for the grid point that is closest to its production point to determine the pixel ID that collected this electron. Since all PIXELs are identical, we only need to make coordinate transformation if any electron is produced outside the pixel where the grid is built and repeat the same operation to finish the whole simulation for a charged track. The speed of the simplified simulator is a few seconds per charged track. The accuracy depends on the granularity of the grid and can be very good with high granularity. However this speed is still too slow to be used in simulation a central Au+Au collision which generates a few thousands of charged tracks. We are developing the third version of the simulator aiming to increase the speed by two orders of magnitude while keeping good accuracy.

Embedding and Pile-Up

The embedding of simulated tracks into the raw data stream (which provides the best 'background environment' for track/particle reconstruction) has been around in the heavy ion community for about fifteen years.

[give a paragraph's description here]

IST Embedding

[need a couple of paragraphs from Gerrit]

PIXEL Pile-up simulation

To simulate the PIXEL pile-up hits, we produced one standalone ROOT file containing only the required PIXEL hits multiplicity in both inner and outer layers. The hits are produced from GEANT using the same setups as in the production for CD0 simulation. The hits in this file are then merged with the PIXEL hits in every central collision that is pushed through STAR reconstruction software.

According to CD0 proposal, the pileup hits density for 1x RHIC-II luminosity is 43/cm^2 on the inner PIXEL layer and 6/cm^2 on outer PIXEL layer. The pile-up hits densities under different assumption on the RHIC-II luminosity are listed in the following table. For each of the assumed luminosity, we produced one pile-up hits file.

luminosity	Inner later pile-up hits density	Inner later number of pile-up hits	outer later pile-up hits density	outer later number of pile-up hits
0.5xRHIC-II	21/cm2	6600	3/cm2	2638
1xRHIC-II	43/cm2	13514	6/cm2	5276
2xRHIC-II	86/cm2	27000	12/cm2	10550
3xRHIC-II	129/cm2	40500	18cm2	15826

This method neglects the fluctuation of pile-up hits density in different location of PIXEL detector since for every event, only one set of pile-up hits is applied. In the future, we plan to directly apply white noise on the PIXEL detector before reconstruction happens. This will include the local density variation and is an improvement to the old method.

Association Makers

Analysis of Simulated Data

1.2.5. Institutional Responsibilities

Institutional commitments

Resources required

Institutional Responsibilities

Software task		BNL	UCLA	KSU	NPI	MIT	LBL	Purdue	
Online									
	IST					X			?
	Pixel						X	X	?
Offline									
Hit Reconstr.	IST					X			
	Pixel						X	X	?
Tracking		X							?
Event Vertex		X		X	X				
Decay Vertex		X		X	X				
Calibration Db	IST					X			?
	Pixel						X	X	
Alignment	IST	X		X		X			
	Pixel	X		X			X	X	
Simulation									
Geometry	IST	X				X			
	Pixel	X					X	X	
Fast/Slow Sim.	IST					X			
	Pixel						X	X	
Embed./Pileup	IST					X			
	Pixel						X	X	
Analysis									
Charm			X	X	X		X	X	
Bottom			X				X	X	
Λ_C				X	X		X		?
Spin							X		

Other collaboration contributed resources

---

[1] H. Bichsel, *Review in Modern Physics*, vol.60, pp. 663, (1988).

[2] "Modeling, Design, and Analysis of Monolithic Charged particle Image Sensors" Shengdong Li, Ph.D thesis, Univ. of California, Irvine, 2007.