Talk on the Mashine Learning Workshop 7-9 December 2015

1. Pattern Recognition in HEP
2. Simulation for NICA T0-T1

Gennady Ososkov

Laboratory of Information Technologies
Joint Intstitute for Nuclear Research,
141980 Dubna, Russia
email: ososkov@jinr.ru

HEP data processing

Is accomplished on the following stages:

- **Pre-processing. It includes:**
  - **Data acquisition**
  - **Data Transformation: to correct from detector distortions and misalignment**
  - **Data selection: to remove noisy, inconsistent and other observations, which do not satisfy acceptance conditions**
- **Pattern recognition for event reconstruction: hit detection, tracking, vertex finding, revealing Cherenkov rings , fake objects removing**
- **Physical parameters estimation**
- **Physical hypothesis testing for invariant mass spectra**
- **Monte-Carlo simulations are used on all stages and allow to**

  - accomplish in advance the experimental design of a hardware setup and data handling algorithms and optimize them from money, materials and time point of view;

  - develop needed software framework and test it;

  - optimize structure and needed equipment of planned detectors minimizing costs, timing with a proposed efficiency and accuracy;

  - calculate in advance all needed distributions or thresholds for goodness-of-fit tests;
- **New inevitable issues of the WLCG and BigPanDa era**

  Parallel programming of optimized algorithms

  Grid-cloud technologies which changed considerably HEP data processing concept

The particular features of HEP data

- data arrive with extremely high rate;
- recognized patterns are discrete and have complex texture;
- very high multiplicity of objects (tracks, Cherenkov radiation rings, showers) to be recognized in each event;
- the number of background events, which are similar to “good” events, is larger than the number of the latter events by several orders of magnitude;
- noise counts are numerous and correlated.

The basic requirements to data processing in current experiments are: maximum speed of computing in combination with the highest attainable accuracy and high efficiency of methods of estimating physical parameters interesting for experimentalists.

Methods to achieve that

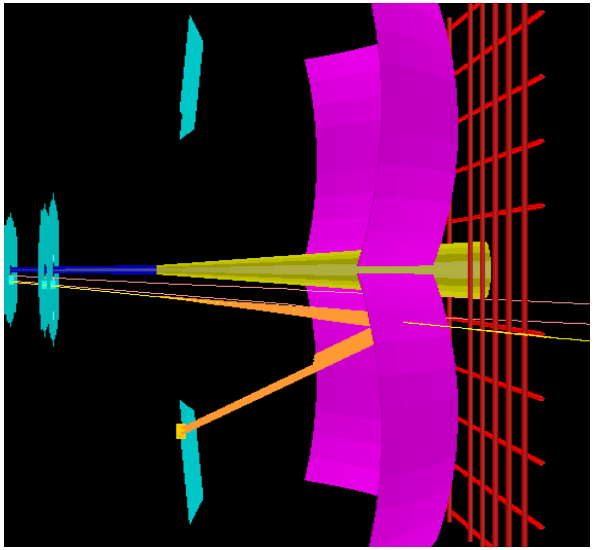
- Cluster analysis
- Hough transform
- Kalman filter
- Cellular automata
- robust approach based fast algorithms
- neural networks of both types:
- feed-forward and recurrent Hopfield networks
- wavelet analysis for data filtering, resonances and jet search
- Likelihood Ratio test and Boosted Decision Trees for hypothesis testing

Experimental examples:
1. CBM RICH detector

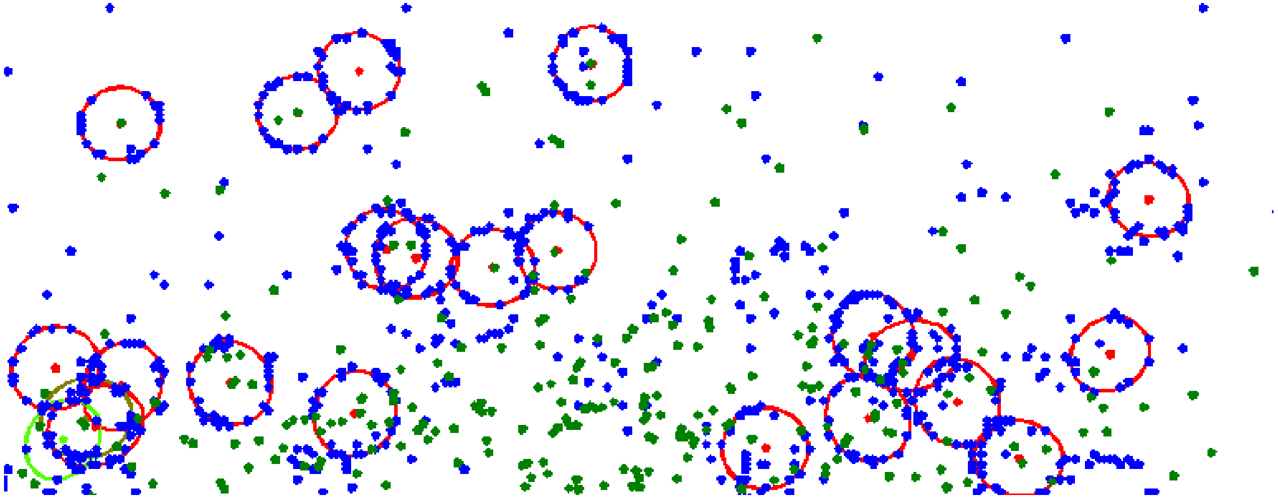
CBM experiment (Germany, GSI, to be running in 2018)
10^7 events per sec,
~1000 tracks per event
~100 numbers per track
Total: terabytes/sec !

No triggers!

All data processing on-line



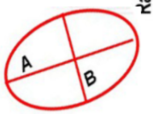
A sketch of the RICH detector

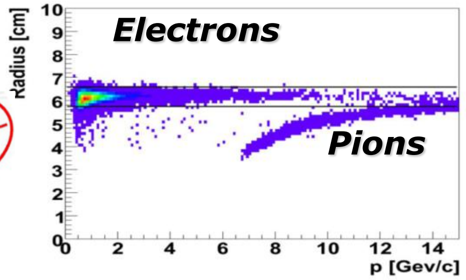


A fragment of photodetector plane.
In average there are 1200 points per event forming 75 rings.

Data processing stages:

- ❑ Ring recognition and their parameters evaluation;
- ❑ Compensating the optical distortions lead to elliptic shapes of rings (ellipse fitting);
- ❑ Matching found rings with tracks of particles which are interesting to physicists
- ❑ Eliminating fake rings which could lead to wrong physical conclusions (ring quality calculating as probability to have a good one)
- ❑ Accomplishing the particle identification with the fixed level of the ring recognition efficiency

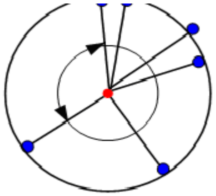




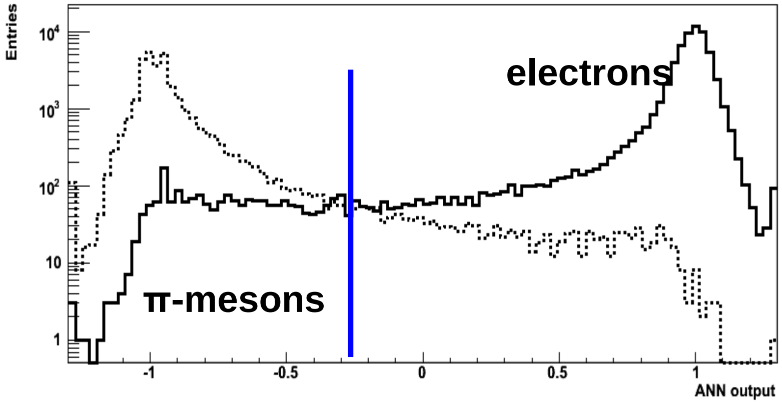
Radius vs momentum
for reconstructed rings.

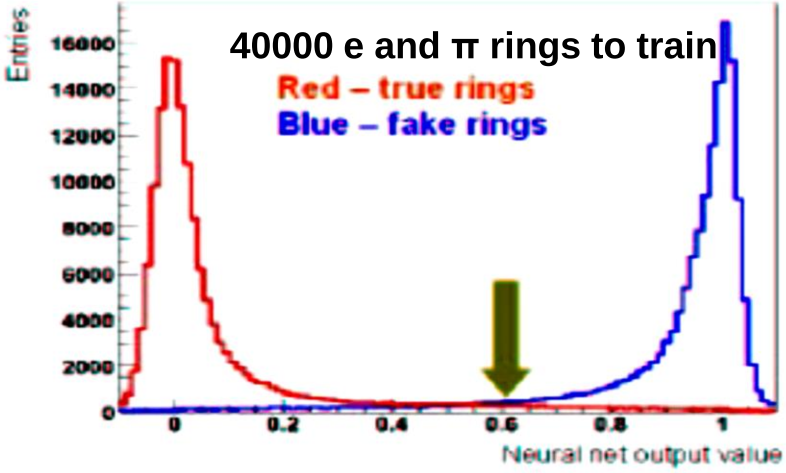
Feed-forward NN for CBM RICH detector

The study has been made to select the <u>most informative ring features</u> needed to **distinguish between good and fake rings and to <u>identify electrons</u>**.
Ten of them have been chosen to be input to ANNs, they are:



1.Number of points in the found ring
2.Its distance to the nearest track
3.The biggest angle between two neighbouring points
4.Number of points in the narrow corridor surrounding ring
5.Radial ring position on the photodetector plane
6.χ^2 of ellipse fitting
7.Both ellipse half-axes (A and B)
8.angle *φ* of the ellipse inclination to abscissa
9.track azimuth
10. track momentum





40000 e and π rings to train
Red – true rings
Blue – fake rings

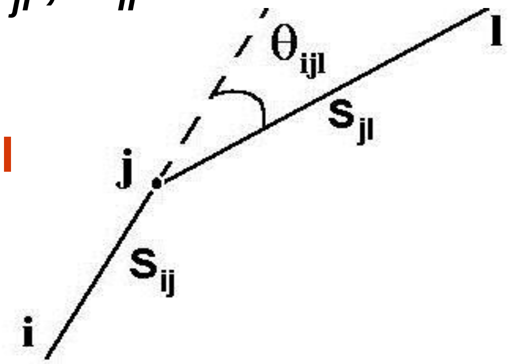
Two samples with 3000 e (+1)and 3000 π (-1) have been simulated to train NN. Electron recognition efficiency was fixed on 90% Probabilities of the 1-st kind error 0.018 and the 2-d kind errors 0.0004 correspondingly were obtained

Recurrent ANNs for tracking

Hopfield's theorem: the energy function

$$E(s) = -\tfrac{1}{2} \Sigma_{ij}\, s_i\, w_{ij}\, s_j$$

of a recurrent NN with the symmetrical weight matrix $w_{ij} = w_{ji}$, $w_{ii} = 0$
has local minima corresponding to NN stability points

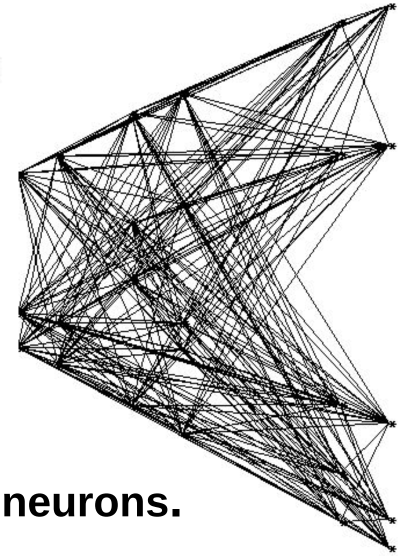


Track recognition by Denby- Peterson (1988) segment model
The idea: support adjacent segments with small angles between them. The special energy function:

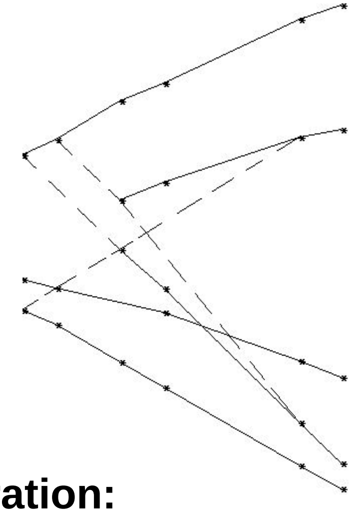
$E = E_{cost} + E_{constraint}$, where

$$E_{cost} = -\frac{1}{2} \sum_{ijkl} \delta_{jk} \frac{\cos^m \theta_{ijl}}{r_{ij} r_{jl}} v_{ij} v_{kl},$$

$E_{constraint}$ punishes segment bifurcations and balances between the number of active neurons and the number of experimental points.

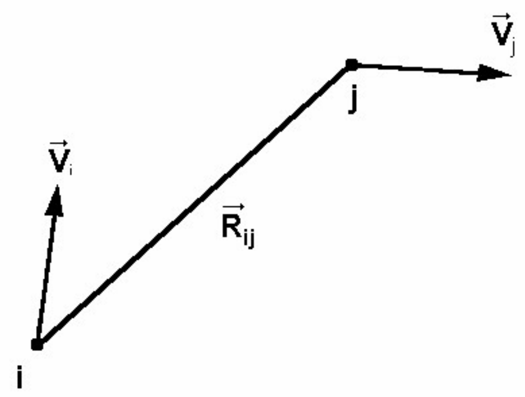


Zero iteration: 244 neurons.



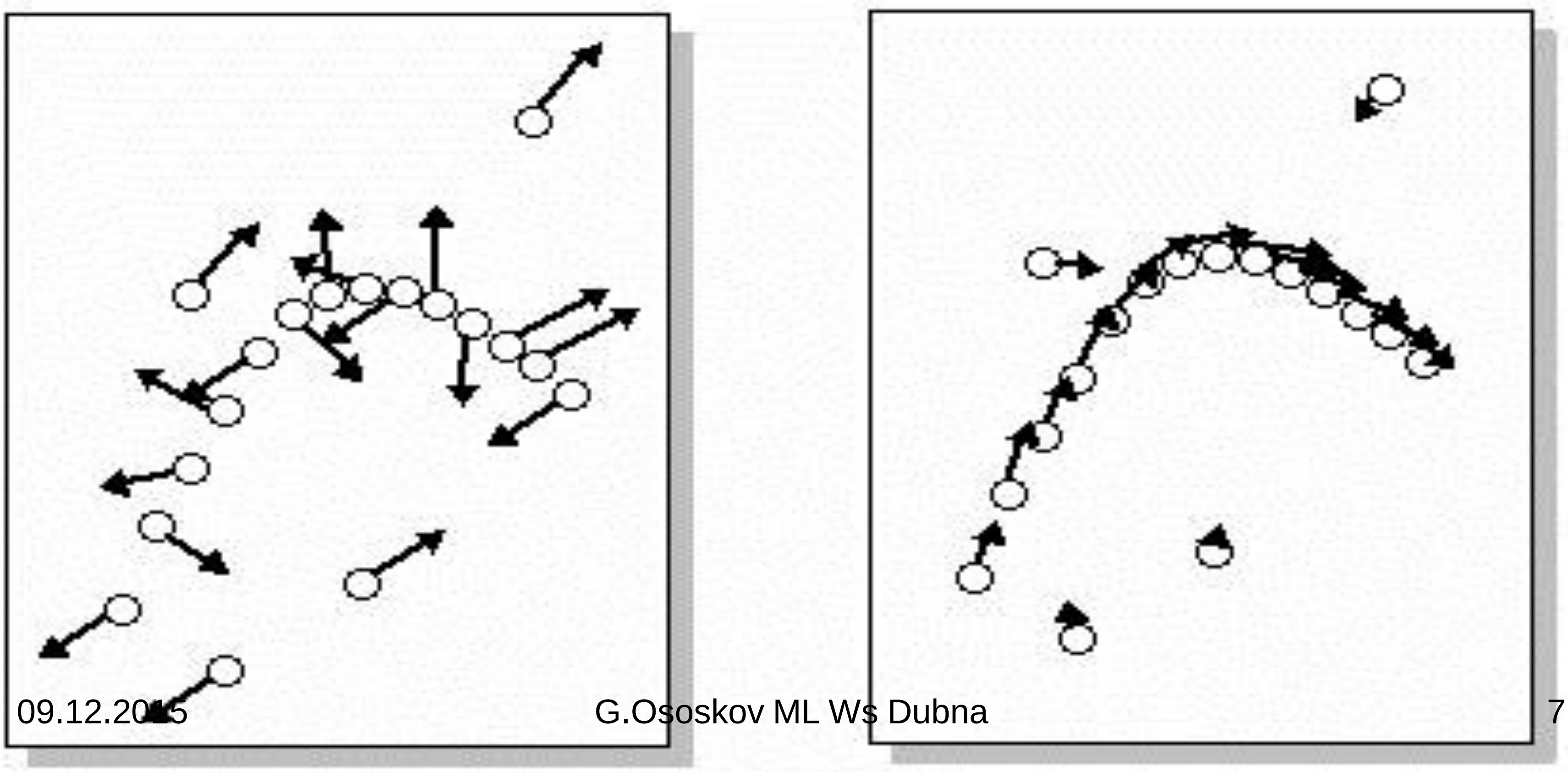
After 30 iteration:
26 neurons with v_{ij} >0.5

Track recognition by rotor models of Hopfield networks

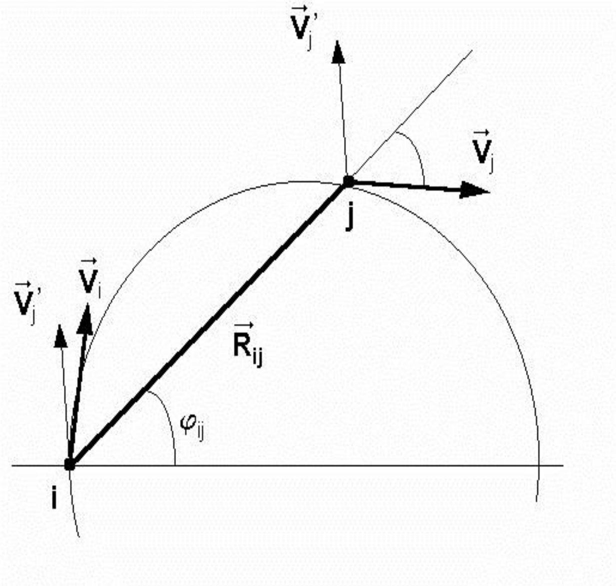


$$E = -\tfrac{1}{2}\sum_{ij}\frac{1}{\left|r_{ij}\right|^{m}} v_i v_j - \tfrac{1}{2}\alpha\sum_{ij}\frac{1}{\left|r_{ij}\right|^{m}}\left(v_i r_{ij}\right)^2$$

The energy function: the first term forces neighbouring rotors to be close to each other. The second term is in charge of the same between rotors and track-segments.



Our innovations



$$v'_j = \begin{pmatrix} \cos 2\phi_{ij} & \sin 2\phi_{ij} \\ \sin 2\phi_{ij} & -\cos 2\phi_{ij} \end{pmatrix} v_j = W_{ij} v_j$$

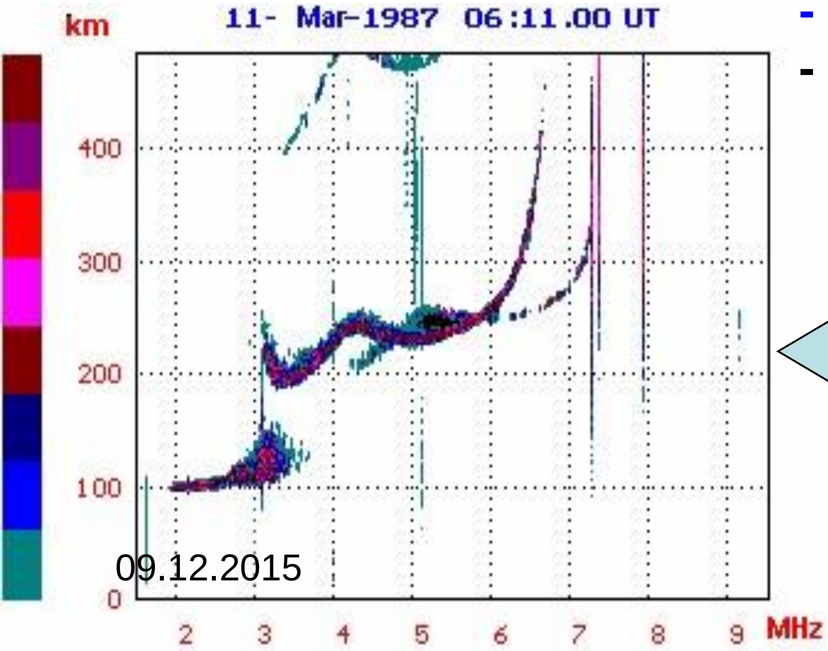
Therefore we obtain a simple energy function without any constrains

$$E = -\frac{1}{2}\sum_{ij} v_i \cdot v'_j$$

ARES experiment demanded extra efforts:

- **prefiltering by cellular automaton;**
- **local Hough algorithm for initial rotor set up;**
- **special robust multipliers for synaptic weights.**

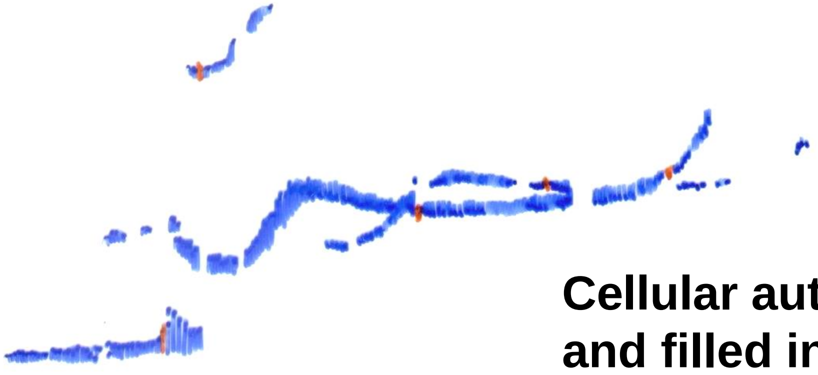
Results: recognition efficiency - 98%



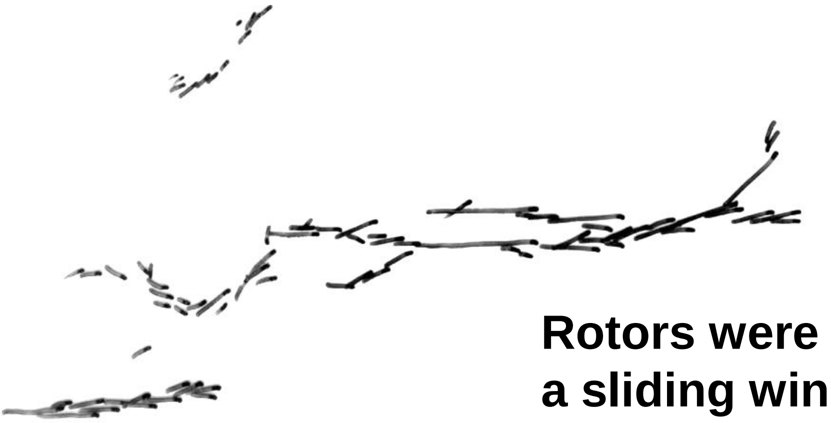
Analysis of ionograms.

Data from the vertical sounding of the ionosphere are measurements of frequency, amplitude, and arrival time of a signal reflected from an ionospheric layer.

Stages of ionogram processing:



Cellular automaton removed vertical noise and filled in missed hits



Rotors were initiliazed by angular histogramming in a sliding window which size was varied depending on the track curvature.

It decreased the neural net evolution up to 3-5 iterations

Recognition result

Up to now the corresponding program is in use in the Irkutsk Institute of the terrestrial magnetism, Russia and in the Lowell University, USA

Neural networks for elastic tracking

An idea was suggested 20 years ago to combine stages of a track recognition and fitting in one procedure when deformable templates (elastic arm) formed by equations of particle motion are all bended in space in order to overlaid the data from the detector.

Ohlsson and Peterson (O&P, 1992) from the Lund University realized this idea as a special Hopfield net with the energy function depending from helix parameters describing a track and binary neurons S_{ia}, each of them is equal to 1 or 0 when *i*-th point belongs or not to the *a*-th track, respectively.

Gyulassy and Harlander (G & H, 1991) proposed their elastic tracking that can physically be described as interaction between the positively charged template and negatively charged spatial points measured in the track. The better the elastic template fits points, the lower the energy of their interaction.

Using Lorenz potential with the time-dependent width $w(t) = b + (a - b)e^{-t/T}$, where *a* is the maximal distance, at which points are still accredited to this template, *b*<< *a* is spatial resolution of a detector, G & H obtained the energy to be minimized by helix parameters π

$$E(\pi, t) = -\frac{1}{N}\sum_{i}^{N} \frac{w^2(t)}{(\vec{z}_i - \vec{r}(\pi, \vec{x}))^2 + w^2(t)} .$$

Here, N is the number of points in the track, $\vec{z}_i$ and $\vec{r}$ are the *i*-th point measured in the space and its distance from the template, respectively, and is the set of the track parameters. When a detector is placed in a homogeneous magnetic field, the track is described by a helix with the curvature κ , helix angle λ, and phase Φ at the point $\vec{z}_i$, i.e.

$$\pi = \{\kappa, \lambda, \Phi, \vec{x}_o\}$$

Elastic neural networks applications

To avoid *E(π,t)* getting caught in local spurious minima the simulated annealing iterative procedure is applied. On the first iteration *w(t)* is taken for the highest temperature, when *E(π,t)* has the only one minimum. Then *w(t)* is narrowed gradually allowing more and more accurate search of the global minimum.

O&P elastic NNs after corresponding modifications were succesfully applied for Cherenkov ring search and track reconstructing (1997).
L. Muresan, R. Muresan, G.Ososkov, Yu. Panebratsev, *Deformable Templates for Circle Recognition*, JINR Rapid Comm. 1[81]-97, 27-44.
G&H elastic tracking was applied for the STAR TPC simulated data with remarkably high track-finding efficiency (1998) B.Lasiuk, D.Lyons, G.Ososkov, T.Ullrich, *Develeopment of an Elastic Tracking Package*, Proc. of CHEP'98, Chickago (1998).
Drift chamber tracks with their left-right ambiguity in magnetic field demanded to invent 2D neurons $S_i=(s_i^+, s_i^-)$ to determine a point to a track accreditation
Baginyan, S., Ososkov, G., Finding tracks detected by a drift tube system, CPC, v 108 (1998) 20-28.

__Important to note: 1. a homogeneous magnetic field of NICA-MPD project will make it possible to apply this elastic arm approach for MPD TPC tracking

2. Recurrent NN are self teaching and unlike feed-forward NN need many iterations to converge to a stable state. Thus their applications are not suitable for on-line tracking

Robust estimates for heavy contaminated samples

Real data are always contaminated. It means the crucial assumption of residual normality is violated! We must replace the Least Square functional $\Sigma_i \varepsilon_i^2$ by a functional $L(p,\sigma)=\Sigma_i w(\varepsilon_i)\varepsilon_i^2$ where ε_i are residuals and weight function $w(\varepsilon)$ is attributed to every point to be recalculated on each step of an iterative procedure.

Due to the lack of theoretical foundations for $w(\varepsilon)$ we have proposed

(1) for an important case of the uniform contamination

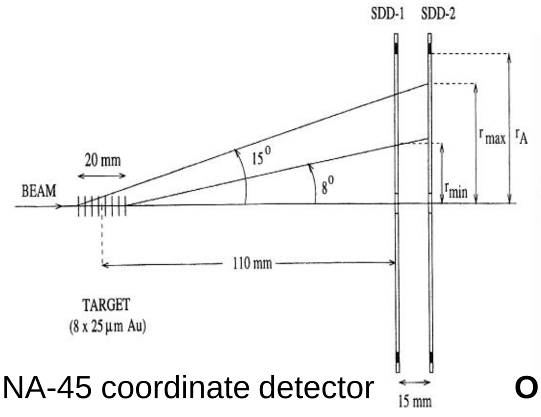
optimal weights $w(\varepsilon)$ $$w_{opt}(\varepsilon) = \frac{1 + c_{opt}}{1 + c_{opt} \cdot \exp\left(\frac{\varepsilon^2}{2}\right)}$$

parameter c_{opt} is determined by the contamination of data not in the whole range of the sample but within its essential part where all useful observations are concentrated

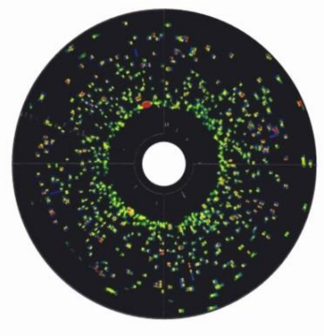
(2) For signals with known amplitudes **2D weights were proposed**

Robust fitting application examples

1. NA-45. Determination of the interaction vertex for only two coordinate planes



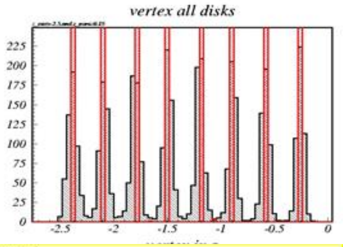
NA-45 coordinate detector



One of 2 silicon disks

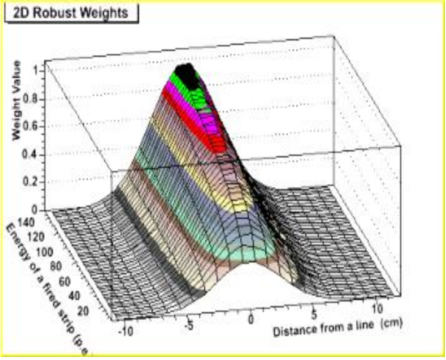
The target consists of eight 25-*μ* gold discs. 700 track events in narrow angular acceptance and large number of noise counts did not allow to recognize individual tracks.

Robust fitting iterations converge in five iterations, although initial approximation was roughly taken as the middle of the target region.

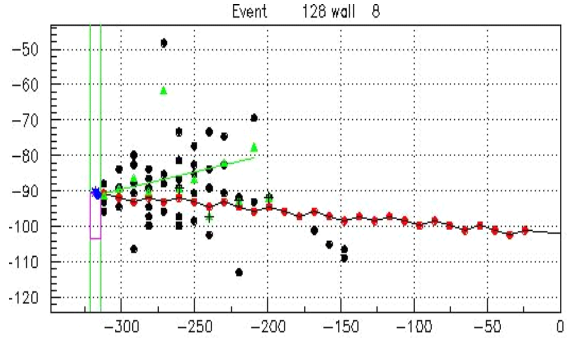


2. Opera. 2D weights for hadron showers and muon tracks.

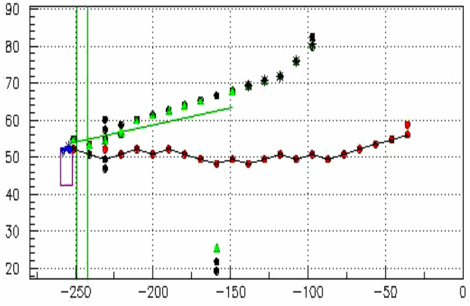
Fitting with 2D robust weights, which depend not only on distance of a point to the fitted track, but also on amplitudes of track hits



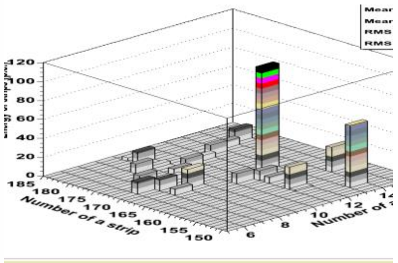
2D weight



hadron shower



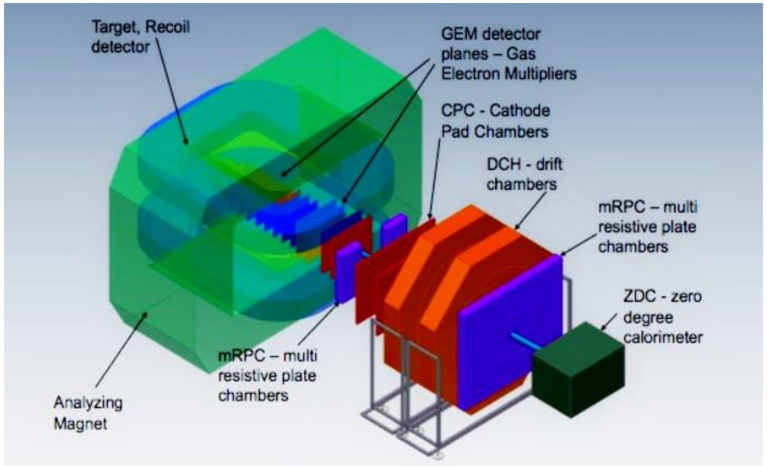
1D muon tracks



2D muon tracks

BM@N GEM tracking

coordinate transformation to simplify seed searching



Our problem is to reconstruct tracks registered by the GEM vertex detector With 12 GEM-stations inside the magnet

Kalman filter (KF) track following is the common procedure for track reconstruction

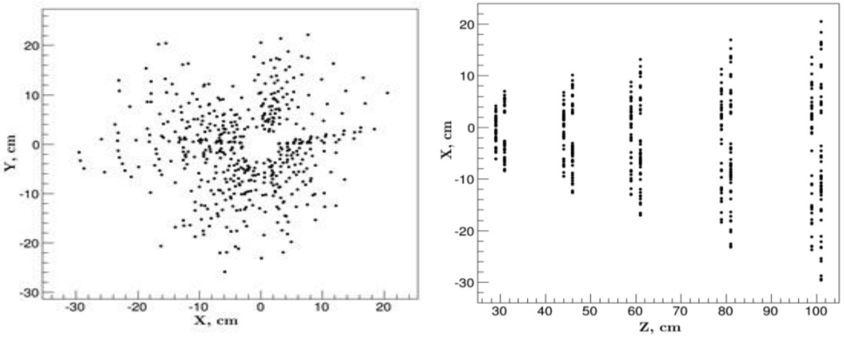
We use well-known LitTrack library from CbmRoot with **KM state vector**

$$\vec{x} = (x, y, t_x, t_y, q/p)^T$$

to be initialized from the available hit set

Therefore the problem of **track-candidate search to initialize KF** on BM@N GEM detector arises.

Even for 100 track event it impossible to recognize to which track a concrete hit belongs without checking it by some tracking procedure like Kalman filter



X,Y and Z,X views of a simulated 100 track event

Pictures are drown without fake simulation

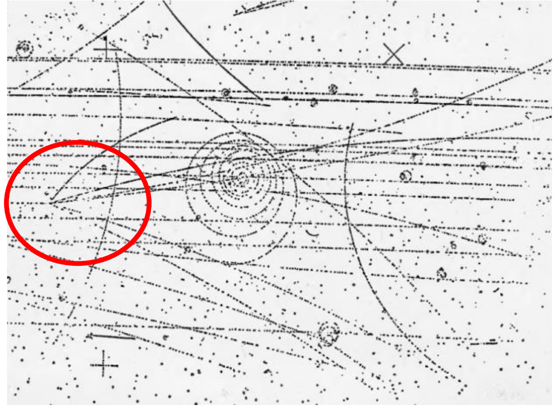
Our solution – to transform the coordinate system in order to obtain a space, where a track hits should be grouped more compact.

Do not forget about the main shortcoming of GEM strip detector: is the appearance of fake hits caused by extra spurious strip crossings

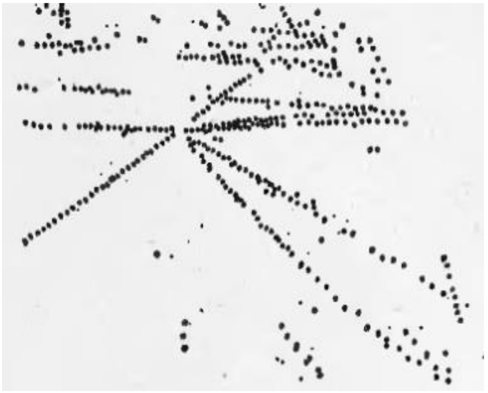
For n real hits one gains n^2- n fakes

Choice of coordinate transform

Prehistory: CERN 1968 2-meters hydrogen bubble chamber



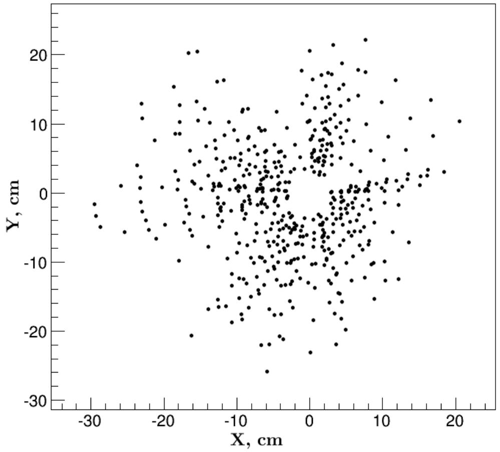
Picture of one projection output



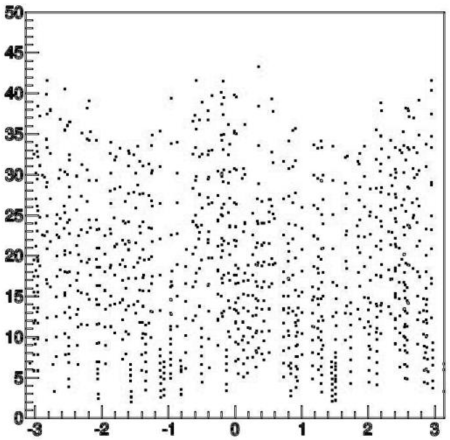
Spiral Reader

in polar coordinates (θ,ρ) with the pole at the event vertex

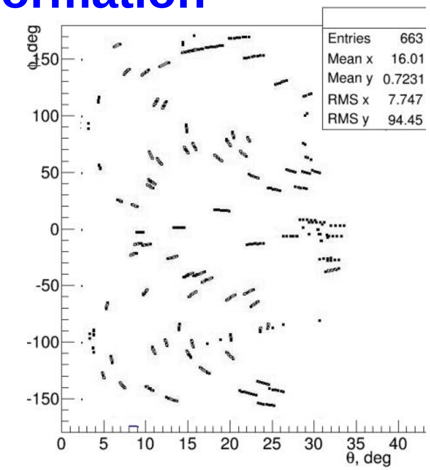
2014-15 Sergey Merts: looking for a suitable transformation



Original 100 tracks (X,Y)

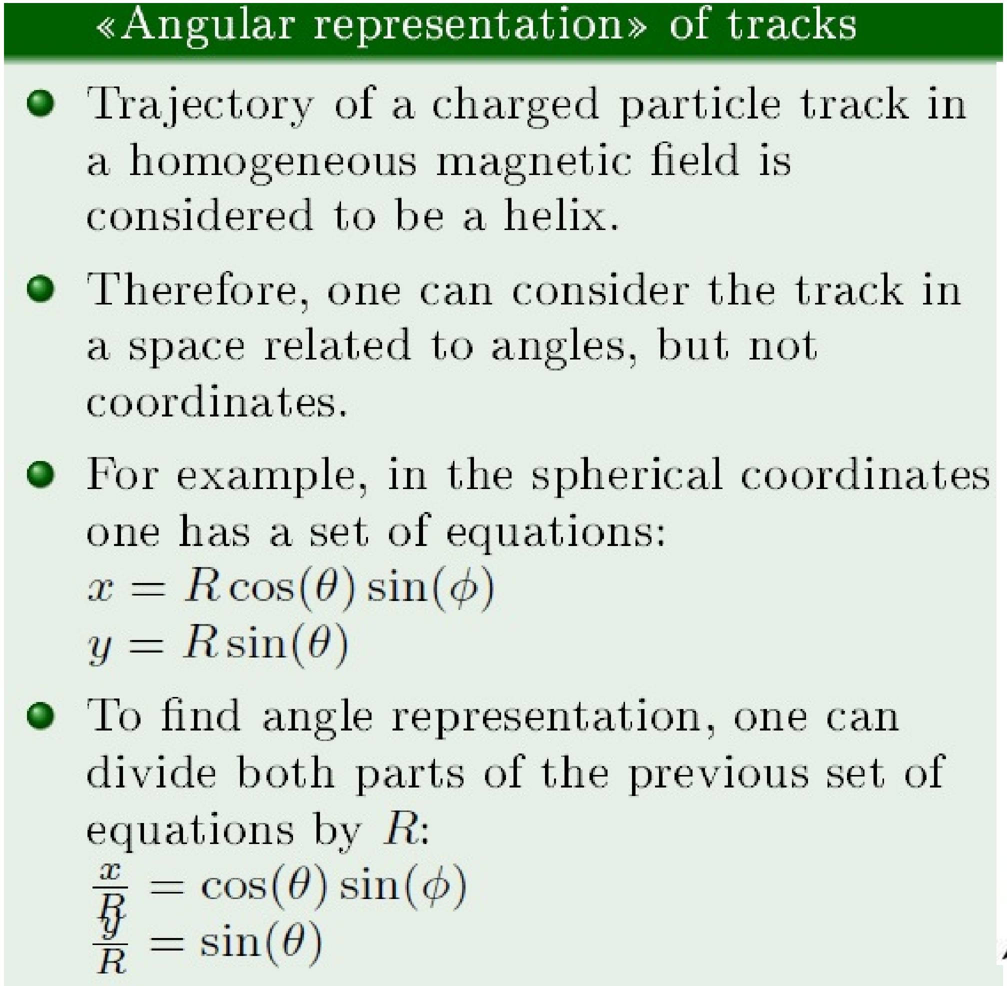


polar coordinates (θ,ρ)



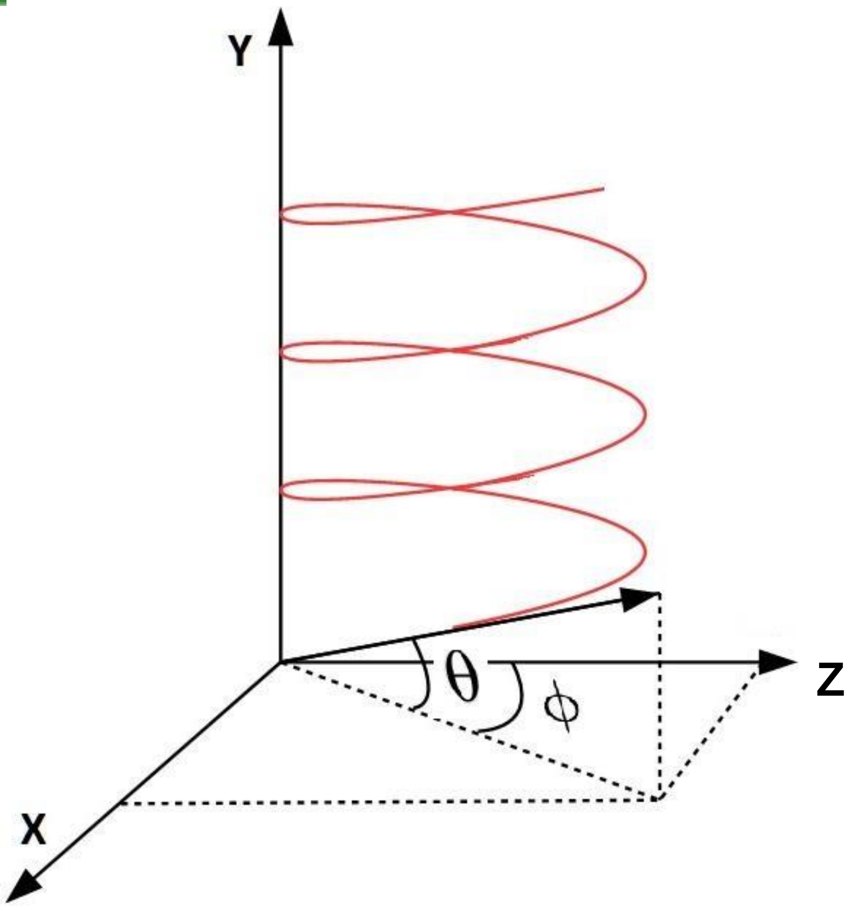
spherical coordinates (φ,θ)

Choice of coordinate transform 2



«Angular representation» of tracks

- Trajectory of a charged particle track in a homogeneous magnetic field is considered to be a helix.
- Therefore, one can consider the track in a space related to angles, but not coordinates.
- For example, in the spherical coordinates one has a set of equations:
  $x = R\cos(\theta)\sin(\phi)$
  $y = R\sin(\theta)$
- To find angle representation, one can divide both parts of the previous set of equations by R:
  $\frac{x}{R} = \cos(\theta)\sin(\phi)$
  $\frac{y}{R} = \sin(\theta)$



Results of transformation to {X/R,Y/R}

The coordinate transformation used:

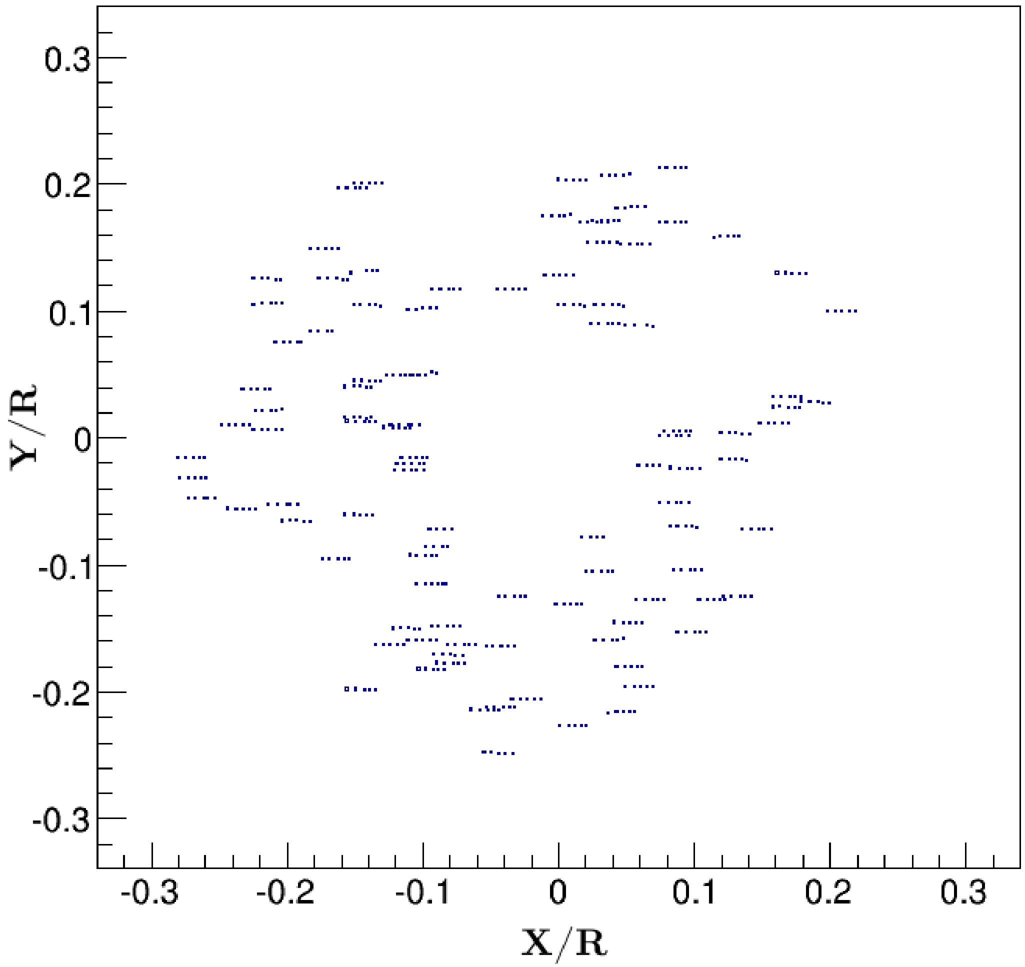
$$\{x, y\} \longmapsto \{\frac{x}{R}, \frac{y}{R}\}$$

$$R = \sqrt{x^2 + y^2 + z^2}$$

Properties

- Tracks are transformed into horizontal lines.
- Numeration of stations allows for searching for hits clearly.
- Distance between the hits pertaining to a horizontal cluster depends on momentum only.

Ideal hits without fakes

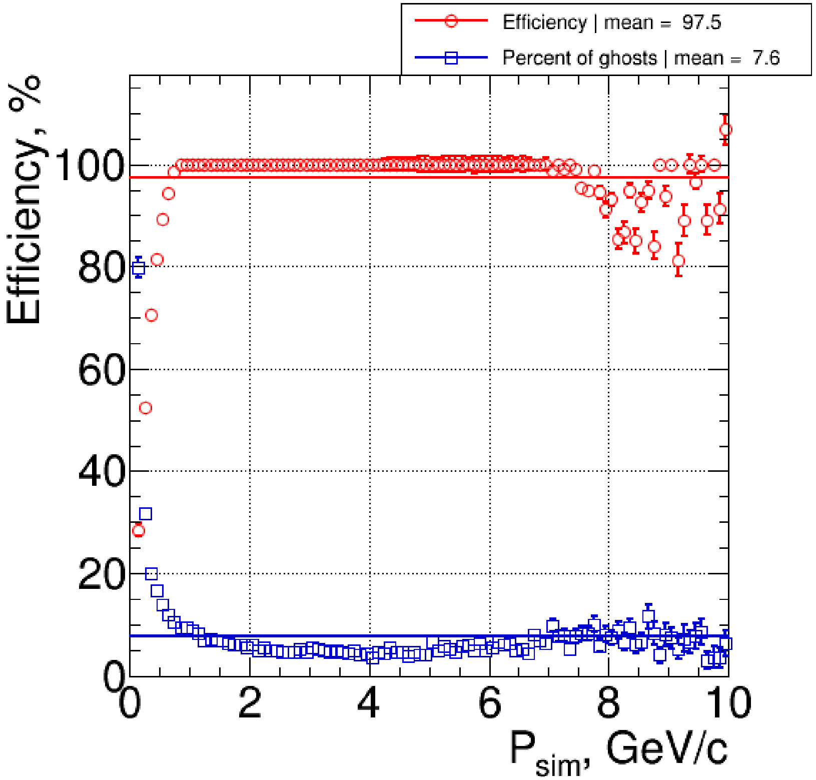


simulated 100 track event in {X/R},{Y/R} coordinates, P=1.5 GeV/c

BM@N track reconstruction in an ideal case

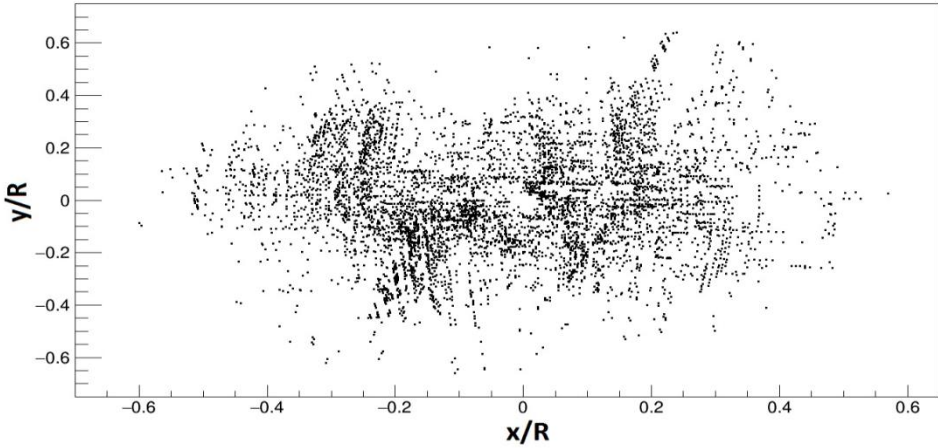
Ideal case – it is simulation with a pixel readout, without taking into account the strip readout of GEM detector. **Everything looks fine:**

- **Comparison of the complexity of “standard” and proposed algorithms gives O(n^3) operations for “standard” while it is O(1) operations for proposed one**
- **efficiency for transformed data is almost 100% and ghost level in low**

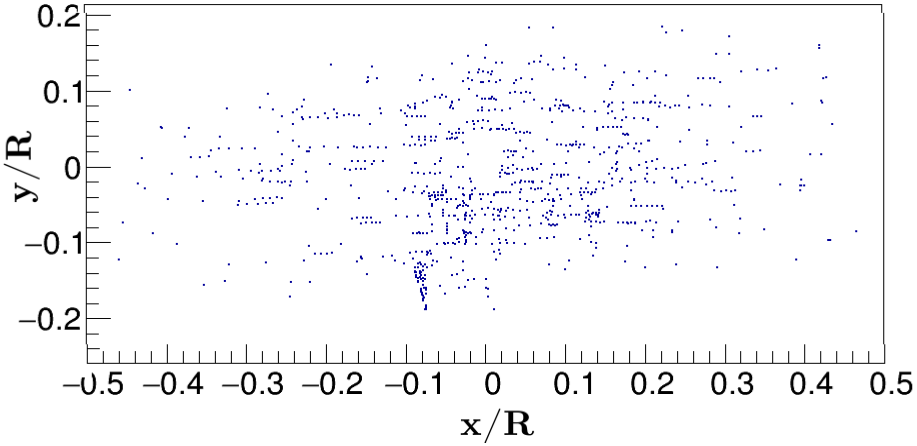


However, in reality the appearance of multiple fake hits completely supresses all hits of useful tracks and causes a sharp increase of ghost tracks.

Closer to fake reality



UrQMD Au-Au events in {X/R,Y/R} after fake rejection, hits for all 12 stations



UrQMD Au-Au events in {X/R,Y/R} after fake rejection, hits for the last 5 stations

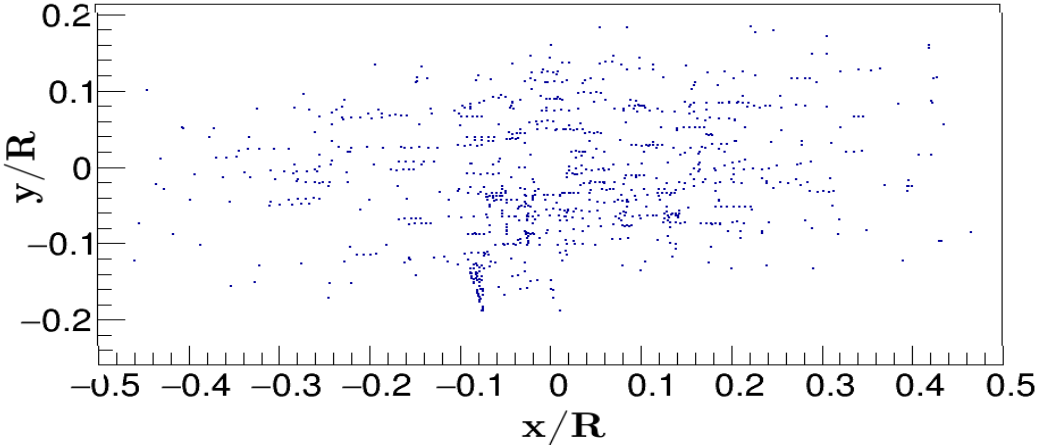
New idea with a potential attributing to transformed hits

Let us consider the basic properties of tracks in such coordinate system

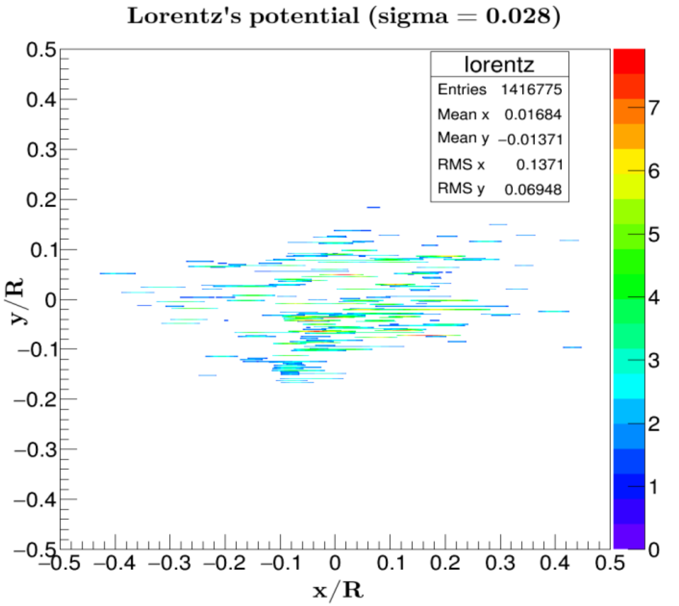
1. **Tracks on {X/Z,Y/Z} plane look as almost horizontal groups of close equidistant hits where distances depend on the track momentum**
2. **So we should enhance the influence of these track properties as closeness and horizontality.**
3. **It can be done by attributing to every hit such a potential that would emphasize only useful track properties.**
4. **Lorenz potential was chosen with horizontal halfwidth σ and ordinate width σ_y**

$$V(x,y) = \frac{1}{1 + \frac{(x-x_i)^2}{\sigma_x^2} + \frac{(y-y_i)^2}{\sigma_y^2}}$$

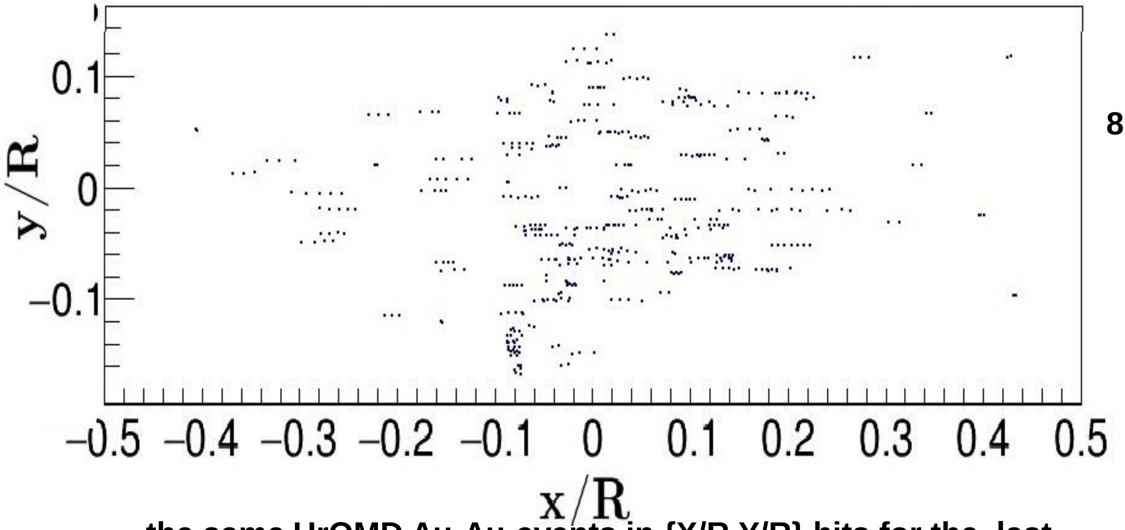
Preliminary results of the potential approach for BM@N



UrQMD Au-Au events in {X/Z,Y/Z} after fake rejection hits for the last 5 stations



8-12 stations view of Lorenz potential with σ=0.028



the same UrQMD Au-Au events in {X/R,Y/R} hits for the last 5 stations after potential cut applying

Lorenz potential with ***σ=0.028 σ_y=0.00075 and cut=1.5*** **was applied.**

A study to be done to find out an explicit dependence of a potential amplitude and corresponding track momentum

Simulation concept of NICA-MPD-SPD Tier0-Tier1 computing facilities

**V. Korenkov, A. Nechaevskiy, G.Ososkov, D. Pryahina,
V. Trofimov, A.Uzhinskiy
LIT JINR**

Simulation of grid-cloud systems

- **Substantial optimality study is needed to avoid possible and quite expensive mistakes on design and development stages of any grid-cloud system**
- **The study of grid-cloud system optimality is based on the optimality criterion which minimizes the equipment set (cost) under unconditional fulfilment of SLA (Service Level Agreement)**
- **Such studies can be efficient when it is based on scrupulous simulations of**
    - **Job stream with knowledge of**
        - **Job types (simulation, analysis, reconstruction)**
        - **Statistical information about distribution of their arrival and execution times**
    - **computing resources (number of compute nodes, the architecture of a computer system, installed software, CPU consumption)**
    - **Data flow**
- **Efficient simulation of grid-cloud systems should take into account the functioning quality of this system to evaluate its performance and to forecast its future taking into account dynamics of its evolution.**

Basic simulation concepts

- The best way to evaluate dynamically the system functioning quality is using its **monitoring tools**
- The simulation program is to be combined with real monitoring system of the grid/cloud service through a **special database (SDB)**
- To ensure a developer from writing the simulation program from zero on each development stage it is more feasible to accept a **twofold model structure,** when it consists from

1. **a core – its stable main part independent on simulated object and**
2. **a declarative module for input of model parameters defining a concrete distributed computing center (DCC), - its setup and parameters obtained from monitoring information, as dataflow, job stream, etc**

- **SDB intention** is just to realize this declarative module work and provide means for output of simulation results
- **Web-portal** is needed to communicate with SDB assigning concrete simulation parameters and storing results in SDB

What simulations should give us
on the design stage of a grid-cloud system

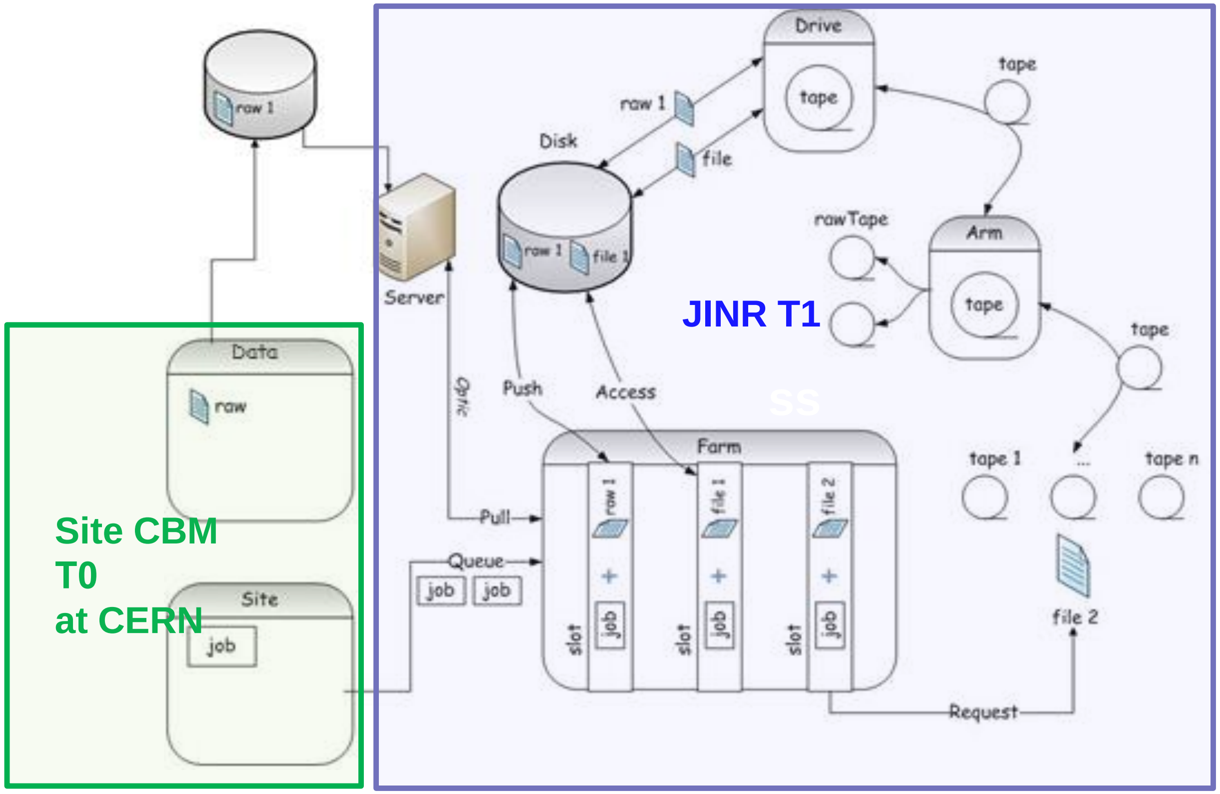
- **Evaluate grid-cloud system performance and reserves under various changes:**
  - **Different workloads**
  - **System configuration**
  - **Different scheduling heuristics**
  - **Hardware malfunctions**
- **Balance the equipment needed for data transfers and storage by minimizing cost, malfunction risk and execution time;**
- **Optimize resource distribution between user groups;**
- **Predict and prevent a number of unexpected situations**
- **Test the system functioning to find bottlenecks.**

How it was realized

- **Our team has already the experience with simulation grid structures inspired by GridSim library (http://www.buyya.com/gridsim) and job scheduler ALEA (http://www.fi.muni.cz/~xklusac/alea).**
- **The new simulation program called SyMSim (Synthesis of Monitoring and SIMulation) was developed according to the above basic concepts and succesfully tested for the JINR CMS Tier 1 center with robotized tape library.**
- **To accomplish that**
  1. **New classes are invented to declare the data store specific for the tape robot library;**
  2. **Input job stream is formed via data base;**
  3. **Data exchange process is modified from packet flow simulation into file transfer simulation;**
  4. **Software means for handling simulation results are provided.**

Tier1 Dataflow simulation

The problem is to simulate a data storage system with robotized tape library, where RAW data are to be transferred from disks of a great HEP experiment. In reality we were charged to design such data storages for the CMS Tier 1 at JINR.



How it works on T1 site:

1. **From disk to tape:**

- If slot and file are available,
job is executed at the farm;

2. **From tape to disk:**

- If file stored in tape library.
job reserves a slot, but is waiting
for necessary file on the disk:
the robot moves tape cartridge to the drive, cartridge's file system mounting to the drive, file is copied to the disk.

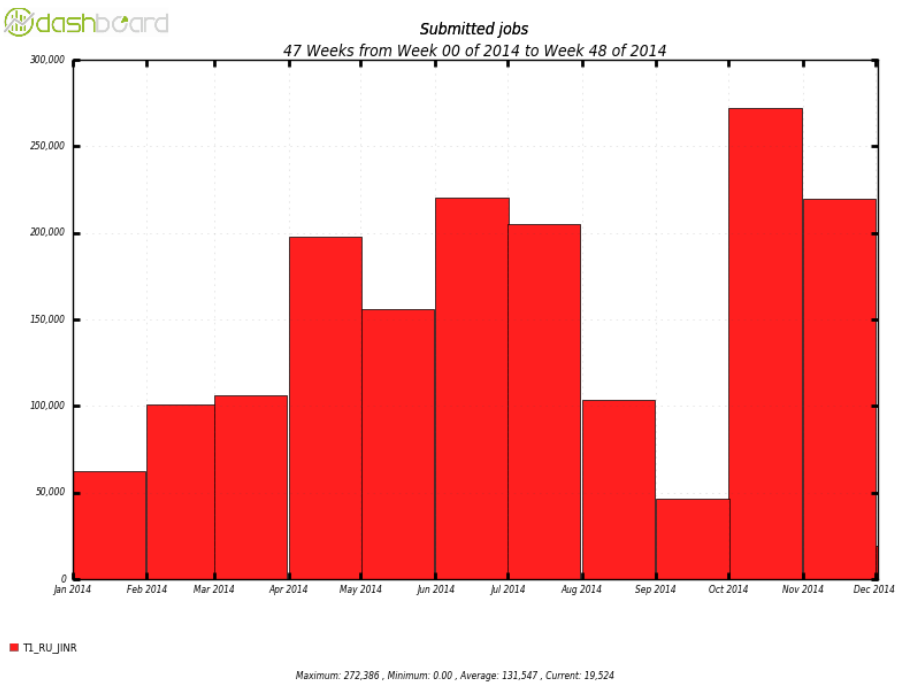
Scheme of the job and data flow at JINR T1

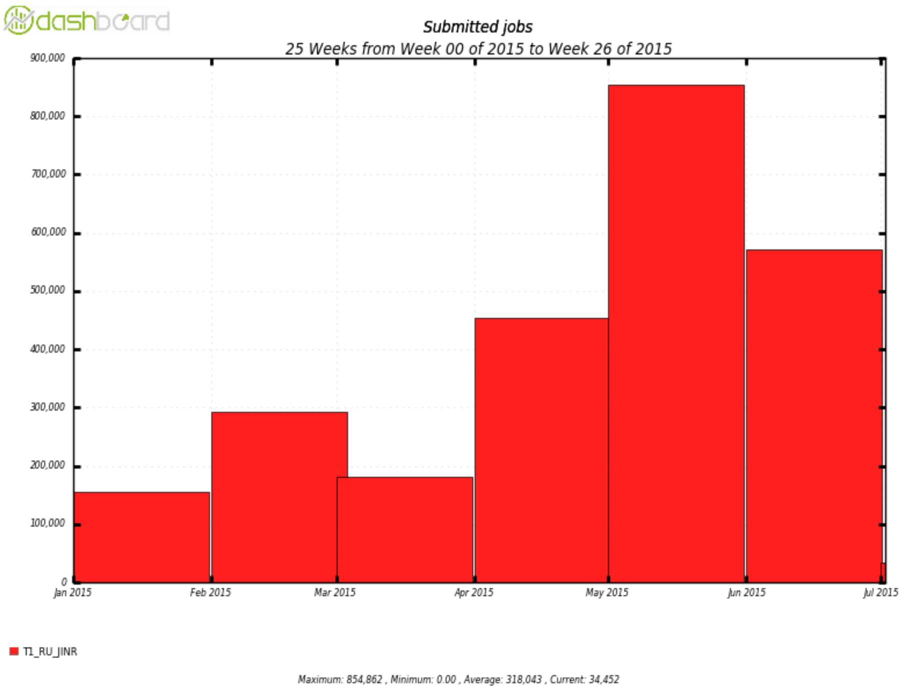
JINR Tier 1 statistics obtained from monitoring

CPU - 2400
Disks - 2400 TB
Tapes - 5 PB

these parameters from real T1 were set to the model

Statistics was taken from



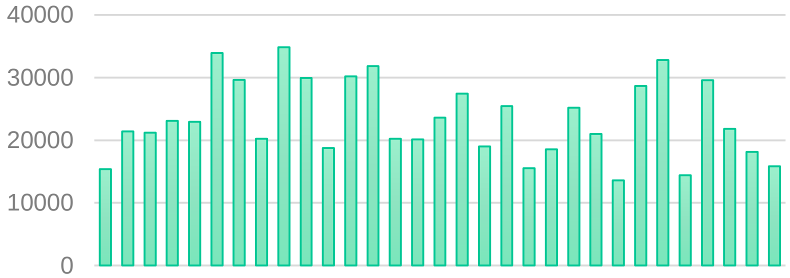


~ 2 mil. Submitted Jobs (2014)

~ 3 mil. Submitted Jobs (6 month of 2015)

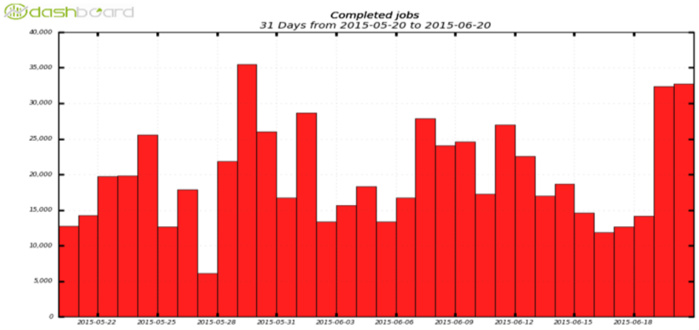
Examples of Real and Generated Workflow

Completed jobs (simulated)



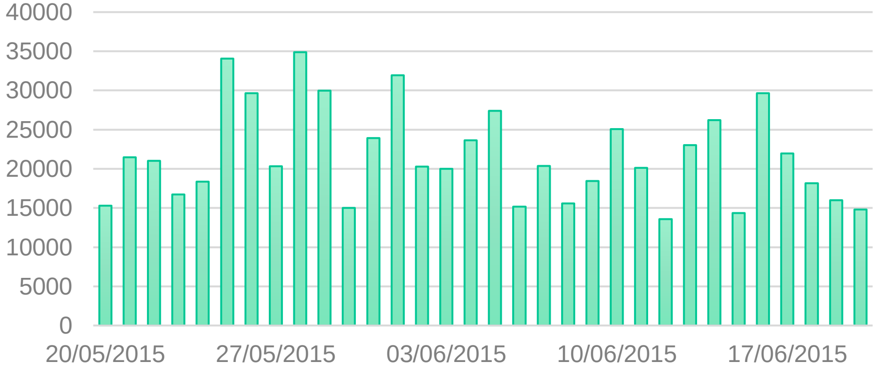
X = 24000 S = 6100

Completed jobs (real)



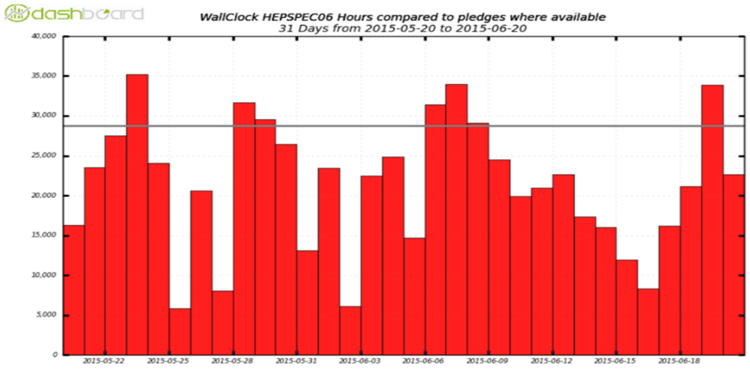
X = 19700 S = 6700

WallClock HEPSPEC06 (simulated)



X = 22000 S = 6400

WallClock HEPSPEC06 (real)

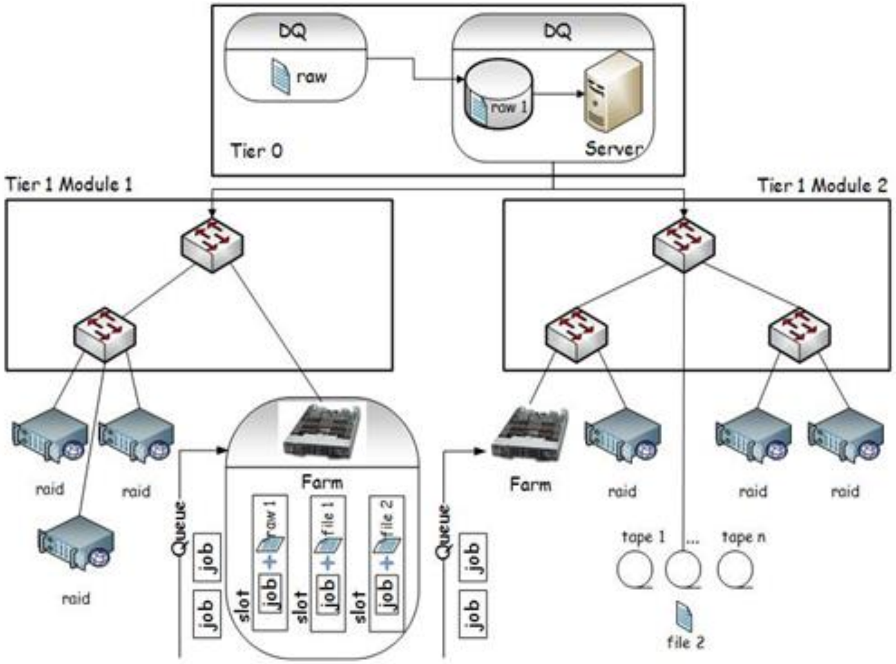


X = 21300 S = 8100

Real and simulated distribution look similar in terms of the error corridors

These two examples among some others were used for the positive validation of the running CMS T1 model and encouraged us to simulate the more sophisticate and planning yet the T0/T1 system of NICA project.

Simulation evolution:
from CMS Tier1 to NICA Tier0-Tier1



Data storage and processing scheme of Tier0-Tier1 level

Tier 0 module denotes the center of data gathering from the experiment (either MPD or SPD). Obtained raw data are to be stored on disks. **One of planned problems is to recommend the volume of the disk store and a temp of data transfer to the robotized library** which is the part of Tier 1 center. This two-level structure is interconnected by a local area network

DQ on this scheme denotes not only DAQ of the corresponding experiment, but includes also the means of communications and buffer cleaning. (AN).

Initial information to start simulation are parameters setup of

- **designed hardware**
- **data flow,**
- **job stream**

their characteristics are taken from
Real data of CMS Tier1 monitoring and TDR DAQ MPD

Simulation of T0/T1 1

Database design

Database contains the description of the grid structure, each of its nodes, links between nodes, running jobs information, execution time, the monitoring results of the various subsystems of the grid and the simulation results.

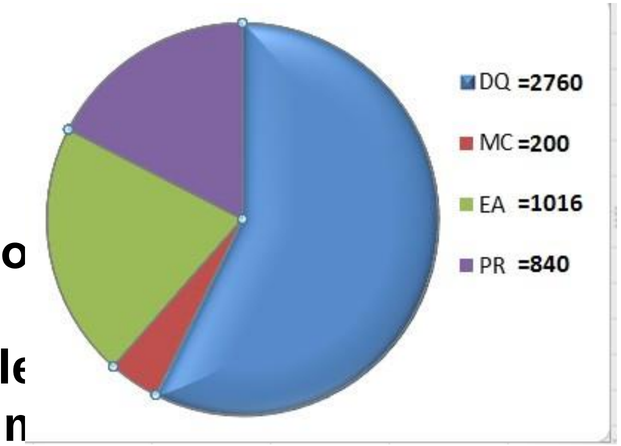
- ***Reminder:*** The simulation program is to be combined with a real monitoring system through **a special database (DB),** which intention is just to input of model parameters and output of simulation results
- **Web-portal is needed** to communicate with DB assigning concrete simulation parameters and storing results in DB

Database main tables

- **Experiments —contains information about the experiments;**
- **Simulation_Parameters — describes starts (runs) simulation program;**
- **Configurations — contains a description of the simulation configuration;**
- **Jobswaiting — contains a description of a job flow (the model of input data);**
- **Results — program results.**

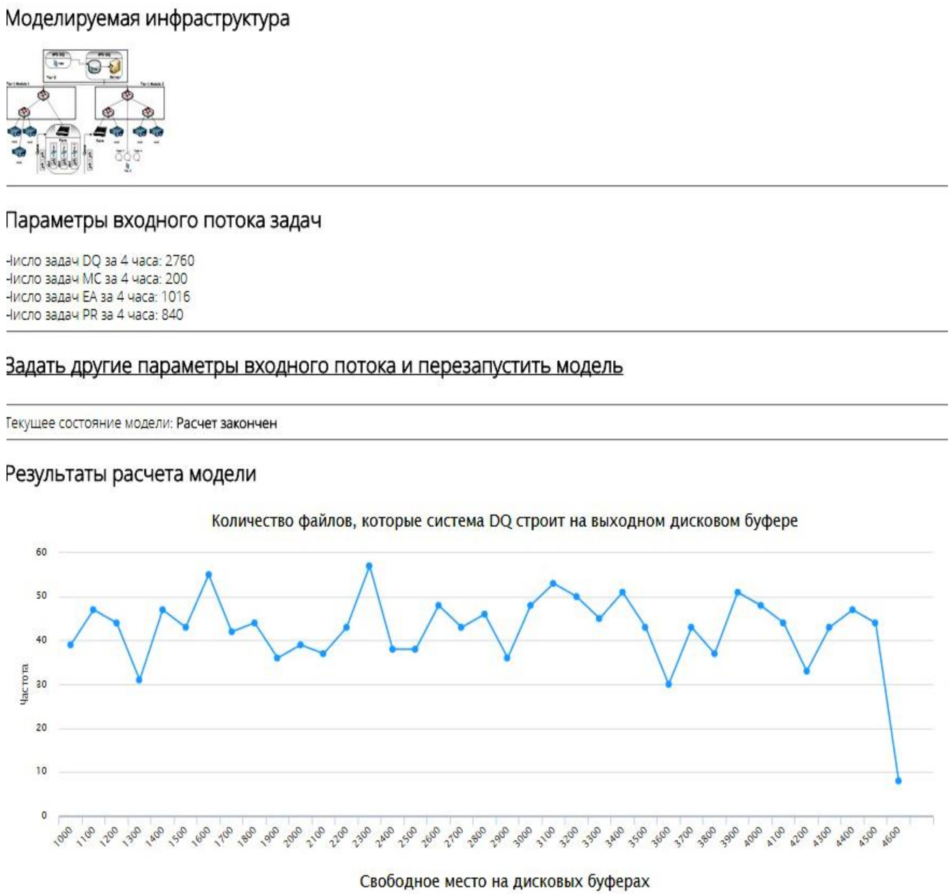
Four types of jobs are generated

1. **Data acquisition (DQ) – simulated "raw" data to be sto**
2. **Monte-Carlo (MC) – do not need input data**
3. **Express analysis (EA) – jobs use recently obtained fil**
4. **Reconstruction processing (PR) – jobs consume the n of resources**



Simulation

- **Web-portal functions**
- Interaction with the database.
- Present current model structure and generated workflow description.
- Set new workflow with different parameters (number of DQ, MC, EA, PR jobs) generation.
- Simulation results representation (graphics, diagrams).



Моделируемая инфраструктура

Параметры входного потока задач

Число задач DQ за 4 часа: 2760
Число задач MC за 4 часа: 200
Число задач EA за 4 часа: 1016
Число задач PR за 4 часа: 840

Задать другие параметры входного потока и перезапустить модель

Текущее состояние модели: Расчет закончен

Результаты расчета модели

Количество файлов, которые система DQ строит на выходном дисковом буфере

Snapshot of SyMSim web-portalt

Simulation algorithm is designed that at the initial time all buffers are empty, the processor is not loaded and data are not transferred. Therefore the initial transition process must be excluded from the analysis. It also happens when the current job flow stops.
The result of the simulation program is a sequence of records in the database, which reflects all the events occurring at the system.

Thanks for your attention!