Processing experimental data and analysis of simulation codes from Nuclear Physics using distributed and parallel computing

Mihai Niculescu

Faculty of Physics

University of Bucharest

A thesis submitted for the degree of

PhilosophiæDoctor (PhD) in Physics

2012 September

Supervisor: Univ. Prof. Alexandru JIPA, PhD

Supervisor: Peter Hristov

Day of the defense: 21 September 2012

"[...] Punctu-acela de micare, mult mai slab ca boaba spumii,
E stapanul fara margini peste marginile lumii... [...] " —**Mihai Eminescu (1881)**

"[...] Weaker than a drop of foam, this small dot that moves and bounds
Is the unrestricted ruler of the world's unbounded bounds. [...]" —**Mihai Eminescu (1881) translation from [1]**

Preface

In this thesis we tried to show the impact of new technologies on scientific work in the large field of heavy ion physics and as a case study, we present the implementation of the event plane method, on a highly parallel technology: the graphic processor. By the end of the thesis, a comparison of the analysis results with the elliptic flow published by ALICE is made.

In Chapter 1 we presented the computing needs at the heavy ion physics experiment ALICE and showed the current state of software and technologies. The new technologies available for some time, Chapter 2, present new performance capabilities and generated a trend in preparing for the new wave of technologies and software, which most indicators show will dominate the future. This was not disregarded by the scientific community and in consequence section 2.2 shows the rising interest in the new technologies by the High Energy Physics community. A real case study was needed to better understand how the new technologies can be applied in HEP and anisotropic flow in heavy ion collisions was my choice. An introduction to the theory of heavy ion physics and anisotropic flow is presented in Chapter 3. Chapter 4 is a brief overview of ALICE experiment and its detectors. The computing part of this thesis presented in Chapter 5, it was necessary to develop a few applications and some of them proved to be useful outside the scope of the subject of the thesis. Here I describe the development of an event viewer application for simulated data generated by UrQMD, section 5.3. This prototype application can be used for educational purposes. Chapter 6 presents the obtained results and discusses them based on comparison with the data published

by ALICE experiment. Finally, the Chapter 7 contains a summary and the main conclusions of this thesis.

Acknowledgements

Back in my teenage years, when I was about 13 years old, I discovered an old almanac from 1984. I open it and found an article containing exactly the quotes above. It was about a thing I didn't heard before: The Big Bang Theory. It talked about ylem, about quarks, gravitation, space-time, the influence of particles density in Universe and their possible effects on the evolution of Universe and other stuff I never imagined they existed. After finishing reading the article, I was disappointed that it could not give me the answer to what happened before Plank time or before time itself. But the article proved to be useful because I knew from that moment what I must find out. For this reason, I want to thank my parents for bringing that almanac in our home.

The direction to the Faculty of Physics, Magurele, couldn't become possible without the encouragement from my teacher in elementary school, Sanda Deleanu. Thank you Ms. Deleanu the greatest teacher of all.

Thanks also to my high school teacher Mr.N. Stanculete for all his support in high school. I am grateful also to my bachelor, master and PhD. thesis coordinator Prof. Alexandru Jipa for all his support during this time. Many thanks for my colleague Sorin-Ion Zgura, who introduced me to the large research field of heavy ion physics, for his constant guidance and computing suggestions. To Ionut Cristian Arsene, for all his precious advises, scientific guidance and help, and last but not least for being the Daic Vader. I am grateful for many advices and long talks with my friend and colleague Ciprian-Mihai Mitu and also thanks for helping with the registration of the thesis.

Many thanks to people at CERN for their advices and comments, Andrei Gheata and Ramiro Voicu.

I especially thank Peter Hristov for very useful comments, advices and review of the thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present a brief review to set the background on the subject of the thesis. A special attention to the underlying physics theory for the subject of the thesis will be presented in Chapter 2.

We start with a brief visit into the realm of High Energy Physics (HEP) in Section 1.1. We present the computing needs at Large Hadron Collider experiment (LHC) in Section 1.2 and dwell deeper and enter into the realm of Heavy Ion Collisions focusing our attention on A Large Ion Collider Experiment (ALICE). In Section 1.3, we present the current computing solutions provided by ALICE in order to fulfill its computing needs. In Section 1.4, we will present a brief historical evolution of central processor units (CPU) and witness its inevitable rebirth as a multi-core processor. The impact of this change in technology brought in HEP community will be introduced in Section 1.5. We will conclude this chapter by presenting the aim of the thesis, both its physics goal and its computational goal, in Section 1.6.

1.1 Brief introduction into High Energy Physics (HEP) and Relativistic Heavy Ion Collisions

High Energy Physics is a large Physics research field which deals with the study of the subatomic constituents of matter and fundamental forces. The name "High Energy" comes from the fact one needs very high energies in order to break these

fundamental particles apart and produce new particles which do not exist in the normal state of matter. The requirement to consider the study of particles in the high energy domain, is given by the ultra-relativistic limit:

$$p \gg mc$$

or when its Lorentz factor is very large:

$$\gamma \gg 1$$

In this case, one can easily see that the energy of a particle is given almost completely by its momentum: $E^2 = m^2c^4 + p^2c^2$

The successes in the last century of both experimental and theoretical physics lead to the creation of a particle physics theory named the Standard Model. The current Standard Model formulation dates back in 1970's, but still describes with great success the interaction between all subatomic particles mediated by all fundamental fields (electromagnetic, weak and strong nuclear), excepting gravitation - which is descripted by the General Theory of Relativity. To date, the Standard Model has reduced all known particles and their interactions into a small set of particles (Figure 1.1), which can combine to form other known types of particles: protons, neutrons, kaons, pions, etc. I am not going to mention all of them here because the number of know particles raises to an order of few hundreds [3].

Although, the Standard Model has proven to be a good description of the interactions of all known particles it is not a complete theory. It has few problems of its own: it lacks a description of gravity at quantum scale and one of the keystones of the model, the Higgs boson, has yet to be revealed by experiments. But this year, strong indications of the discovery of Higgs were announced by ATLAS and CMS experiments [4][5]. The Higgs mechanism was first proposed in the mid 1960's by Robert Brout, Francois Englert and Peter Higgs in order to explain why the bosons W and Z have such huge masses and the photons have zero mass (Figure 1.1).

In order to produce and study the massive particles from Figure 1.1 and its composite particles, experiments must go to very high energies Table 1.1 in which

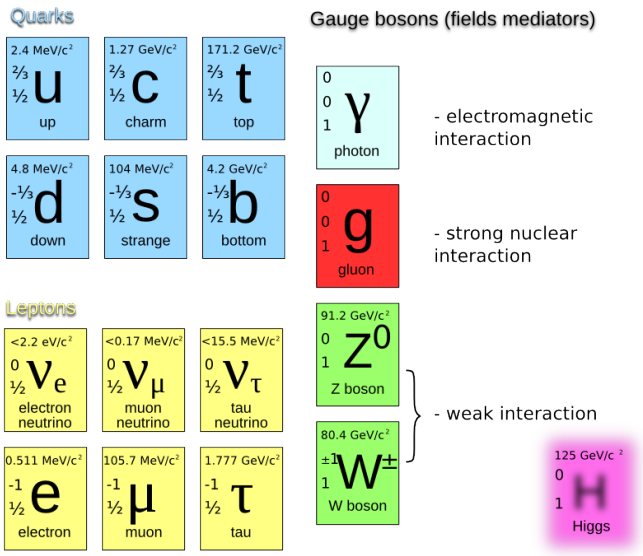

Figure 1.1: Elementary particles of The Standard Model - Matter particles (quarks and leptons) on the left side and force carriers (bosons) on the right side

the accelerated particles reach speeds near the speed of light ($c \approx 3 \cdot 10^8 m/s$).

Relativistic Heavy Ion Collisions is the part of the High Energy Physics which deals with the Nuclear Physics of colliding nucleus. Heavy Ion Collisions (HIC) are the only way we can study in laboratory the properties of nuclear matter in a state close to the one that occurred shortly ($\approx 1\mu s$) after the Big Bang.

Experiment	**Laboratory**	$\sqrt{s_{NN}}$ **(GeV)**
Bevalac	LBL	2.3
AGS	BNL	5
SPS	CERN	17.3
BRAHMS	BNL	200
ALICE	CERN	7000

Table 1.1: Beam energy reached at HIC experiments - Overview of colliding beam energies at different experiments around the world.

1.2 Computing challenges at A Large Ion Collider Experiment (ALICE)

In this section we give a brief presentation of the computing requirements of ALICE experiment as this thesis was based three years ago. A more detailed description of the ALICE experiment can be found in Chapter 3.

ALICE experiment is part of the Large Hadron Collider from CERN, an experiment optimized to study Heavy Ion Collisions. Being a High Energy Physics experiment, the scale of the computing requirements presented here could be applyied for other experiments at LHC. We describe computing requirements for the processing of data produced by ALICE in pp and A-A collisions every year. This information can be found in ALICE Technical Design Reports (ALICE TDR) [6] and LHC Computing Review (LHCCR) [7]

1.2.1 Input parameters

The input paramenters are based on the nominal figures of a standard data-taking year and derived from information contained in ALICE Physics Performance Report [8] and in ALICE Trigger , DAQ, HLT and Control System TDRs [9]

The best estimates for the input parameters during a Pb-Pb collision and considering a luminosity of $\approx 3 \cdot 10^{30} cm^{-2} s^{-1}$ and acquisition rate at 200KHz during TPC drift time of 88 μs are shown un Table 1.2

Center-of-mass energy	5.5 A TeV
Luminosity	$5 \cdot 10^{25} cm^{-2} s^{-1}$ in 2008
	$5 \cdot 10^{26} cm^{-2} s^{-1}$ from 2009
Total reaction cross-section	8 b
Nominal collision rate	$4 \cdot 10^{3}$Hz at average luminosity

Table 1.2: Parameters adopted for Pb-Pb collisions - taken from [6]

Assuming a charged-particle density of dN/dy = 4000 per event and an average acquisition rate of 100 Hz for both pp and Pb-Pb collisions, the data taking parameters for ALICE are show in Table 1.3

	pp	Pb-Pb
Event recording rate (Hz)	100	100
Event recording bandwidth (MB/s)	100	1250
Running time per year (Ms)	10	1
Events per year	10^9	10^8

Table 1.3: ALICE Data taking parameters - considering an average event size for: pp of 1 MB and Pb-Pb of 12.5 MB

1.2.2 Processing power

For the required processing power some conservative estimates can be obtained based on RHIC previous experience and theoretical extrapolations at LHC energy. The processing power parameters shown in Table 1.4 were calculated using an algorithm named Kilo Specmarks Integer year 2000 (KSi2K), maintened by Standard Performance Evaluation Corporation.

		pp	A-A
Reconstruction	KSi2K x s/event	5.9	740.0
Chaotic analysis	KSi2K x s/event	0.6	8.3
Scheduled analysis	KSi2K x s/event	16.0	240.0
Simulation	KSi2K x s/event	39.0	17000.0
Reconstruction passes	-	3	3
Chaotic analysis passes	-	20	20.0
Scheduled analysis passes	-	3	3

Table 1.4: ALICE Processing power parameters - Computing power required for processing ALICE data and Monte Carlo data as calculated with KSi2K algorithm [6]

It is understandable why the simulation needs more computing power than the other algorithms, since it generates events, do particle tracking and digitization. Since the total multiplicity is bigger in PbPb collision than in pp collisions, the processing power requirements for nucleus-nucleos collisions is also bigger.

1.2.3 Data storage

All data produce by ALICE detector (raw, calibration, condition) and by the collaboration (reconstruction data, analysis objects, Monte Carlo data) is stored

during the lifetime of the experiment. Several Tiers (levels) of storage are provided for backup and rapid access of the data [6]. A rough estimation of storage needs is shown in Table 1.5.

	Real data (MB)				MC data (MB)	
	Raw	**ESD**	**AOD**	**Event catalogue**	**Raw**	**ESD**
pp per event	1.0	0.04	0.004	0.01	0.4	0.04
pp per year ($\times 10^9$)	1.1	0.18			0.4	0.1
Heavy-ion per event	12.5	2.5	0.25	0.01	300	2.5
per year ($\times 10^9$)	1.4	1.0			3.0	0.9

Table 1.5: Estimation of event size at ALICE - for real data and from Monte Carlo (MC) simulations [6]

Data types used in Table 1.5:

- **RAW** - data as recorded by DAQ or from MC simulations
- **ESD - Event Summary Data** - is produced by the reconstruction programs from RAW data. It is smaller than RAW data because these are ROOT files with a compresion algorithm
- **AOD - Analysis Object Data** - files obtained from ESDs applying Physics cuts. Hence, they should be smaller than ESDs

1.3 ALICE Computing framework

In order to solve the computing challenges presented in Section 1.2, ALICE collaboration developed a computing framework based on ROOT system [10], called AliRoot [11].

The main objectives of AliRoot is to provide a common computing framework for the collaboration to do:

- **Simulation** - of pp and Heavy-Ion collisions and detector response
- **Reconstruction** - of real and simulated data
- **Analysis** - of real and Monte Carlo data
- **Visualization** - of real and Monte Carlo data

1.3.1 ROOT

Starting with the beginning of 1990s, it was clear that the future of software development is going to be Object-Oriented (OO) and the old programming languages (like FORTRAN) are going to be replaced by the more recent programming languages, such as C++. And such, a new wave was started to implement in more modern programming languages the old libraries written in FORTRAN: CERNLIB, PAW, GEANT3.

ROOT is a OO framework written in C++, started by Rene Brun and Fons Rademakers with the goal to solve the data analysis challenges of HEP. ROOT achieves its goal using a structured class hierarchy , provides Run Time Type Information (RTTI) for each object active in the system and a complete set of well integrated classes for: scripting (provided by CINT - C INTerpreter), documentation, graphics (2D and 3D), user interfaces (UI), analysis (multidimensional histogramming, random number generators, minimization algorithms, etc.) and input-output (IO) [10].

Being a framework, one is constrained to whole ROOT system. But, the ROOT system can be extended with user classes and one can benefit from all advantages of a well designed computing HEP framework: code reutilization, rely on tested code, focus on task at hand, use a complete set of classes.

1.3.2 AliRoot

Based on a C++ framework 1.3.1, ALICE Off-line Project started developing a framework, called AliRoot, for simulation, reconstruction and analysis.

From a design point of view, AliRoot follows good OO design techniques: code reusability and modularity. Also, one of the principles in designing AliRoot was to minimize the amount of user code and maximize the participation of the physicist in the development of the code.

Modularity of the code allows adding new code or replacement of parts of the framework with minimal or no impact on the whole system. This objective is achieved by splitting all detectors code in their own classes and filesystem directories. This modularity can be seen in a schematic view of AliRoot, in Figure 1.2.

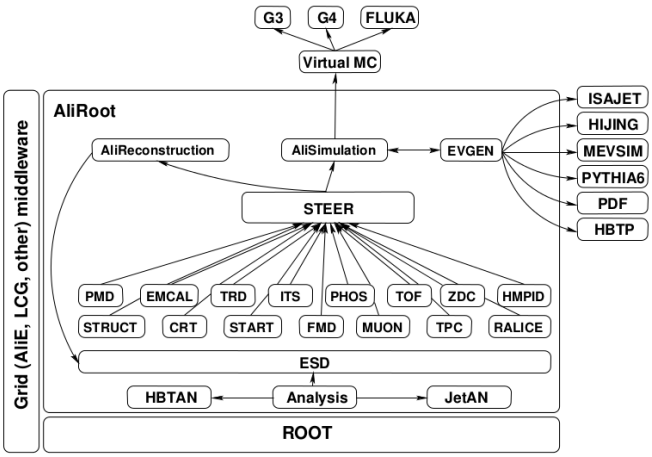

Figure 1.2: Schematic view of the AliRoot framework - ROOT is the foundation of AliRoot and each detector has its specific classes and their own filesystem directory; At the top, VirtualMC -is a common interface to access different Monte Carlo simulation packages

The STEER module is responsible for steering program for simulation, reconstruction and analysis. It also provides functions for general run management, creation and destruction of data structures, initialization and termination of program phases. The sub-detectors are independent modules that contain the specific code for simulation and reconstruction while the analysis code is progressively added. Detector response simulation can be performed via different transport codes via the Virtual Monte Carlo mechanism.

Reusability means to keep backward compatibility while evolving the system. This is a protection of the code that contains a large amount of scientific knowledge and experience.

1.3.3 PROOF

The Parallel ROOT Facility (PROOF) is a distributed ROOT solution based on event parallelism [12]. It allows data mining and data analysis on large sets of data on heterogenous clusters, optimizing CPU and IO utilization. Being part of ROOT framework, PROOF benefits from all advantages of good HEP framework: tools for data analysis and visualization. PROOF is based on Master-Slave architecture design (Figure 1.3): a local ROOT client session, a master PROOF

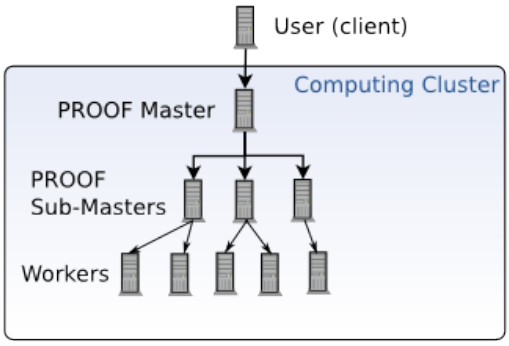

Figure 1.3: A Diagram of a PROOF cluster - The Parallel ROOT Facility realizes basic parallelism using a distributed computing model

server and a number of PROOF worker slaves. All the work is started from the user using a local ROOT session and send to the Master Server to be distributed to the worker nodes (slaves). All workers process in parallel the specified work and when finished, they send back the results to the Master server. After the results are merged on the Master, they are send to the user.

The most important design features of PROOF are:

- **Transparency**: distributed system is perceived as an extension of the local ROOT session
- **Scalability**: performance scales with the number of CPUs and storage disks
- **Adaptability**: the system adapts to variations in the remote environment

Successful implementations of PROOF clusters exists at CERN, named CAF (CERN Analysis Facility) and around the world, [13] as example for proof facilities dedicated for ALICE experiment.

Due to the fact PROOF is a distributed system, one disadvantage is that it can not be used when doing analysis on correlated events. But this is not a problem for HEP analysis codes since in high energy physics the events are uncorrelated. Another disadvantage might be when used in Online systems, where fast results are expected in real-time and network latencies can have a major impact.

Usually, PROOF is used for fast analysis on small samples and can help in tunning of analysis code.

1.3.4 ALICE interface to Grid

Another computing solution for the large amount of data storage and data analysis presented in Section 1.2 is the Grid. In this model, all resources are distributed at the HEP computing facilities of the institutes and universities participating in the experiment, Figure 1.4.

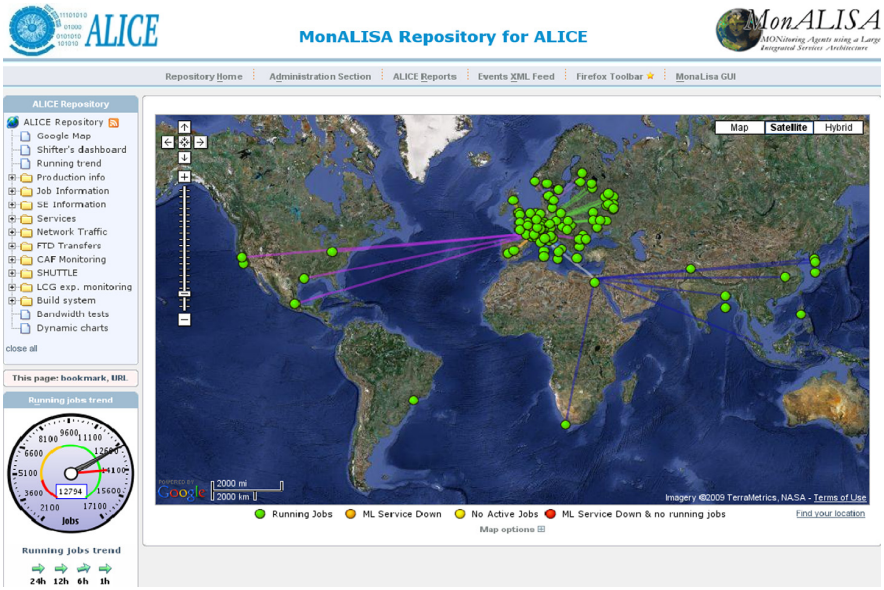

Figure 1.4: MonALISA showing ALICE Grid structure - Sites distribution around the globe [14] [15].

Different implementations for Grid middleware software can be build to access the grid computing resources: Globus toolkit [16], gLite [17], ARC [18], etc. ALICE implementation of grid middleware is called AliEn (**Al**ice **En**vironment).

AliEn is a transparent interface to the computing resources shared by the collaboration, shielding the user from the complexity of grid computing. It consists of the following components and services: authentication, authorization and auditing services; workload and data management systems; file and metadata catalogs; the information service; Grid and job monitoring services; storage and computing elements.

Although, AliEn has proven to be a successful solution for ALICE collaboration, it has the same problems as PROOF computing: it is a distributed computing environment and network latencies between computing centers dis-

tributed worldwide can be a problem when one needs very fast results. But unlike usual operating a PROOF cluster, the computing resources are bigger and shared among more people. This makes AliEn very useful for massive productions or complete analysis (organized and chaotic).

Chapter 2

Trends in computing technology and software

2.1 A historical snapshot

For decades, the law that described processor performance was Moore's law: performance doubles every 18 months [19]. This was achieved increasing the number of transistors that can be squeezed on a single chip and the frequencies at which the processors operate. But around 2003, problems appeared when trying to reduce the dissipated heat generated by the transistors. It became clear that although more transistors can be built to fit on a single chip, the frequency can not be increased, Figure 2.1.

Chip industry started to focus its attention from squeezing more transistors to squeezing more processors (cores) into a single processor socket. This trend created interest in multi-core systems architectures.

2.1.1 Parallel architectures overview

Based upon how a computing system uses the number of instructions and data streams, Flynn classified parallel architectures in four categories [21]:

- **SISD**: (Single Instruction, Single Data stream) this type of processor has no parallelism as a single computing element controls one data stream at a time. Common single-core processors, single-core super-scalar processors

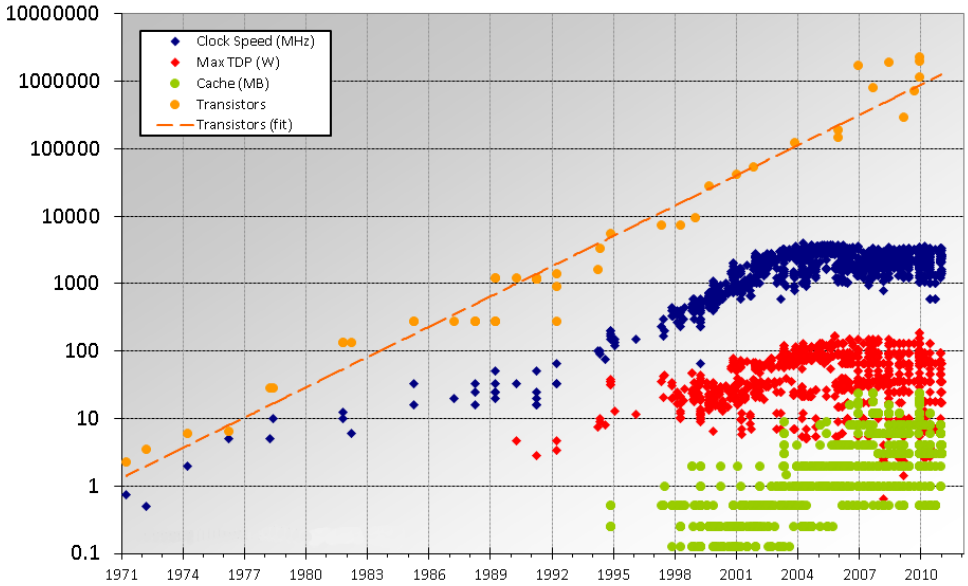

Figure 2.1: Features evolution of Intel's processors - courtesy CERN OpenLab [20]

- **SIMD**: (Single Instruction, Multiple Data stream) a processor which executes a single instruction that performs operations on multiple data. In this category fall: array processors (or vector processors), Graphical Processor Units GPUs, associative processors
- **MISD**: (Multiple Instructions, Single Data stream) Many processing units (example: CPUs) perform different operations on a single data stream. Not a very common architecture: systolic array architecture.
- **MIMD**: (Multiple Instructions, Multiple Data stream) an architecture where multiple different instructions are executed asynchronously and independently over multiple data streams. In this category we have: multi-core super-scalar processors, VLIW (Very Long Instruction Word) processors

2.1.2 Parallelization software

The improvement in performance gained by the use of a multi-core processor depends greatly on the software algorithms and their implementation. This time,

the performance of an application is not described by Moores law but instead by Amdahls law.

In parallelization, Amdahl's law gives the maximum speedup of an application that can be parallelized and the number of parallel computing elements:

$$S(N) = \frac{1}{(1-P) + \frac{P}{N}} \tag{2.1}$$

where S - is the maximum speedup; N - number of parallel computing elements; P - percentage of parallel code

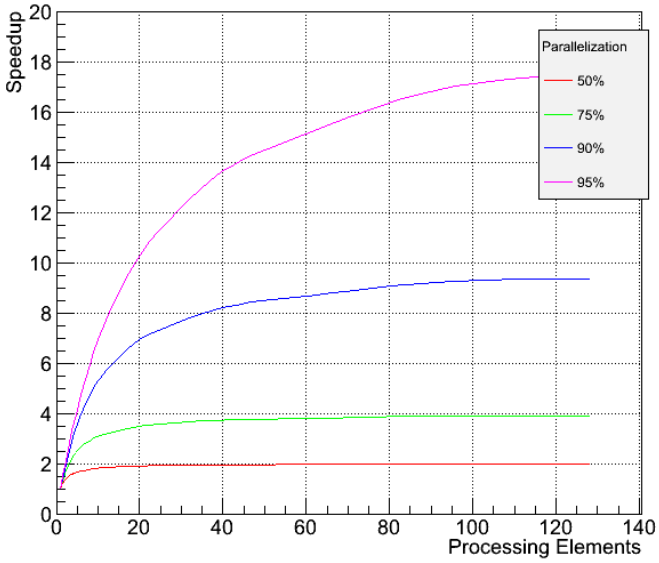

Figure 2.2: Plot of Amdahl's equation for parallel programs - Gained speedup depends on percentage of parallel portions and on the number of processing elements

Software toolkits and libraries were created to parallelize current software to run on CPUs and GPUs, or even both. Here we list a few of them and in my consideration the most important.

CPU software:

- **POSIX Threads** [22]: is an old and simple implementation of threads on Unix* like operations systems (Unix, Mac OSX, Linux, BSD, etc). Some implementations exists on Windows OS family.

- **OpenMP** [23]: a free multiplatform, scalable model implementation of multithreading
- **MPI** [24]: Message Passing Interface - for multi parallel processes. Implements parallelization using annotations as C macros.
- **Cray's Chapel** [25]: a multi threaded library using a high level of abstraction for CPUs and GPUs. Implements data parallelism, task parallelism and nested parallelism.
- **Intel's Threading Building Blocks (TBB)** [26]: an open source multiplatform C++ library that provides parallelism using built-in algorithms and templates

GPU software:

- **OpenHMPP** [27]: Open Hybrid Multicore Parallel Programming
- **CUDA** [28]: Compute Unified Device Architecture: a GPU general purpose parallel compute toolkit for Nvidia graphic cards
- **Stream** [29]: Advanced Micro Devices (AMD/ATI) implementation of a general purpose parallel computing toolkit for ATI graphic cards, now named ATI APP
- **OpenCL** [30]: Open Computing Language: a general purpose parallel computing toolkit for CPUs, GPUs and other processors. Developed by Khronos Group and OpenCL alliance (Apple, AMD, NVIDIA, etc)

In the absence of a standard for parallelization, processor and software industry began to create their own software toolkits and libraries, each with their benefits and disadvantages. It is obvious the advantage that a program written with a toolkit dedicated for a specific processor will perform better than any general purpose toolkit designed for multiple processors architectures. This can be also a disadvantage if one wants to target multiple processor architectures. But in some cases, it is desirable to develop for only one type of processor architecture. For example, when the program code is highly parallelizable and one wants to

use a large number of processing elements, as it is the case of GPUs where the order of processing elements is of hundreds.

2.2 Graphic processors

Using graphic cards for scientific computing is not a new idea and it has been around before graphic processor (GPU) creators decided to publish Application Programming Interfaces (API). Driven by the gaming industry, graphic cards and software libraries have been developed to use GPUs as a general-purpose computing device (GPGPU). The primary GPGPU providers are NVIDIA and AMD/ATI. Most modern graphics cards from NVIDIA can be programmed using Compute Unified Device Architecture (CUDA) an API extension to C language. Comparing CPU and GPU, the GPU is a specialized processor for highly parallel computation and assigns more transistors for processing than rather than data caching and flow control, Figure 2.3.

Figure 2.3: Schematic view of CPU and GPU - The GPU assigns more data processing elements (green boxes) than a CPU [28]

AMD/ATI developed its own toolkit called Stream, but has its focus on developing OpenCL (Open Computing Language), a GPGPU software developed in consortium at Khronos Group [30]. OpenCL goal is to be a cross platform GPGPU toolkit; this means, using same API, the GPU application can be compiled and run on processors from NVIDIA, ATI, CELL or S3 graphic cards, even CPUs.

In this section we give a brief overview of CUDA and how it is already used in HEP experiments.

2.2.1 Compute Unified Device Architecture (CUDA)

CUDA is a technology developed by NVIDIA that allows programmers to use a language based on C (C with extensions) to write applications for execution on GPU. Although, CUDA comes with an API for C language, bindings for other languages exists too (C++, FORTRAN, RUBY, Python, etc).

Freely available from NVIDIA site, CUDA comes with a Toolkit and Software Development Toolkit (SDK). The CUDA Toolkit contains software and documentation necessary to build GPU applications: nvcc C compiler, libraries for fast Fourier transformation (CUFFT) and basic linear algebra (CUBLAS), a cuda profiler, etc. Optionally, one can install the CUDA SDK which contains examples and makes learning easy. The execution model is SIMD (Single Instruction Multiple Data). One special function, called kernel is executed N times in parallel by N different CUDA threads. Although the threads are automatically managed by CUDA and do not require explicit management, the programmer has some control to the execution environment (number of blocks, threads, thread syncronization, etc.)

The programming model, illustrated in Figure 2.4, considers GPU (device) as a coprocessor to the CPU (host). In this heterogeneous environment, the GPU holds its own memory and processing capabilities.

2.3 Overview of Graphic processor utilization in high energy physics

In this section we present the sample study we did in [31].

2.3.1 NA62 Experiment

The NA62 experiment [32] (Figure 2.5) from CERN (European Organization for Nuclear Reasearch, near Geneva) aims at measuring the Branching Ratio of $K^+ \to \pi + \nu + \nu$, predicted in the standard model at level 10^{-10}. The trigger system is a very crucial part for the experiment. The reduction of interesting data has to be very effective in order to decrease the total bandwidth requirements for

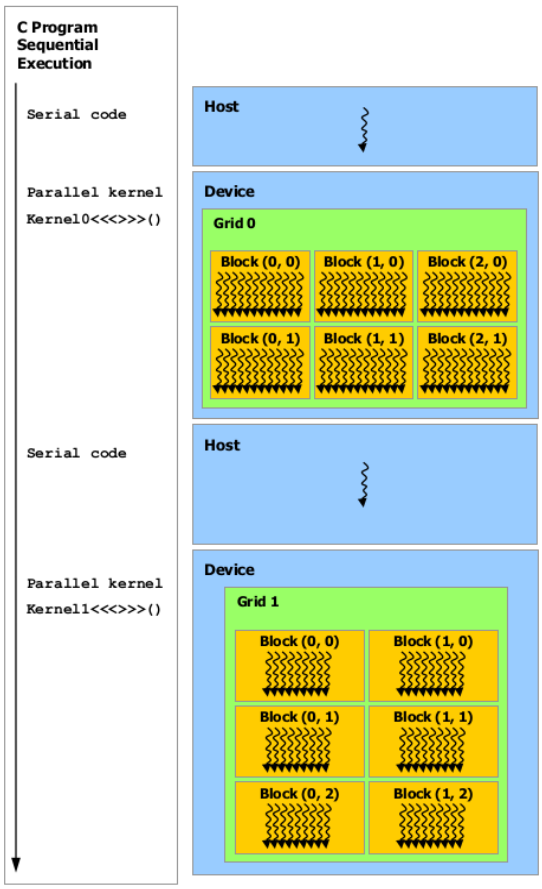

Figure 2.4: Heterogeneous programming model in CUDA - Serial code executes on the host while parallel code executes on the device [28]

the data readout.

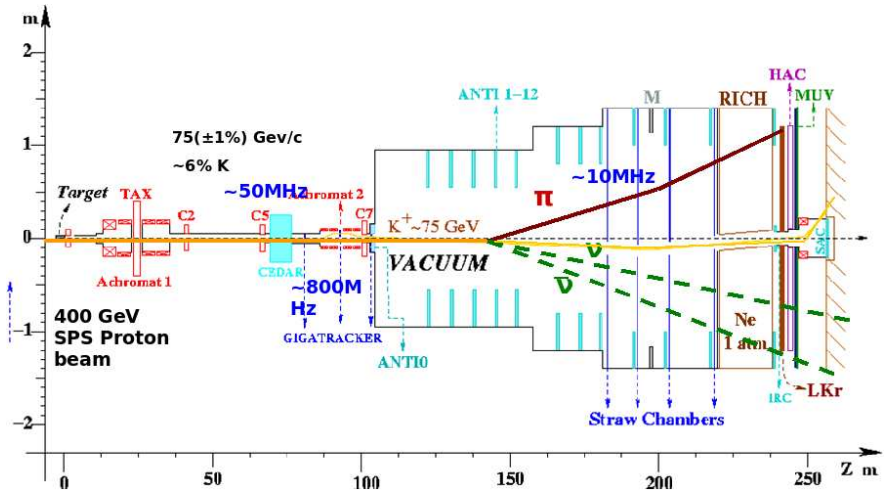

Figure 2.5: NA62 experiment at CERN - using CUDA for the trigger system [33]

The usage of graphic card processing in the construction of a trigger system, both in hardware and software, allowed to make decisions in real-time and increased the computing capability in the trigger, Table 2.1. To test the performances and feasibility, three algorithms were implemented. The tests were done on NVIDIA Tesla hardware with one GPU GT200 containing 240 cores, 4 GB DDR3 memory with 800 MHz and a bandwidth of 102 GB/s.

Matrix	Time CPU (seconds)	Time GPU (seconds)
1024x1024	4.0	0.52
2048x2048	130.62	1.51
3072x3072	490.98	5.25
4096x4096	1270.22	9.94

Table 2.1: Benchmark results for NA62 experiment - comparing CPU versus GPU performance [33]

The NA62 experiment concluded that the ratio of cost over performance is very appealing and the new hardware and software scheme implied by graphic cards can be adapted for high energy physics.

2.3.2 PANDA Experiment

One of track fitting algorithms used by PANDA experiment (GSI Darmstadt) [34] was ported to CUDA. The algorithm ported to CUDA uses tasks from FairCuda interface.

FairRoot is the simulation and data analysis framework used at FAIR (Facility for Antiproton and Ion Research, Darmstadt, Germany) [35]. CUDA is integrated in FairRoot in two steps:

- **Building System** Using FindCuda.cmake. The user do not have to take care of Makefiles or which compiler should be called (e.g. NVCC or GCC).
- **FairCuda interface** An interface is implemented which enables the use of GPUs implemented function from within a ROOT CINT session. The CUDA implemented kernels are wraped by a class (FairCuda) that is implemented in ROOT and has a dictionary. From a ROOT CINT session the user simply call the wraper functions which call the GPU functions (kernels).

Using GPUs for track fitting one can win orders of magnitudes in performance compared to the CPUs, however one has to determine how to divide the data into smaller chunks for distribution among the thread processors (GPUs).

Different benchmarks were performed on CPU and GPU in different modes: single float precision, double precision, and emulation mode - special mode in CUDA to emulate application running on CPU, Table 2.2

Mode/Track/Event	50	100	1000	2000
GPU (Emulation)	6.0	15.0	180	370
CPU	3.0	5.0	120	230
GPU (double)	1.2	1.5	3.2	5.0
GPU (float)	1.0	1.2	1.8	3.2

Table 2.2: Benchmarks for track fitting at PANDA experiment - a speedup of 70 is gained on GPU [36]

2.3.3 CBM Experiment

The CBM (Condensed Barionic Matter) Experiment [37] at FAIR is a dedicated fixed target heavy ion experiment.

The Kalman filter algorithm from CBM experiment was parallelized for Intel SSE and CELL using SIMD (Single Instruction Multiple Data) model. A speed up of 10000 times was determined for the SIMD version against the initial implementation. Porting the algorithm to GPU was the logical next step [38].

Porting the modified algorithm to GPU required noticing the differences between the two architectures: SIMD and SIMT, which is found on GPU hardware. GT200 chip contains 30 of these multiprocessors. Comparison results are found in Table 2.3.

Device	**Clock Speed (GHz)**	**Tracks/time** ($10^6/s$)
Intel Xeon (1 Core)	2.66	0.680
AMD Opteron (1 Core)	1.8	0.538
Cell SPE (1 processor)	2.4	1.15
Intel Core 2 (1 Core)	2.4	1.47
NVIDIA 8800 GTS 512	1.6	13.0
NVIDIA GTX 280	1.3	21.7

Table 2.3: Benchmarks for CBM Kalman filter - A higher value of Tracks/time is better [38]

2.4 Overview

In this chapter we have seen how the processors stopped increasing their operating frequency and become multi-core. Due to it highly parallel architecture, the graphic processor draw attention of the scientific community and high energy physics was not left outside this trend. We made a small survey (but relevant), of the utilization of GPU in HEP and see remarkable speedup gains, as almost 137x in the case of NA62 experiment. We are now raising the question if GPUs can be used for offline analysis, and to answer this we need a real world example. For this we choose to implement anisotropic flow analysis on GPU. An introduction to the theory of anisotropic flow is presented in the next chapter.

Chapter 3

Introduction to the Heavy Ion Physics

As can be seen in 1.1, the Standard Model reduced all known particles to a small set of particles and interactions. The electromagnetic force and weak interaction can be described within the framework of Quantum Electro-Weak Dynamics. Gravity lacks a quantum theory, but its description at macro scale is beautifully provided by the General Theory of Relativity. At collider energies, gravity can be totally neglected. Finally, the strong nuclear interaction, responsible for binding hadrons togheter is described by Quantum Chromo Dynamics (QCD). This Chapter gives an overview of basic concepts of QCD and set the background for the physics of this thesis.

3.1 Quantum Chromo Dynamics (QCD)

QCD is a non-Abelian gauge theory, which means that the strong interaction shows almost no resemblance to e.g., the electromagnetic interactions. The interaction between quark-quark can be calculated using lattice QCD, an illustration of the potential is shown in Figure 3.1.

It is seen that the quark-quark potential rises with distance, indicating that it will require an infinite amount of energy to separate two quarks. This is the theoretical explanation of the phenomenon of quarks *confinement in hadrons*. An-

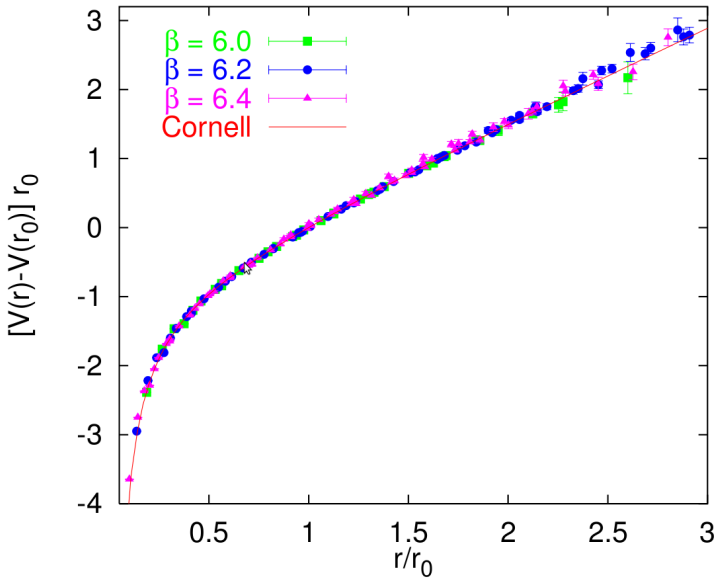

Figure 3.1: Potential between two quarks - calculated using lattice QCD and $r_0 = 0.5$ fm and $V(r_0) = 0$ [39]

other important property of the strong interaction, namely that the qq-potential drops as the distance between the quarks becomes small, a concept known as *asymptotic freedom*. Confinement and asymptotic freedom are key concepts in the dynamics of heavy ion collisions.

The confinement of quarks in hadrons is explained with the phenomenon known as pair production. While the the system quark anti-quark gains more energy and they distance themselves, the potential increases until there is enough energy to produce another pair quark anti-quark. The new particles will then recombine to form hadronic states. In this way it becomes impossible to isolate a single quark because quark-antiquark pairs will be created to form new particles instead, shown in Figure 3.2.

On the other side of the graphic from Figure 3.1, where the potential between quarks becomes very small and the attraction is also small, the quarks experience asymptotic freedom. The discoveries at Relativistic Heavy Ion Collider (RHIC) [40] supports the conjecture that this leads to the formation of a state of matter not observed since briefly ($t \leq 10^6$s) after the Big Bang. This state of matter is called a Quark Gluon Plasma (QGP), since it consists of free (free to interact

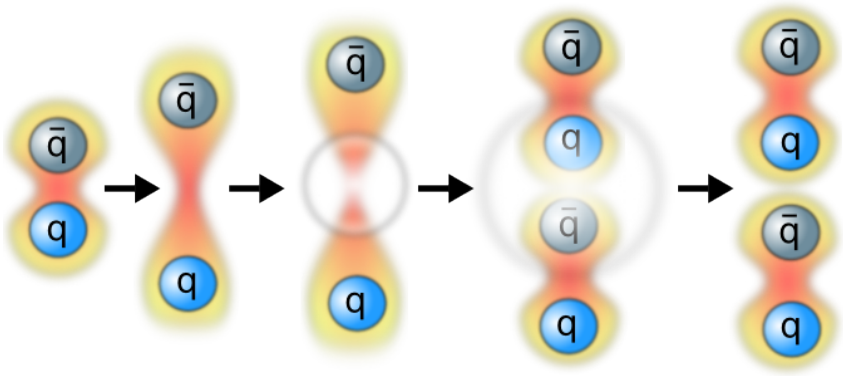

Figure 3.2: Illustration of pair production due to QCD confinement - Quarks are always bound in hadrons

within a small volume, that is) quarks and gluons. The formation and study of the QGP is one of the primary challenges in Heavy Ion Physics.

There are two different QCD approaches: Lattice QCD (lQCD) where the calculations are done on a discrete grid in spacetime and perturbative QCD (pQCD) working from the Lagrangians of QCD in the high Q (momentum transfer) limit.

3.2 The Search for Quark Gluon Plasma (QGP)

The first prediction of the existence of a highly compressed phase, where the partons (quarks and gluons) are asymptotically free was made in 1975 [41]. It was proposed that Heavy Ion Collisions can be use to study the properties of this new state of nuclear matter. In Figure 3.3 we see a conceptual illustration how the QGP is created. In heavy ion collisions this compression is of course extremely violent and the lifetime of the created state very short (of the order of 1 fm/c = 10^{-23} s or even shorter).

It is generally accepted that in the early lifetime of the Universe, after $\approx 1\mu s$ QGP may existed more similar to an ideal gas. This state, called weakly interacting QGP (wQGP) was possible during the birth of the Universe due to low baryon density $N_B/N \approx 10^{-10}$, compared with the baryon density obtained in heavy ion collision $N_B/N \approx 0.1$. In the conditions at RHIC energies and specific baryon density at heavy ion collisions, QGP presented more like a strongly interacting QGP.

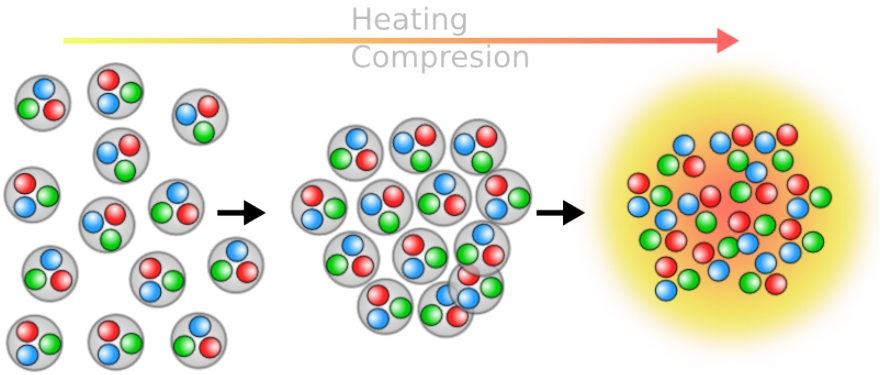

Figure 3.3: Illustration of the QGP process formation - Heating and compressing nuclear matter leads to the QGP

Calculations based on lattice QCD confirmed the existence of a critical point in the phase diagram, near the temperature $T_C \approx 170MeV$ which leads to a phase transition. Figure 3.4 show the dependence of the nuclear energy density ϵ in units of temperature T^4 as a function of the temperature. The steep curve around temperature T_C is an evidence of a phase transition from hadronic to partonic matter. The quantity ϵ/T^4 is proportional to the number of degrees of freedom in the system. At temperatures higher than T_C , the number of degrees of freedom saturates but at a value smaller than the Stefan-Boltzmann limit for an ideal gas. This is an indication of still strong interactions happening between quarks and gluons in the high energy density and temperature phase.

3.2.1 The phase diagram

The left side of Figure 3.5 is a sketch of the (T, μ_B) phase diagram, where μ_B is the baryo-chemical potential, of the nuclear matter as most physicists accept it today. The nuclear matter in its normal state is situated at T = 0 and μ_B = 937 MeV. The dot at μ_B 900 MeV and small temperature at the bottom of the figure together with the small line represents the liquid-gas phase transition line. The hadron gas phase occupies the relatively low temperature and baryo-chemical potential region from the bottom left of the figure. The continuous line ending with a critical point is the first order phase transition line which separates the hadron gas phase from the QGP phase (at intermediate μ_B) or the

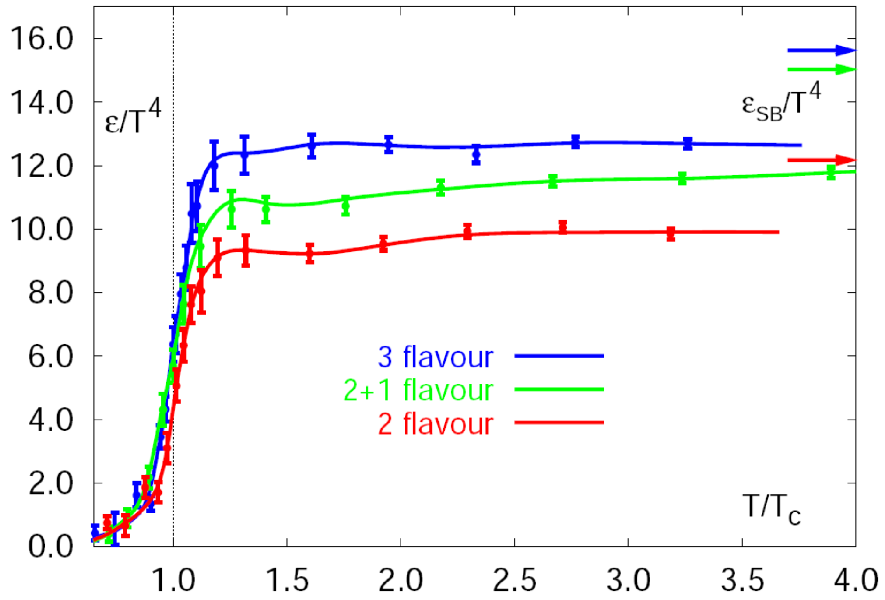

Figure 3.4: Energy density as a function of temperature - The arrows represent Stefan-Boltzmann limit for an ideal gas [42]

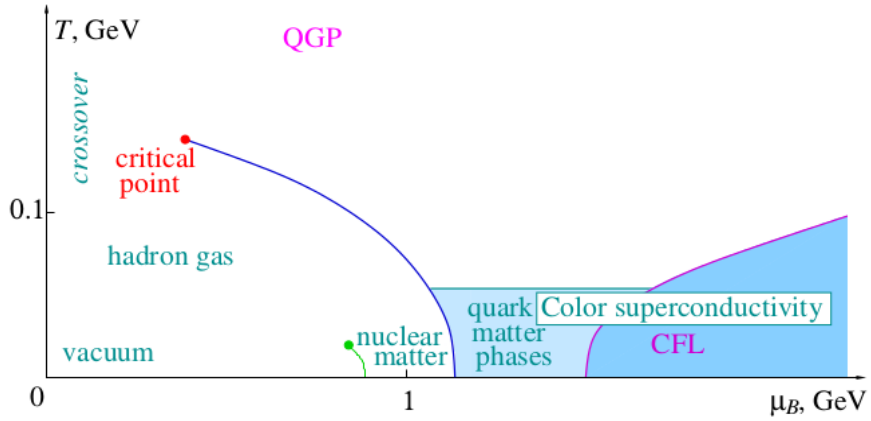

Figure 3.5: Sketch of phase for nuclear matter - Normal nuclear matter is situated at T = 0 and $\mu_B = 937$MeV [43]

quark dominated phases situated at low temperatures and high baryo-chemical potential. At low μ_B , it is thought today that the transition between the hadron gas phase and the confined QGP is of second order at the critical point and a smooth cross-over (high order phase-transition) at lower μ_B [44].

The exact location of the critical point is the subject of intensive theoretical and experimental work (SPS experiments and the future FAIR and NICA facilities).

3.3 Testing existence of QGP

There is no single definitive observable which can demonstrate the existence of QGP. The reason is that this state of nuclear matter has a short life and the produced particles that can carry information go through subsequent cooling and are bring back in the hadronic state. Here we will review some of the possible signatures of QGP that have been proposed or observed in experiments.

3.3.1 High Pt suppression

If the medium created in a heavy ion collision is truly strongly interacting it should affect particles propagation through this medium. Due to various low P_T collective effects such as color screening or Cronin enhancement this effect should be most pronounced at high P_T . A measure of this effect is expressed through the nuclear modification factor, R_{AB}, defined as:

$$R_{AB} = \frac{d^2N^{AB}/dp_T d\eta}{\langle N_{bin}\rangle\, d^2N^{NN}/dp_T d\eta} \tag{3.1}$$

where N_{bin} is the average number of binary collisions in a A+B collision, $d^2N_{AB}/dp_T d\eta$ is the differential yield in the A+B collision and $d^2N_{NN}/dp_T d\eta$ is the differential yield in a nucleon-nucleon collision. In both sides of the figure it is visible that in Au+Au collisions the high p_T charged particles are suppressed compared to p + p collisions. Moreover, in d+Au collisions where the formation of a QGP is not expected, an enhancement is actually seen, called the Cronin effect [47], confirming that the suppression in Au+Au collisions is not due to particular conditions of the colliding nuclei. The measured suppression at ALICE of high

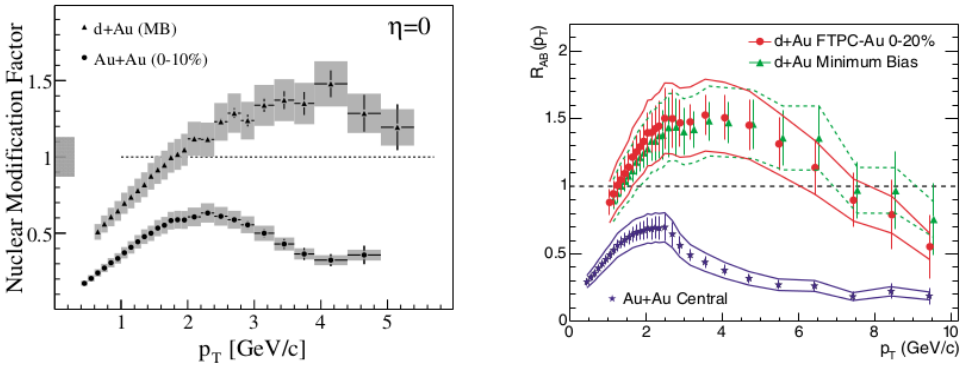

Figure 3.6: Nuclear modification factor versus p_T - Left: central Au+Au collisions (BRAHMS) [45]; Right: d+Au collisions (STAR) [46]

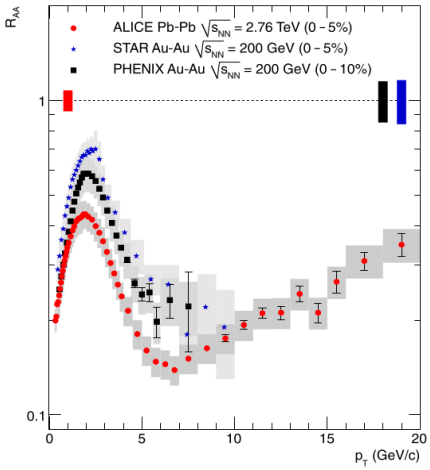

Figure 3.7: Comparison of Nuclear modification factor at ALICE and RHIC experiments (PHENIX and STAR) - The vertical bars around $R_{AA} = 1.0$ indicate the p_T independent scaling errors on R_{AA} [48]

p_T is stronger than that observed at RHIC which suggests an enhanced energy loss at LHC, thus the formation of a denser medium. The comparison is shown in Figure 3.7.

3.4 Hydrodynamics in Heavy Ion Collisions

The Figure 3.8 shows a schema of the evolution of the system in heavy ion collisions. Two relativistic nuclei collide inside the light-cone and through secondary collisions the system may reach thermal equilibrium leading to the formation of QGP. This state is short lived and the system expands and cools forcing quarks and gluons to hadronize. At this stage, the particles formed are seen by the detectors.

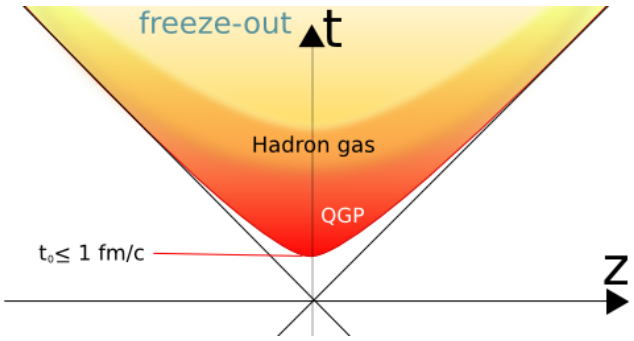

Figure 3.8: Schematic view of the dynamic matter in HIC - The Colliding nuclei move along the light-cone and beam axis

The expansion of the strongly interacting matter can be described using laws of ideal hydrodynamics, an idea first formulated by Landau in 1953 [49]. A quantitative agreement with theory predictions came from the first time from RHIC, both in central and non-central collisions [50].

In order to describe the expansion of the parton system using hydrodynamics, one needs to describe the equation of state using observables from HIC. The general equation of state of an ideal gas should have the form of:

$$P = P(\epsilon, n) \tag{3.2}$$

where P is expressed as a function of ϵ (energy density) and n (baryon density).

For other types of flow for e.g, viscous flow one needs to insert other coefficients as viscosity η, etc.

Starting from the basic hydrodynamical equation of energy-momentum conservation [51],

$$\partial_\mu T^{\mu\nu} = 0 \tag{3.3}$$

where current conservation is zero and $T^{\mu\nu}$ is the energy-momentum tensor. one can show the equations for ideal fluid [51]:

$$\dot{n} = -n\vartheta \tag{3.4}$$

$$\dot{\epsilon} = -(\epsilon + P)\vartheta \tag{3.5}$$

$$\dot{u}^\mu = \frac{\nabla^\mu P}{\epsilon + P} \tag{3.6}$$

where u^μ is the flow four-velocity vector, P pressure, ϵ energy density, the "dot" denotes time derivative, $\vartheta \equiv \partial \cdot u$ denotes the local expansion rate. The first two equations describe the dilution of the local baryon and energy density due to the local expansion rate ϑ, while the third describes the acceleration of the fluid by the spatial pressure gradients in the local rest frame, with the enthalpy $\epsilon + P$ acting as inertia. As a consequence of the space anisotropy in non central heavy ion collisions, the pressure gradients will also exhibit an anisotropy. Hence, the fluid elements will move, "flow", anisotropically.

3.4.1 Anisotropic flow

Due to the initial space anisotropy in non central HIC, and multiple interactions this anisotropy is transfered to momentum space. An illustration of space anisotropy of Au+Au collision at $\sqrt{s_{NN}} = 200GeV$ and impact parameter b = 8 fm is shown in Figure 3.9.

To quantify anisotropic flow in heavy ion collisions one must measure the flow harmonics v_n obtained from the Fourier expansion of the azimuthal distribution of particles, $r(\phi)$, for e.g. $r(\phi) = dp_T(\phi)/d\phi$. This function represents the distribution of the particles emitted into $d\phi$ at azimuthal angle ϕ. Its Fourier expansion

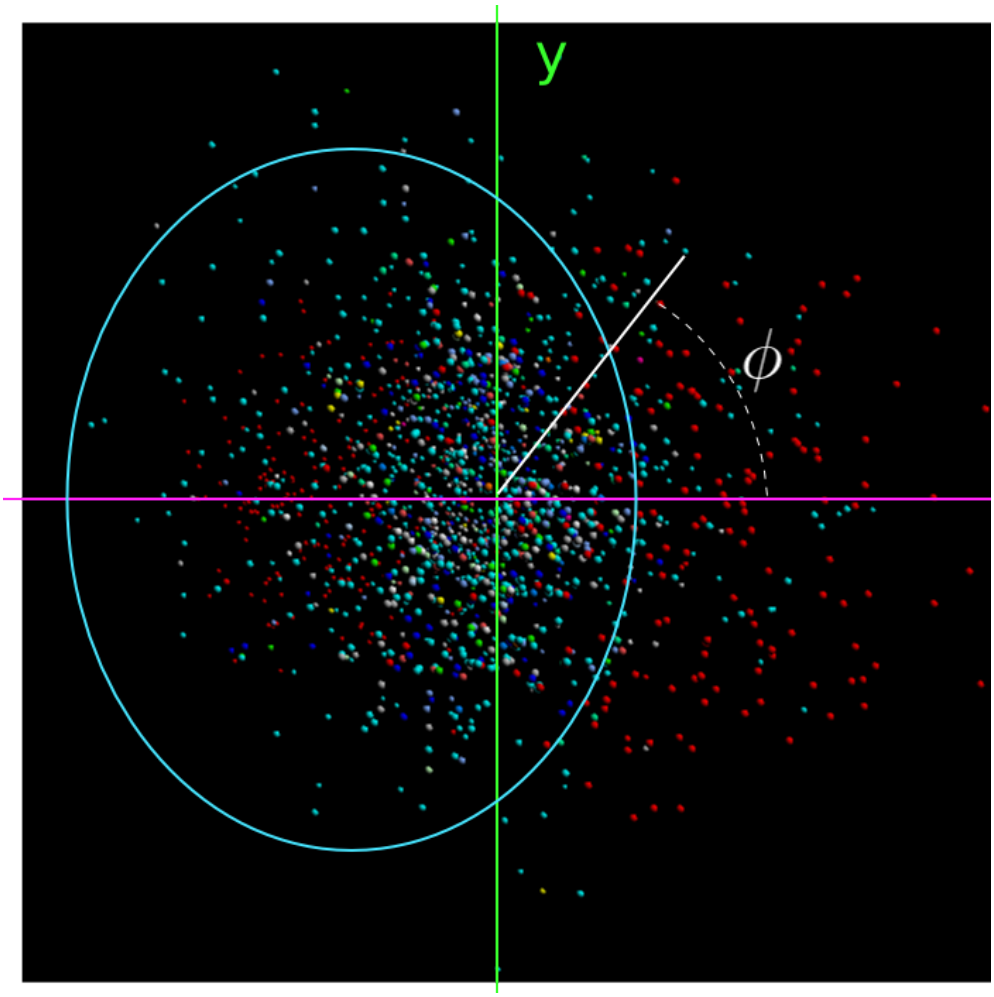

Figure 3.9: Space anisotropy of initial collision in Au+Au ($\sqrt{s_{NN}} = 200GeV$ and impact parameter $b \approx 8fm$) - The collision beam axis (Z) is perpendicular to the view and not shown. The red small spheres represent hadrons from the initial nuclei (also called spectators). The blue ellipse indicates the participant region

can be written [51]:

$$r(\phi) = \frac{x_0}{2\pi} + \frac{1}{\pi}\sum_{n=1}^{\infty}[x_n cos(n\phi) + y_n \sin(n\phi)] \tag{3.7}$$

where the coefficients of the Fourier distribution, x_n and y_n, become sums over the particles found in the rapidity window:

$$x_n = \int_0^{2\pi} r(\phi)\cos(n\phi)d\phi = \sum_v r_v \cos(n\phi_v) \tag{3.8}$$

$$y_n = \int_0^{2\pi} r(\phi)\sin(n\phi)d\phi = \sum_v r_v \sin(n\phi_v) \tag{3.9}$$

Each pair of x_n and y_n define a "n-type" flow harmonic :

$$v_n = \sqrt{x_n^2 + y_n^2} \tag{3.10}$$

The flow harmonics v_n can be related to a normalized distribution ($\int_0^{2\pi} r(\phi)d\phi = 1$) and found to be:

$$v_n = \langle\cos(n\phi)\rangle \tag{3.11}$$

where $\langle\rangle$ denotes an average over all particles in all events.

In particular, $v_1 = \langle\cos(\phi)\rangle$ is called directed flow, $v_2 = \langle\cos(2\phi)\rangle$ is elliptic flow, v_3 is triangular flow and so on.

In a coordinate system such as the one in Figure 3.9, the first two flow harmonics can be easily related to the momentum of particles:

$$v_1 = \left\langle \frac{p_x}{p_T} \right\rangle_i \tag{3.12}$$

$$v_2 = \left\langle \frac{p_x^2 - p_y^2}{p_T^2} \right\rangle_i \tag{3.13}$$

where the transverse momentum is $p_T = \sqrt{p_x^2 + p_y^2}$

The first harmonic, directed flow describes the sidewards motion of fragments in heavy ion collisions, and it carries early information from the collision. The shape of directed flow at mid-rapidity is of special interest because it might reveal a signature of a possible phase transition from normal nuclear matter to a Quark-Gluon Plasma (QGP) [52].

The elliptic flow, v_2 is a quantity very sensitive to initial spatial conditions of the system and describes the anisotropic emission of particles "in" or "out" of the reaction plane. During the expansion of the system, the spatial anisotropy decreases in time and inhibits anisotropic flow. Thus, the elliptic flow is particularly sensitive to the early stages of the system.

Elliptic flow v_2 and v_3 - called triangular flow - were observed at RHIC [53] [54] [55] and at LHC [56] [57] [58] [59] [60] and shown that v_3 depends on fluctuations of the initial geometry of the system. Other higher harmonics, v_4 - quadrangular flow, v_5 - pentagonal flow were measured at ALICE [56] [58]. In [58], it is shown that v_4 receives strong contributions from elliptical deformations of the initial state. In [61], it is argued that higher harmonic coefficients are greatly affected by the ratio η/s of shear viscosity to entropy density and the dependence of transverse momentum of the v_3 and v_5 is very sensitive to this ratio.

3.5 Standard event plane method (EP)

One the most used and intuitive methods for the anisotropic flow analysis is the Standard Event Plane method. In this method, the flow parameters v_n are obtained from the triple differential distribution and the event plane is calculated, as an estimation of the reaction plane ψ_R, for every flow harmonic.

For practical reasons, the equation 3.11 can not be used due to the fact the collision does not happen in the laboratory system, as presented in Figure 3.9. Instead, the impact parameter b changes randomly event by event. Since the reaction plane is defined as the plane spanned by the impact parameter vector and the beam axis, results that the reaction plane angle changes randomly too, shown in Figure 3.10.

In this case, the particle emission with respect to the reaction plane can be written as [62]:

$$E\frac{d^3N}{d^3p} = \frac{1}{2\pi}\frac{d^2N}{p_t dp_t dy}\left(1 + \sum_{n=1}^{\infty} 2v_n \cos[n(\phi - \psi_R)]\right) \tag{3.14}$$

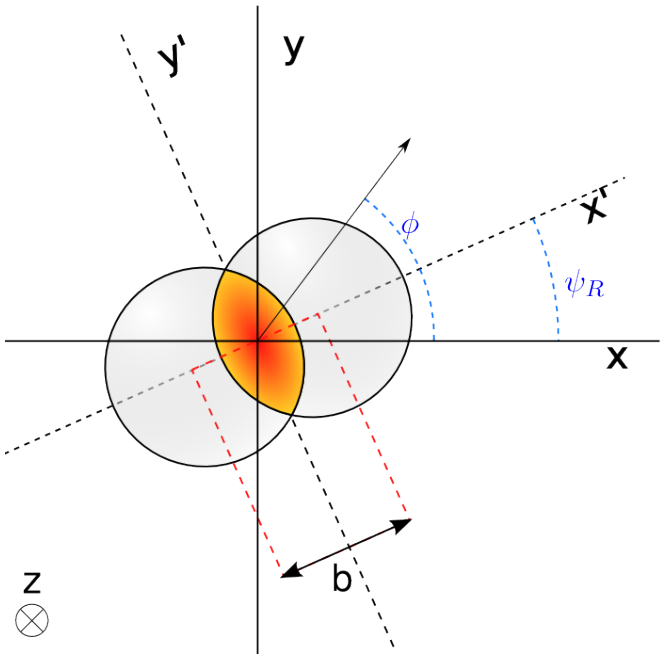

Figure 3.10: Heavy Ion Collisions in system laboratory - The reaction plane angle ψ_R changes randomly event by event

with the observed flow harmonics:

$$v_n^{obs} = \langle \cos[n(\phi - \psi_R)] \rangle_i \tag{3.15}$$

where ψ_R is the reaction plane angle; The orientation of the reaction plane changes randomly and can not be determined experimentally, instead we use an estimate of the reaction plane angle and this is the event plane [62]:

$$\psi_{EP} \simeq \psi_R = \frac{1}{n} \tan^{-1} \frac{\sum_i w_i \sin(n\phi_i)}{\sum_i w_i \cos(n\phi_i)} \tag{3.16}$$

where w_i are weights and can be set to enhance contribution of particles with large flow.

The corrected values of flow coefficients are obtained using the event plane resolution and the observed flow harmonics from Eq. 3.15:

$$v_n = \frac{v_n^{obs}}{R} \tag{3.17}$$

3.5.1 Event plane resolution

In order to estimate the event plane resolution need it by Eq. 3.17, there are two methods:

Full-event method

The full event resolution of the event plane is defined as:

$$R_{full-event} = \langle \cos[n(\psi_n - \psi_R)] \rangle_i$$

For a known value of v_n it can be calculated [62]:

$$\frac{\sqrt{\pi}}{2\sqrt{2}} \chi_n e^{-\frac{\chi_n^2}{4}} \times \left[I_{\frac{n-1}{2}} \left(\frac{\chi_n^2}{4} \right) + I_{\frac{n+1}{2}} \left(\frac{\chi_n^2}{4} \right) \right] \tag{3.18}$$

where $\chi_n = v_n/\sigma$, $\sigma = \sqrt{\frac{\langle w^2 \rangle}{2M\langle w \rangle^2}}$, M is the particle multiplicity and I_x are modified Bessel functions of order x.

Sub-event method

The sub-event method to calculate the event plane resolution splits the event into two separated equal multiplicity sub-events. The way they are split is arbitrary.

The sub-event plane angle is calculated for each sub-event, using the same equation as in Eq. 3.16:

$$\psi_{EP}^A = \frac{1}{n} \tan^{-1} \frac{\sum_i w_i \sin(n\phi_i)}{\sum_i w_i \cos(n\phi_i)} \tag{3.19}$$

where the sum is restricted to the particles in the sub-event.

The sub-event resolution is now given by the difference $\delta\psi_{sub} = \psi^A - \psi^B$:

$$R_{sub} = \sqrt{\langle \cos[n\delta\psi_{sub}] \rangle}$$

3.5.2 Autocorrelations

The flow harmonics v_n are meant to measure the average correlation between each particle and the rest of the event. This autocorrelation can introduce a

bias on the flow measurement and moves slightly the flow vector. The effect of autocorrelations is larger at lower multiplicity and smaller when the multiplicity is high.

There are two methods to avoid autocorrelation:

Sub-event correlation

The event is split in two sub-events and each particle is correlated with the event plane angle from the opposite sub-event. The flow harmonics for the each sub-event are calculated as:

$$v_n^A = \frac{1}{N/2} \frac{\sum_{i \in A} \left(\cos \left[n(\phi_i - \psi_n^B) \right] \right)}{\langle \cos \left[n(\psi_n^B - \psi_R) \right] \rangle} \tag{3.20}$$

The average v_n on sub-events will be:

$$v_n = \frac{1}{2}(v_n^A + v_n^B) \tag{3.21}$$

Full-event correlation

The flow harmonics are calculated by subtracting each particle from the flow. This implies for each particle to do a selection of particles (which do not contain particle i) and compute the event plane on this selection:

$$v_n = \frac{1}{N} \sum_{i=1}^{N} v_{n,i} = \frac{1}{N} \frac{\sum_{i=1}^{N} \left(\cos \left[n(\phi_i - \psi_{n,i}) \right] \right)}{\langle \cos \left[n(\psi_n - \psi_R) \right] \rangle} \tag{3.22}$$

where ϕ_i is the azimuthal angle of the particle i, $\psi_{n,i}$ is the event plane angle. We observe the full-event resolution of the event plane at the denominator.

Because the flow is estimated better when more particles are used in its calculation, usually the best choice is to use the full-event correlation. The advantage of the sub-event correlation is better when reducing non-flow effects.

3.5.3 Setting the flow weights coefficients

The weight coefficients w_i present in the estimation of the integrated flow V_n or the event plane angle in Eq. 3.16 depend on transverse momentum, particle mass, and rapidity and can be set arbitrary. One can use unitary values for all measurements. On the other hand, weights should be chosen in such a way to enhance the contribution of the particles with higher flow. Ideally, the weights should be proportional to the flow harmonics v_n [63]. Such that one must find a function $w_i(p_t, y) \propto v_n(p_t, y)$. At RHIC, these weights were set as $w^i = p_t^i$.

From an experimental point of view, the weight coefficients w_i must take into account the azimuthal anisotropy of the detector acceptance.

3.5.4 Differential and Integrated flow

The terms differential and integrated flow refer on the how the flow harmonics are computed [62]. If the flow is studied with respect to the transverse momentum (or pseudorapidity) $v_n = v_n(p_t, \eta)$ we call it *differential* flow. If the flow is a global property of the event(s), we call it *integrated* flow.

When studying the differential flow we compute $v_n = v_n(p_t, \eta)$ in separate kinematic windows (or bins) in a specific centrality, as in Eq. 3.17:

$$v_n(p_t) = \frac{\langle v_n \rangle_{p_t}}{R} \tag{3.23}$$

In this way, the differential flow coefficients v_n computed at each p_t bin describe the their dependence of p_t.

The integrated flow coefficients are calculated as the average on event(s) between $v_n(p_t, \eta)$ weighted by the number of particles at each p_t bin of the dN/dp_t distribution:

$$\langle v_n \rangle = \frac{1}{N} \int_0^\infty v_n(p_t) \frac{dN}{dp_t} dp_t \tag{3.24}$$

Chapter 4

Experimental setup

In this chapter we give a brief presentation of the ALICE experiment detectors which are useful for flow analysis. We start with a presentation of the Large Hadron Collider (LHC) in Section 4.1 and move our attention to A Large Ion Collider Experiment (ALICE) in Section 4.2.

4.1 Large Hadron Collider (LHC)

The LHC is circular tunnel with a circumference of 27 km located near the border of France and Switzerland and about 100 m below earth surface, Figure 4.1.

The first proposal of LHC project was done in the 1984, but due to the technological challenges the project come to fruition with a steady lean growth. The collider was completed and started its operation in 2008, when a massive magnet quench took place causing extensive damage to parts of the LHC machine. Repairs and upgrades to prevent further incidents lasted almost a year so the LHC restart took place in the fall of 2009.

There are high hopes at LHC to solve long standing problems from high energy physics. The main physics goals at LHC are:

- **Higgs Discovery** Higgs is a building block of the Standard model whose existence can explain how the particles gain mass. On 4 July 20012, ATLAS and CMS presented the discovery of a particle in the mass region around 125-126 GeV [4] [5] which could be the expected SM Higgs boson. Further

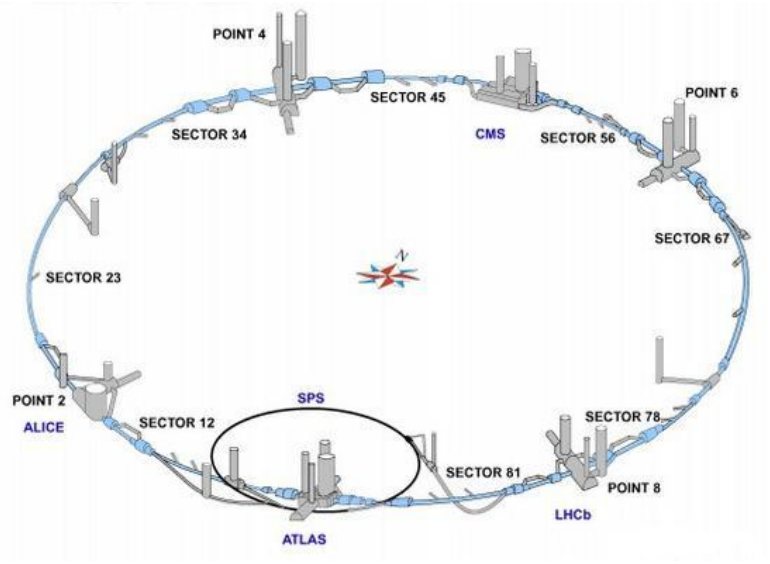

Figure 4.1: Schema of LHC complex [2] - comprises six detector experiments: ALICE [64], CMS [65], ATLAS [66], TOTEM [67], LHCb [68], LHCf [69]

studies will clarify the properties of the new particle.

- **Quark Gluon Plasma** ALICE is a dedicated experiment for HIC. One of interests at ALICE is the study of the properties of QGP
- **Matter-Antimatter asymmetry** There is a discrepancy between current observations and the current standard cosmological theory, the Big Bang Theory regarding the ratio of the matter and antimatter in the Universe. The problem is that in the early Universe this ration was 1.0, but today we observe different. This problem will be addressed by the LHCb.
- **Physics beyond Standard Model** This includes ideas and theories that have yet to be validated or invalidated, such as : extra dimensions, dark matter and dark energy, super-symmetry, etc.

LHC collisions program include colliding beams of: proton-proton (p-p), lead-lead nuclei (Pb-Pb) and asymmetric collisions proton-Ion (pA) to the maximum center of mass energy of 14TeV for p-p and 5.5TeV per nucleon pair for Pb-Pb.

4.2 A Large Ion Collider Experiment (ALICE)

ALICE is one of the experiments at LHC dedicated for the study of heavy ion collisions. Approved in 1998, the ALICE experiment is embedded in the former L3 Large Electron-Positron Collider (LEP) cavern benefiting from the large solenoid magnet already existing at the location. ALICE is consists of 18 subdetectors and it occupies a total volume of $6656m^3$ (16 m height, 16 m width and 26 m long). A schematic view of the ALICE detectors is seen in Figure 4.2.

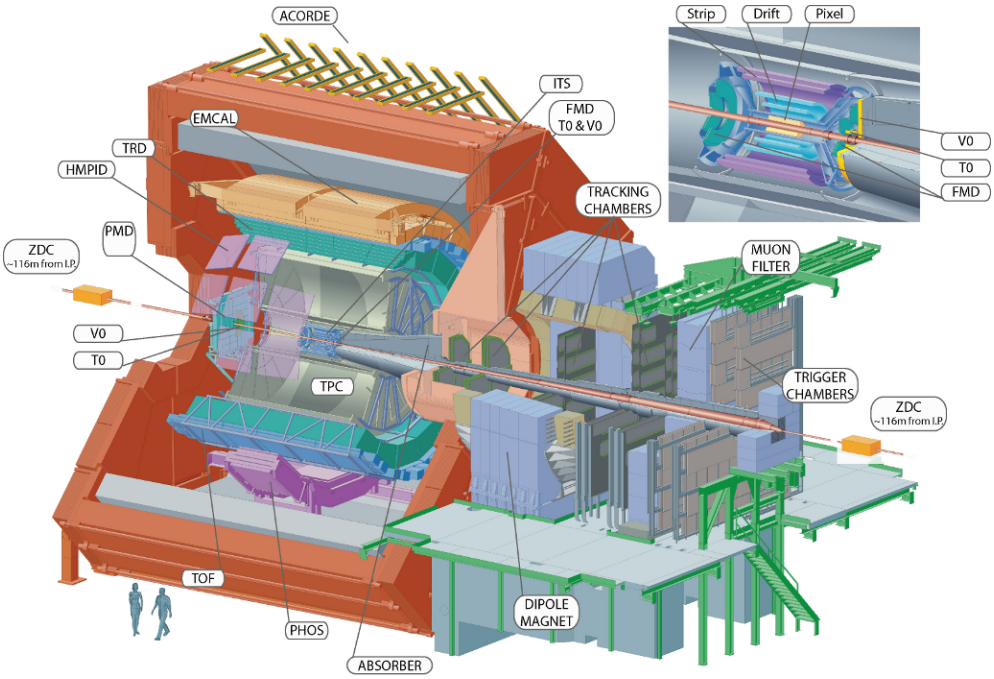

Figure 4.2: ALICE detectors - Taken from ALICE Figure Repository [70]

The primary physics goals of ALICE experiments are the study of the properties the confined nuclear matter, to answer the question of why quark masses are much lighter than the hadrons they build, and the study of the deconfined matter in Quark Gluon Plasma.

In the next sections we will give an overview of the ALICE detectors, a detailed description can be found in [71].

4.2.1 Central detectors

Time Projection Chamber (TPC)

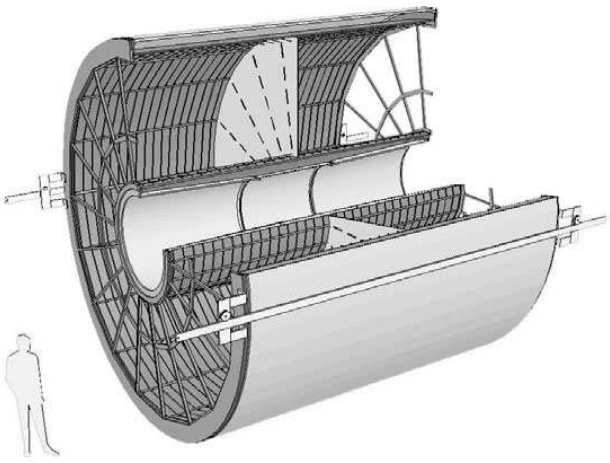

Figure 4.3: TPC detector - Time Projection Chamber

The TPC detector is one of the most important ALICE systems and one of the biggest: its outer radius is 2.78 m and its inner radius is 0.86 m, Figure 4.3. It is a gaseous detector with a volume of $90m^3$ surrounded by a cylindrical field cage. The TPC anode plane sits in the center of the TPC and two cathode planes sit at each end of the TPC cylinder. As charged particles traverse the Ne/CO_2 gas in the TPC, the knocked electrons are drifted towards the end plates. The drift time can be used to determine the z coordinate, while the r and ϕ coordinates are obtained directly from the position of the end plates.

Combined with other subsystems, TPC can provide full tracking in pseudo-rapidity $|\eta| = 0.9$, [71]. The TPC is capable of detecting the particles in the transverse momentum range $0.1 < p_t < 100GeV/c$, with a transverse momentum resolution of about 4.5% for $p_t < 20GeV/c$ in p-p collisions and about 6% for $p_t < 20GeV/c$ in central Pb-Pb collisions. The TPC covers full azimuth, with the exception of dead zones between the neighboring sectors (about 10% of the azimuthal angle).

Inner Tracking System (ITS)

ITS is the inner most ALICE detector with a cylindrical shape surrounding the beam pipe. It consists of of six cylindrical layers of silicon detectors, located at a radii between 4 and 43 cm and grouped in 3 subsystems, Figure 4.4:

- **Silicon Strip Detector (SSD)** an inner radius of 38 cm, provides further tracking information and connects the ITS tracks with TPC tracks.
- **Silicon Drift Detector (SDD)** an inner radius of 15 cm, provides tracking and particle identification information for low momentum particles ($p < 200 MeV/c$).
- **Silicon Pixel Detector (SPD)** an inner radius of 3.9 cm is the closes to the beam pipe. Provides accurate measurement of the collision vertex (with a resolution better than $100\mu m$), can be used as triggering detector and can provide tracklets information (vertex and clusters in each layer)

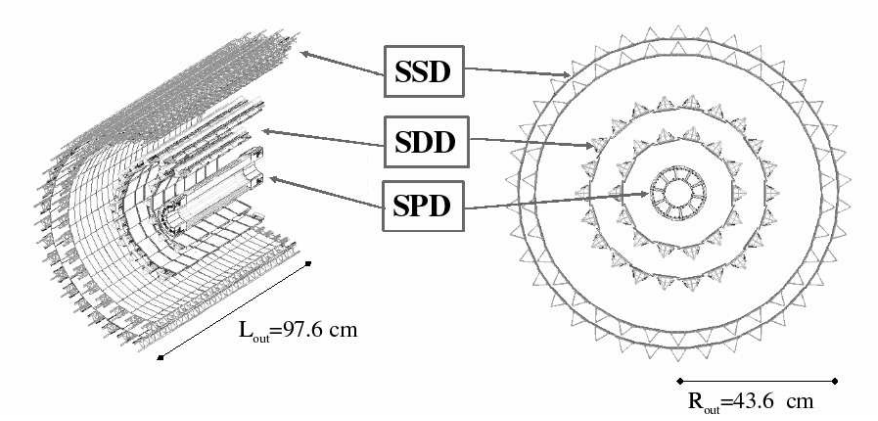

Figure 4.4: ITS layout - is made of 3 subsystems: SPD, SDD, SSD

Coverage in pseudo-rapidity is $|\eta| = 0.9$ and the azimuthal acceptance is non-uniform in a 360^o due to cooling problems in two innermost layers.

Transition Radiation Detector (TRD)

The purpose of the TRD is to provide electrons identification with momenta above 1 GeV/c [71], seen in Figure 4.5. It consists of a radiator of carbon fiber laminated Rohacell/polypropylene which causes electrons to emit transition radiation.

Combined with the tracking properties of a multi-wire proportional chamber attached to the radiating material the TRD can separate pions and electrons with momenta of above 1 GeV/c. The TRD has full azimuthal coverage in $|\eta| < 0.84$.

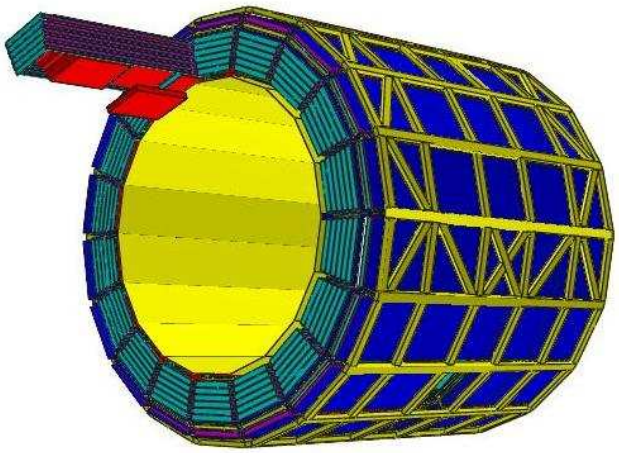

Figure 4.5: TRD layout - contains 18 super modules each with 30 readout chambers (red). The yellow area is the heat shield surrounding TPC. The TOF detector(dark blue) surrounds TRD [71]

Time Of Flight (TOF)

TOF is a sensitive gaseous detector surrounding TRD, see Figure 4.5. It consists of a large array of chambers with high uniform electric field, based on Multi-gap Resistive-Plate Chamber. With this technology, TOF does not experience electrons drift time associated with other types of gaseous detectors [71].

TOF is used for particle identification covering the pseudo-rapidity region $|\eta| < 0.9$ in the intermediate momentum range, below 2.5GeV/c for pions and kaons, below 4 GeV/c for protons with better separation than 3σ of π/K and K/p. When coupled with ITS and TPC, TOF can be used for particle identification, also of vector-meson resonances and open heavy-flavoured states. The time of flight t determined with TOF and particles momentum p and traveled distance l from tracking in ITS, TPC and TRD, the particles mass can be found from:

$$m = p\sqrt{\frac{t^2}{l^2} - 1} \tag{4.1}$$

High Momentum Particle Identification Detector (HMPID)

The HMPID is a detector based on Cherenkov radiation, using a techniques called Ring Imaging Cherenkov (RICH) and consists of 7 modules about $1.5 \times 1.5m^2$. The radiator, is a 15 mm thick layer of low chromaticity C_6F_{14} (perfluorohexane) liquid with an index of refraction of n = 1.2989 at $\lambda = 175nm$. When a charged particle travels in a medium faster than the speed of light (in that medium) $v > c/n$ it emits radiation, called Cherenkov radiation. If one measures the angle of the light cone created by the Cherenkov radiation, than the particle speed traversing through that medium can be determined [72]: $\cos\theta_C = \frac{1}{\beta n}$. In this case, the charged particles traverses the radiator and the radiated photons are catch by the photon counters of HMPID.

The purpose of HMPID detector is to provide particle identification capability beyond the posibilities of ITS, TPC and TOF, by measuring hadrons with transverse momentum $p_t > 1GeV/c$. It covers the pseudo-rapidity range $|\eta| = 0.6$ in the azimuthal range $1.2° < \phi < 58.8°$.

PHOton Spectrometer (PHOS)

PHOS is specialized spectrometer for the detection of electromagnetic radiation over a wide range with high spatial and energy resolutions. It can also discriminate direct photons them from decay photons. The detection is done with lead-tungstate crystals coupled to photo-diodes and preamplifiers. A multi-wire proportional chamber is used for photons identification. It covers $|\eta| = 0.12$ in the azimuthal range $220° < \phi < 320°$.

ElectroMagnetic CALorimeter (EMCal)

The EMCal is a large cylindrical Pb-scintillator sampling calorimeter located between ALICE magnet coils and the central detectors space frame (TPC, TRD, TOF). EMCal enables ALICE to study jet physics and jets interaction with the high dense medium created in heavy ion physics. EMCal in ALICE allows full jet reconstruction for p-p and Pb-Pb collisions even in the presence of high background environment. Due to its fast triggers EMCal can be used as a trigger

for hard jets, photons and electrons. It covers $|\eta| = 0.7$ in the azimuthal range $80° < \phi < 180°$.

ALICE COsmic Ray DEtector (ACORDE)

Due to the underground location of ALICE experiment, a detector designed for the study of muon bundles generated by the cosmic ray primary particles in atmosphere was placed on the top sides of the central ALICE magnet, the ALICE COsmic Ray Detector, [73]. ACORDE is an array of 60 plastic scintillator modules with the a two-fold role in ALICE [71]:

- study of high energy cosmic rays in the region $10^{15} - 10^{17}$ ev
- provide a fast trigger signal, for the commissioning, calibration and alignment procedures of some of the ALICE tracking detectors

4.2.2 MUON Spectrometer

The MUON spectrometer provides additional information for the study of charmonium in the hot, dense medium created in Pb-Pb collisions at ALICE.

The MUON spectrometer shown in Figure 4.6, consists of the following components: frontal absorber for hadrons and photons so they do not make it to the spectrometer, 10 detection planes for high resolution tracking, large dipole magnet, passive muon filter wall to allow only muons to trigger the detector, four planes of trigger chambers and inner beam shield [71].

The minimum muon momentum detected by MUON spectrometer is $4GeV/c$ in full azimuthal angle coverage (360°) and polar angle $171° \leq \theta \leq 178°$. The pseudo-rapidity coverage is $-4.0 < \eta < -2.5$.

4.2.3 Forward detectors

Zero Degree Calorimeter (ZDC)

The purpose of ZDC detector is to detect the number of spectator nucleons carried in the forward direction (at about 0 degrees from the beam direction). This

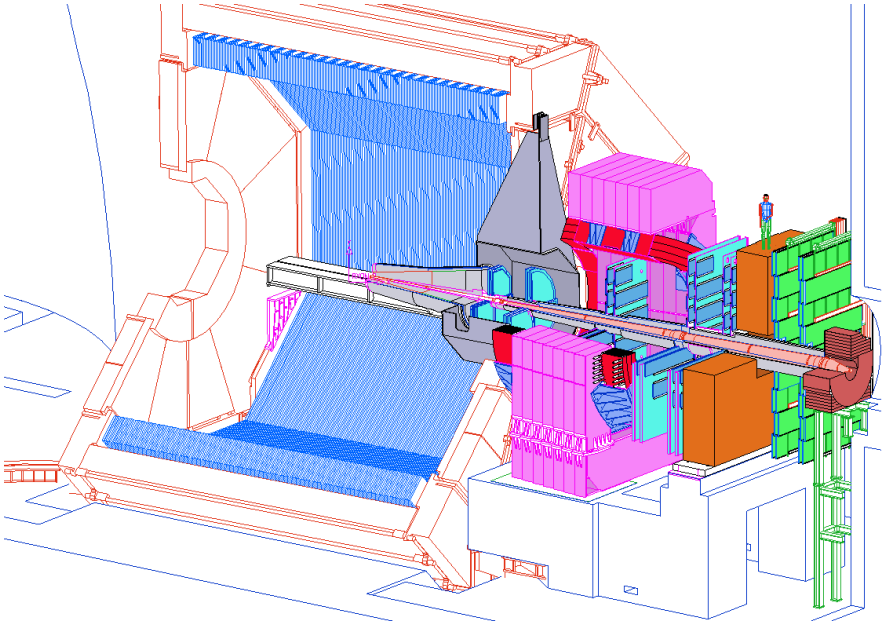

Figure 4.6: MUON spectrometer - identifies muons above about 4 GeV/c and low transverse momentum charmonia at large rapidities

measurement is useful to know the number of participants and so provide information about the centrality of the collision. The energy deposited in ZDC provides necessary information to find the number of participants:

$$E_{ZDC} = E_{NN} \times N_{spectators} \tag{4.2}$$

$$N_{participants} = A - N_{spectators} \tag{4.3}$$

where E_{NN} is the energy per nucleon of the beam and A is the nuclei mass number. This is not valid for very peripheral collisions since the fragments stay in the beam pipes [71]. For this reason, two electromagnetic calorimeters (ZEM) complement ZDC sitting on either side of the interaction point away 7 meters. The purpose of the ZEM is to distinguish between peripheral events where the spectators are large enough to escape the hadron calorimeters and continue in the beam pipe and central events with very few spectators.

Photon Multiplicity Detector (PMD)

The PMD is a preshower detector measuring the multiplicity and spatial distribution of photons in the forward region $2.3 < \eta < 3.7$ [74] with full azimuthal coverage. It can also provide estimates of transverse electromagnetic energy and the reaction plane on a event by event basis. The PMD is located on top of L3, on opposite side of MUON spectrometer and at about 3.64 m from the vertex. The sensitive element of the detector consists of large arrays of gas proportional counters in a honeycomb cellular structure, Figure 4.7.

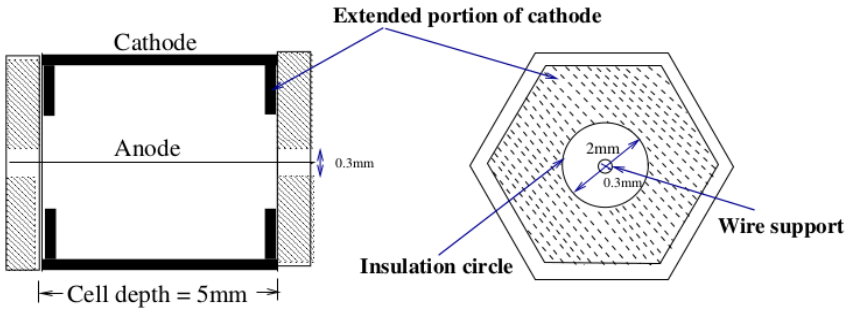

Figure 4.7: Schematic view of a unit cell of the PMD - The detector contains almost 2×10^5 cells, each having an area of $1cm^2$ [71]

The PMD is composed of a veto detector used to discriminate charge particles, a layer of iron and lead that stops the charge particles and permits photons to initiate an electromagnetic shower that produce large signals on several cells, a schematic view is shown in Figure 4.8.

Forward Multiplicity Detector (FMD)

The FMD provides charge particle multiplicity information complementary to SPD. It is composed of 3 subsytems, each with a different number of disks placed at different distances from the interaction point [71]: FMD1, FMD2, FMD3, schematic view shown in Figure 4.9:

The positions of the FMD rings as well as the pseudorapidity coverage for each ring are listed in Table 4.1.

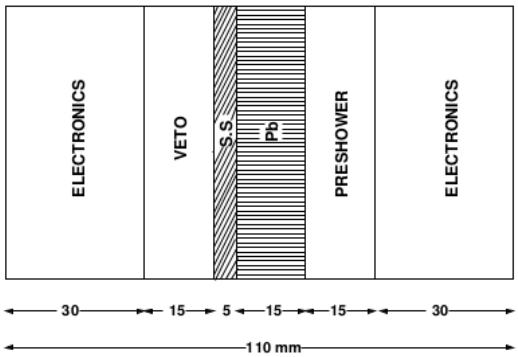

Figure 4.8: Schematic view of cross-section of the PMD - The PMD has vertical symmetry, split in two equal halves [74]

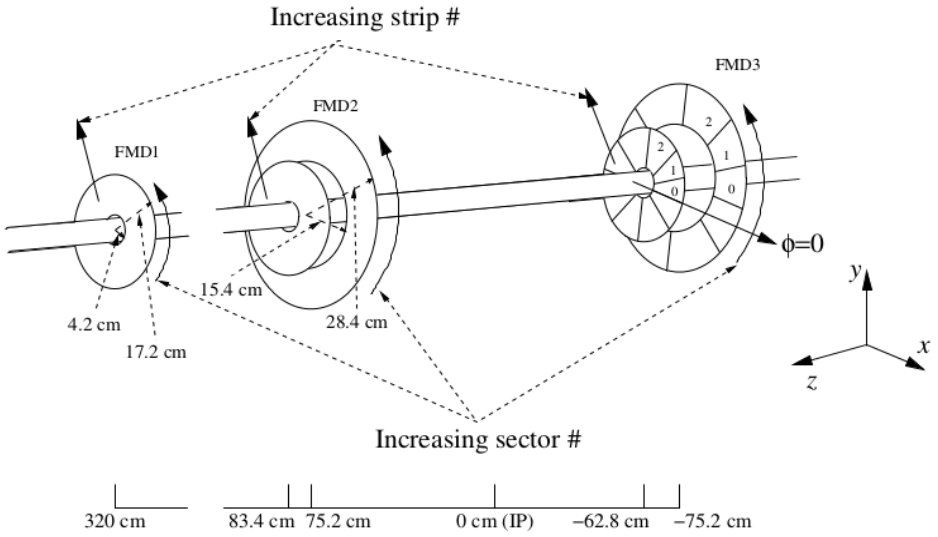

Figure 4.9: Schematic view of FMD - showing the five rings placed around the beam pipe. The MUON arm is on the right [71]

Ring	**z (cm)**	R_{in} **(cm)**	R_{out} **(cm)**	**pseudo-rapiditity** η
FMD1	320.0	4.2	17.2	$3.68 < \eta < 5.03$
FMD2i	83.4	4.2	17.2	$2.28 < \eta < 3.68$
FMD2o	75.2	15.4	28.4	$1.70 < \eta < 2.29$
FMD3o	-75.2	15.4	28.4	$-2.29 < \eta < -1.70$
FMD3o	-62.8	4.2	17.2	$-3.40 < \eta < -2.01$

Table 4.1: FMD rings size, positions and pseudo-rapidity coverage - distance is relative from the detector plane to the interaction point [75]

VZERO (V0)

The VO detector is made of two arrays of scintillator material, located 90 cm from the vertex and 340 cm on the side opposite to the MUON spectrometer from the vertex. Each of the detectors are segmented into 32 elementary counters distributed in 4 rings [71]. The measurement of the time-of-flight difference between the two parts of the detector allows to identify and reject the beam-gas events, thus providing a minimum bias trigger for the central barrel detectors and a validation signal for the muon trigger. The V0 can be used for centrality indicator for Pb-Pb collisions.

Ring	**V0A**		**V0C**	
	η_{max}/η_{min}	$\theta_{min}/\theta_{max}$	η_{max}/η_{min}	$\theta_{min}/\theta_{max}$
0	5.1/4.5	0.7/1.3	-3.7/-3.2	177.0/175.3
1	4.5/3.9	1.3/2.3	-3.2/-2.7	175.3/172.4
2	3.9/3.4	2.3/3.8	-2.7/-2.2	172.4/167.5
3	3.4/2.8	3.8/6.9	-2.2/-1.7	167.5/159.8

Table 4.2: V0A and V0C rings acceptances - in pseudorapidity space and angular (degrees)[75]

TZERO (T0)

The T0 main goals are: vertex measurement with a precision of ± 1.5 cm and a start time for the TOF system, with a time resolution of 50 ps. It can also provide a collision (L0) trigger. It has two subdetectors: T0A and T0C (Table 4.3), each equipped with 12 Cherenkov radiators connected to Photo Multiplier Tubes.

Parameters	**T0-A**	**T0-C**
z (cm)	+375	-72.7
η_{min}/η_{max} (pseudorapidity)	+4.61/+4.92	-3.28/-2.97

Table 4.3: T0-A and T0-C acceptances - distance from the interaction vertex [75]

The time is estimated using the formula [71]:

$$T0 = \frac{T0A + T0C}{2} + T_{delay} \tag{4.4}$$

where T_{delay} is the fixed delay of the analogue mean timer.

Chapter 5

Implementation of flow analysis of simulated data

In this chapter we present the implementation of the elliptic flow analysis algorithm and the software developed for this purpose. Before the implementation of the elliptic flow analysis algorithm, we present two file converters developed and used for the analysis and for a visualization application. These results are taken from [76].

5.1 Input data: UrQMD file

The analyzed data used in this thesis was obtained using the Ultra Relativistic Quantum Molecular Dynamics (UrQMD) simulation code [77]. UrQMD is a microscopic transport model that simulates the time evolution of the hadron-hadron collision using stochastic methods, more about the model and the physics is found in [78].

UrQMD can generate several types of files for simulations: F13, F14, F15, F16, F19, F20. Each of this files is a so call ASCII file, basically a text file contain simulation information. The simulation can be performed in a given time interval $[t_0, t]$ and output simulation calculations at certain time-steps. This gives the evolution of the expanding system in time, much like the picture frames in a movie. For this reason, I call these time-steps: *frames*. The information that is

contained in a file, for example in F14 file is [79]:

- **Events** mass and atomic number of projectile and target, impact parameter, cross section, energy in center-of-mass, maximum simulation time, seed for random number generator, etc.
- **Frames** total number of particles in the current time 'frame', time, number of decays, number of collisions, etc.
- **Particles** particle id, coordinates (x,y,z), momentum (px,py,pz), energy, mass, charge, double isospin projection on z axis, etc

A disadvantage of text files is that they are difficult to traverse randomly for read-write operations. For example, it is difficult to read data from event n and frame j without first reading all previous events *n-1*, frames and particles including the particles from previous *j-1* frames for the interest event. This posed a problem for the application I wanted to write and thus, two applications were developed in order to converter the files into more convenient file types. This converters are presented in the next section.

5.2 File converters

5.2.1 f14tosqlite

The first converter, f14tosqlite is a command line application that converts a F14 UrQMD generated file into a SQLite3 database. SQLite is a light Structured Query Language (SQL) database engine that does not require a server to be run on [80]. The reason for chosen this back-end was mainly related to the easiness of SQL language to query the file, its wide spread use and also due to its easy integration with most of the existing software libraries and toolkits. Figure 5.1 shows its straightforward interface. After running the f14tosqlite over an .f14 file, one obtains a database structured in tables: *events*, *frames*, *particles*. All data from f14 file is contained in these tables. This database can be read with any software that is capable of reading SQLite3 databases. As an example, Figure 5.2 shows the application *SQLite Database Browser* interrogates the converted file.

```
mihai@mihai-M17xR3:~/work/Docs/PhD_Thesis$ f14tosqlite -h
====================== Acknowledgement ======================================
This application was made possible by the work for Phd at
University of Bucharest, Faculty of Physics through program
Burse Doctorale from Romanian Ministry of Labor, Family and Social Protection
Project POSDRU/88/1.5/S/56668
==============================================================================

About: This application converts a UrQMD .f14 file to a sqlite database file

f14tosqlite version 0.2 Licensed under GPL v2
Usage: f14tosqlite -f file.f14 [-o output.db] [-n events]
Options
:       -f      specify the input f14 file
        -o      specify the output file
                default: output.db
        -n      get only the first n events
                default: 0 - all
        -h      show this help
Copyright Mihai Niculescu <q.quark@gmail.com> 2009
```

Figure 5.1: f14tosqlite command line interface - The application is made publicly available at [81].

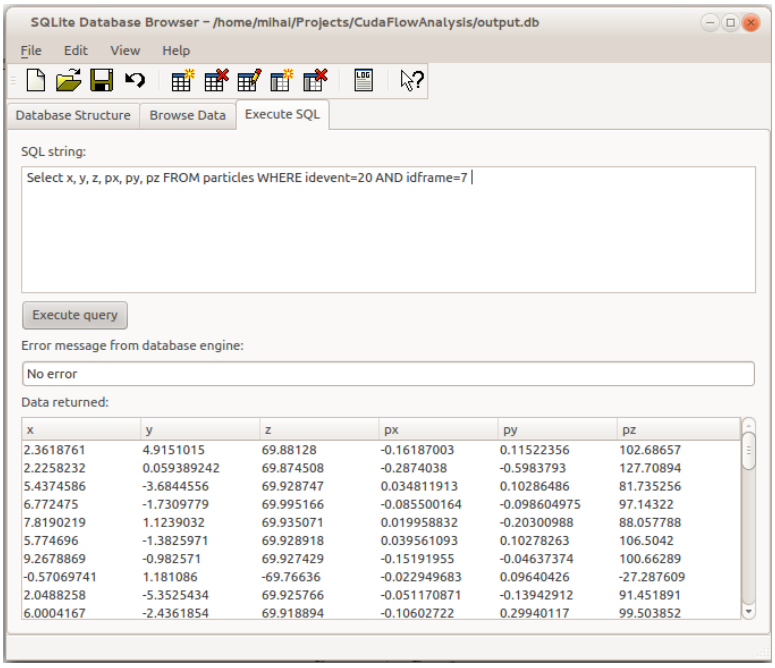

Figure 5.2: SQLite Database Browser - querying a database file converted with f14tosqlite

5.2.2 f14toROOT

This application converts f14 files into a ROOT file containing a ROOT Tree, named "dataTree". The Tree is structured in 3 branches containing the objects as leafs: *sqfEvent*, *sqfFrame* and respectively *sqfParticle*. To be able to read the converted ROOT file, one needs ROOT installed [10] and the library created when installing this application. f14toROOT command line interface is the same as of f14tosqlite, Figure 5.1.

The application is freely available for download in the public domain (public license) at the project site [82].

5.3 Visualization

Due to the increased GPU's performance power and the fact the analysis algorithm runs on graphic video card, it was a step closer to provide visualization of the particle system. Another important factor for providing animation was that the generated UrQMD F14 file provides data for the expansion of the system at different time 'frames'. Figure 5.4 shows the main application window which in the center of the window is rendered a Au+Au collision at $\sqrt{s_{NN}} = 200$ GeV.

The application is made of multiple libraries and toolkits. The Graphical User Interface (GUI) is designed using a cross platform toolkit, called Qt [83]. Besides GUI elements, Qt was useful for accessing SQLite databases created with f14tosqlite, by providing its own Application Programming Interface (API) implementation. Another area in which Qt proved useful was for context interface for the 3D visualization. These features helped for easy interface the code with different libraries. The 3D visualization was done in OpenGL, a cross platform graphics library used world wide in many visualization applications ranging from video games to computer assisted designers (CAD) [84]. The computation of the elliptic flow is done on GPU using CUDA, see Section 2.2.1 and the results are plotted using ROOT 1.3.1. The elliptic flow analysis algorithm runs simultaneous with the visualization while rendering is continuously providing smooth frames per second even on modest hardware (tests were performed on a NVIDIA graphic board GeForce 9600 GT), a recorded video is found on [85].

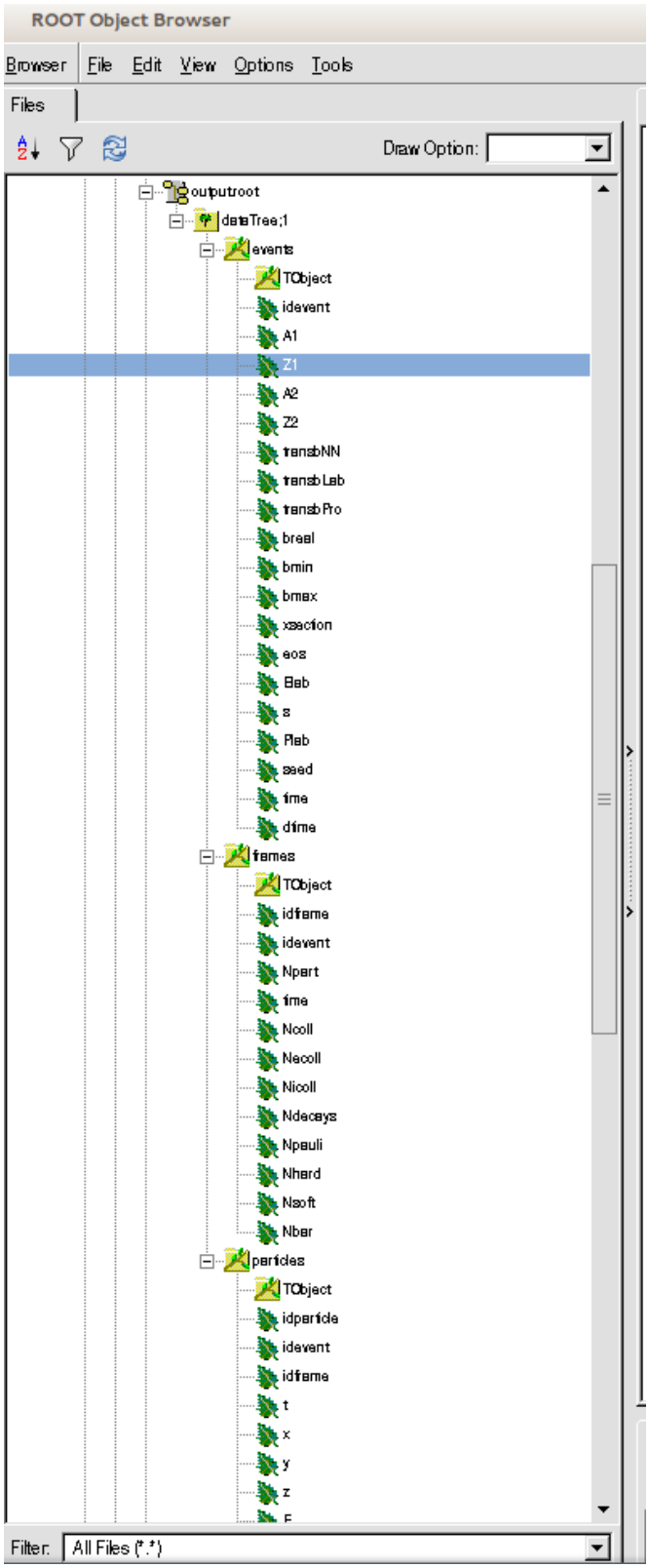

Figure 5.3: Structure of ROOT file generated with f14toROOT -

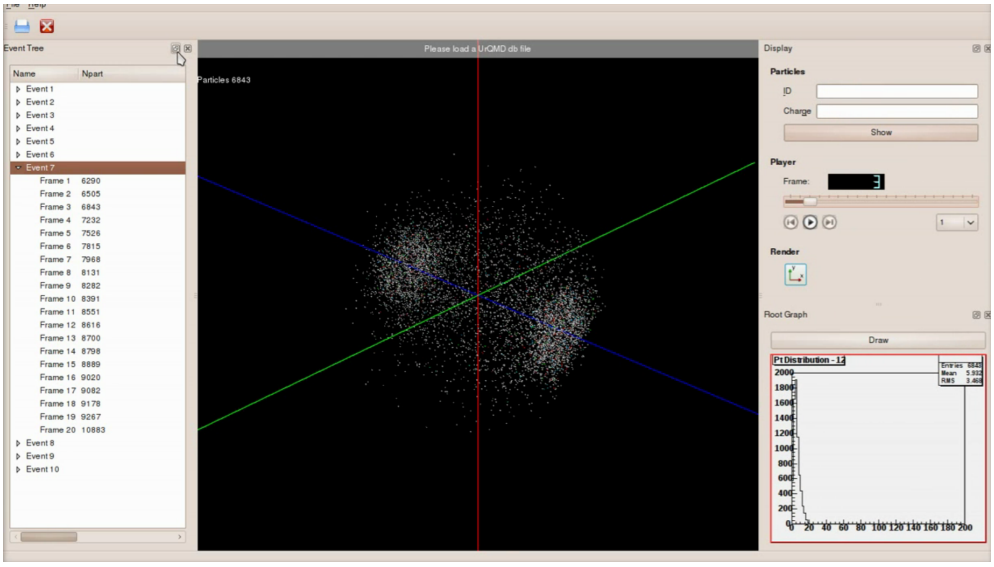

Figure 5.4: UrQMD Event Viewer - Left side: the list of events and frames from database; Center: 3D perspective view of the interacting system; Right: real-time particle filter (top), frame selection in the current event (middle), transverse momentum distribution of all particles in the current time frame (bottom)

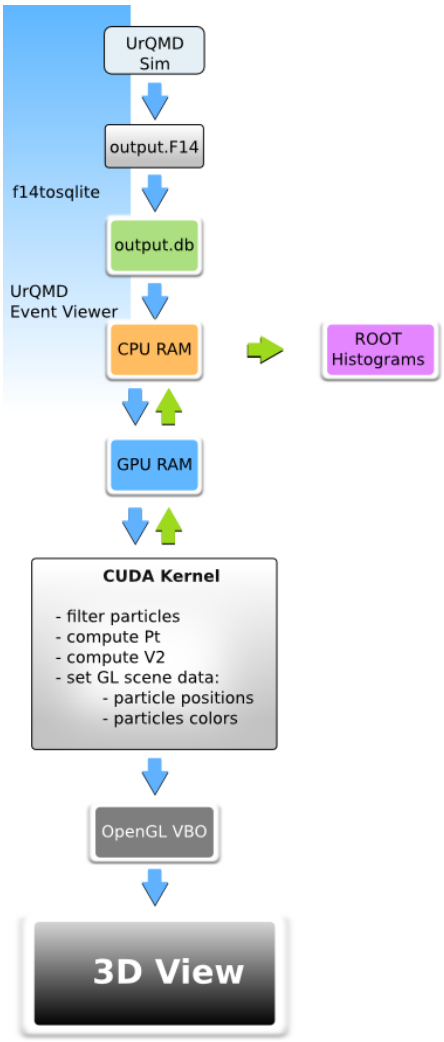

Figure 5.5: UrQMD Event Viewer Data flow - schematic view

Chapter 6

Simulations and Results

6.1 Flow reconstruction from UrQMD data

The differential flow v_2 has been studied for lead-lead collisions at $\sqrt{s_{NN}} = 2.76GeV$ using data generated by UrQMD simulation code and compared with ALICE published results.

6.1.1 Simulation details

The UrQMD version used for the simulations was 3.3 with updated patches and modified, in order to support LHC energies, according to the UrQMD user manual [79].

The simulations were done for two data sets:

- **Test Data set** a set of 1000 simulated p-p collisions and 1000 Au-Au with impact parameter varied between **0-12 fm** at $\sqrt{s_{NN}} = 200GeV$. This set was used as a test for the flow analysis algorithm and to get a feeling of the flow harmonics directed flow v_1 and elliptic flow v_2. The simulation contained data for the expansion of the fireball at every time step $dt = 10fm/c$ till a time after freeze-out *200 fm/c*, need by the UrQMD Event Viewer presented in Section 5.3

- **Case study Data set** For the study of the elliptic flow, were simulated 26K events of Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76GeV$ UrQMD. The impact

parameter was chosen to correspond with the centrality classes used by ALICE in the published results for central and midcentral collisions. In order to limit the simulation time, the simulations were done only after freeze-out at time-step 200 fm/c.

Impact parameter (b)

For the study case in Section 6.1.1, it was need it to determine the impact parameter for the centrality classes presented by ALICE in [56].

The impact parameter was calculated with Glauber Monte Carlo [86], summarized in Table 6.1.

Centrality (%)	**Impact parameter b (fm)**
0 - 5	0 - 3.29
30 - 40	7.4 - 11.02

Table 6.1: Centrality intervals and impact parameter - for Pb-Pb $\sqrt{s_{NN}} =$ 2.76TeV. Taken from [86]

6.1.2 Azimuthal distribution of particles

The azimuthal distribution of particles is given by Eq. 3.7 or Eq. 3.14. The azimuthal angle of generated particles is shown in Figure 6.1. From the frame 2 fm/c to 64 fm/c, the fluctuations in the ϕ distribution attenuates. This shows that initially, near the collision vertex (both in time and space), there is an uneven distribution of the particles while near the end, the expanded system will continue to contain particles distributed uniform in all angles. In a space coordinates system, this translates that all particles are coming into detectors from all angles from the interaction point.

Figure 6.2, shows the azimuthal distribution, as given by Eq. 3.14, in polar coordinates considering only the first 2 harmonics. It proves the definition of v_2 as elliptic flow, the azimuthal distribution in polar coordinates is an ellipse when the second harmonics is non-zero.

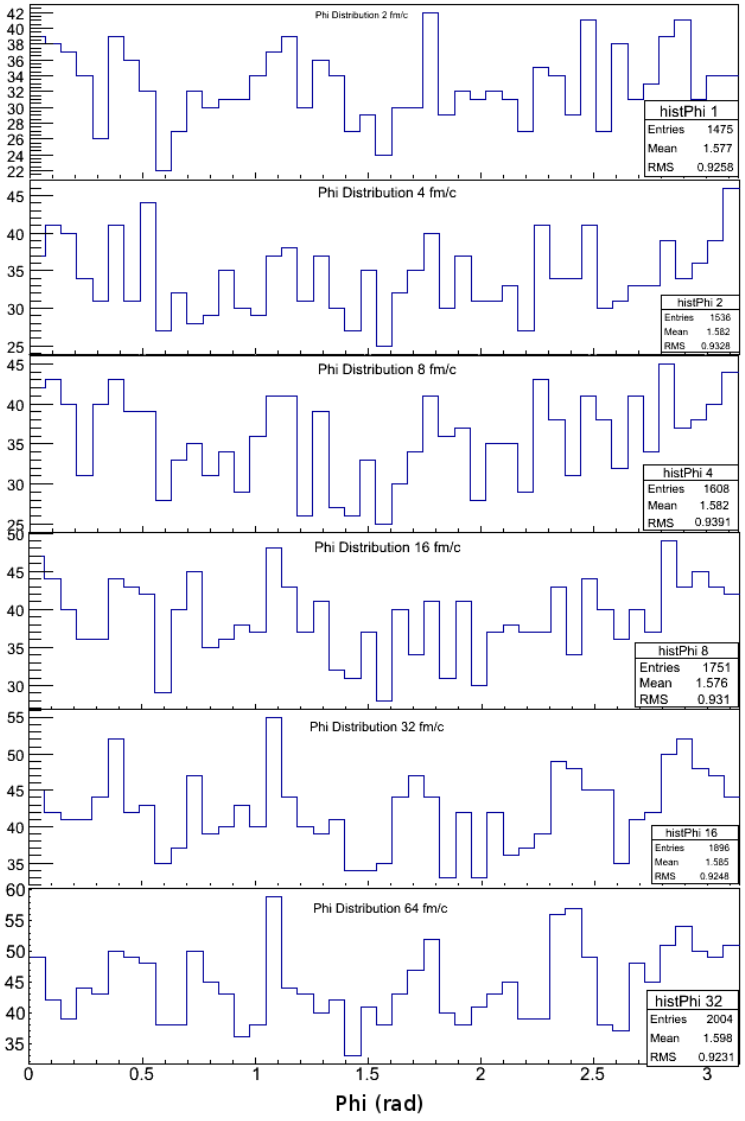

Figure 6.1: Phi Distribution of particles - at different time-steps (frames), for all particles in $\phi \in [0, \pi]$

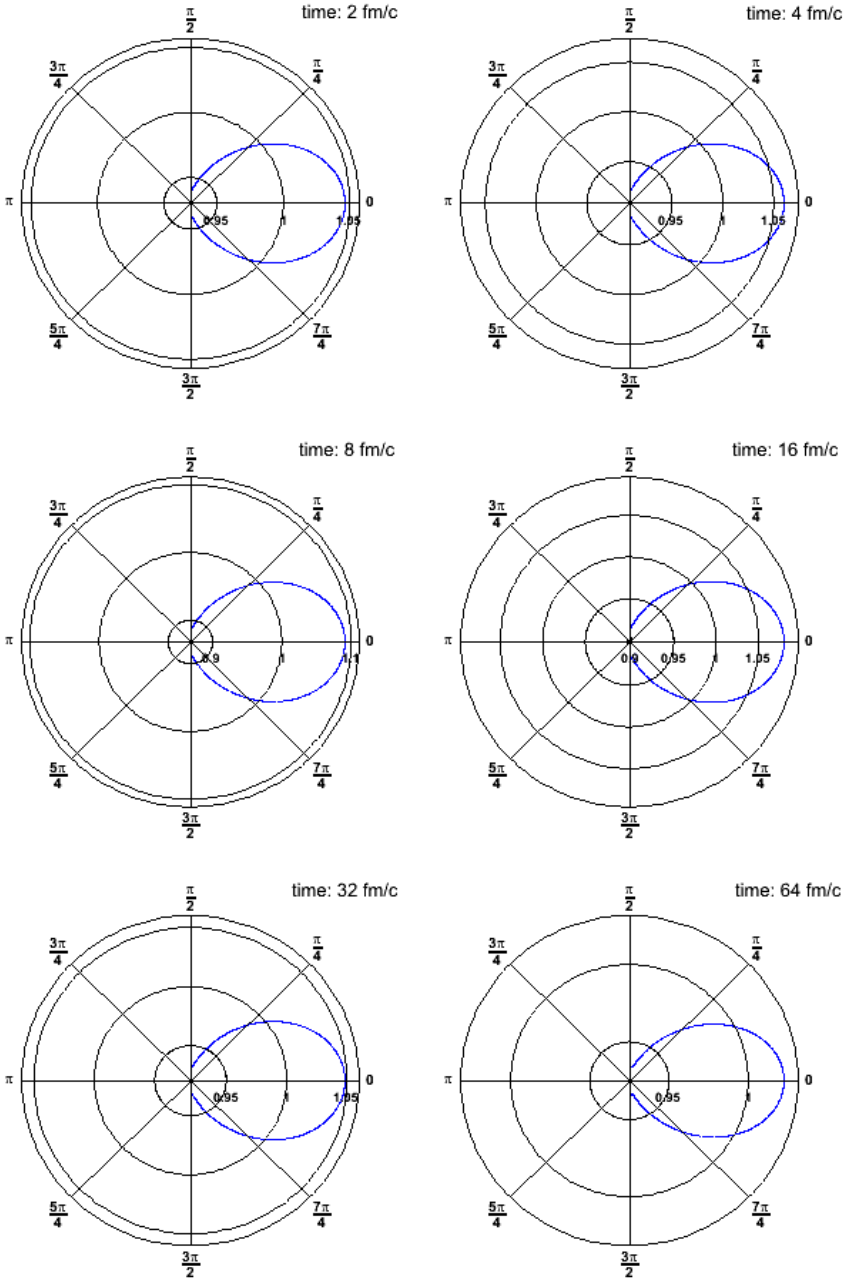

Figure 6.2: Azimuthal distribution $r(\phi)$ in polar coordinates - at different time-steps (frames), for all particles

6.1.3 Differential elliptic flow of charged particles

According to the event plane method Sec. 3.5, the elliptic flow $v_2(p_t)$ is obtained by dividing the observed v_2^{obs} by the event plane resolution:

$$v_n = \frac{v_n^{obs}}{R}$$

with the observed flow harmonic,

$$v_n^{obs} = \langle \cos[n(\phi - \psi_R)] \rangle_i$$

where the event plane resolution is:

$$R = \langle \cos[n(\psi_n - \psi_R)] \rangle_i$$

Since, UrQMD generates the reaction plane angle equal to zero $\psi_R = 0$, can be seen in Fig. 6.3, it results also the event plane is zero $\psi_{EP} = \psi_R$. And so, the observed flow harmonics are equal to the corrected flow harmonics due to event plane resolution being equal to 1:

$$v_n = \quad v_n^{obs} = \langle \cos n\phi \rangle_i \tag{6.1}$$

$$R = \qquad 1 \tag{6.2}$$

Although, the reaction plane angle is zero in UrQMD, the analysis method used to calculate the differential flow v_2 did the computation of this angle, and found to be, evidently, zero. This was required for the benchmark results presented in the next sections.

Flow harmonics per particle can simply be related to particles momentum:

$$v_1 = \quad \frac{p_x}{p_T} \tag{6.3}$$

$$v_2 = \quad \frac{p_x^2 - p_y^2}{p_T^2} \tag{6.4}$$

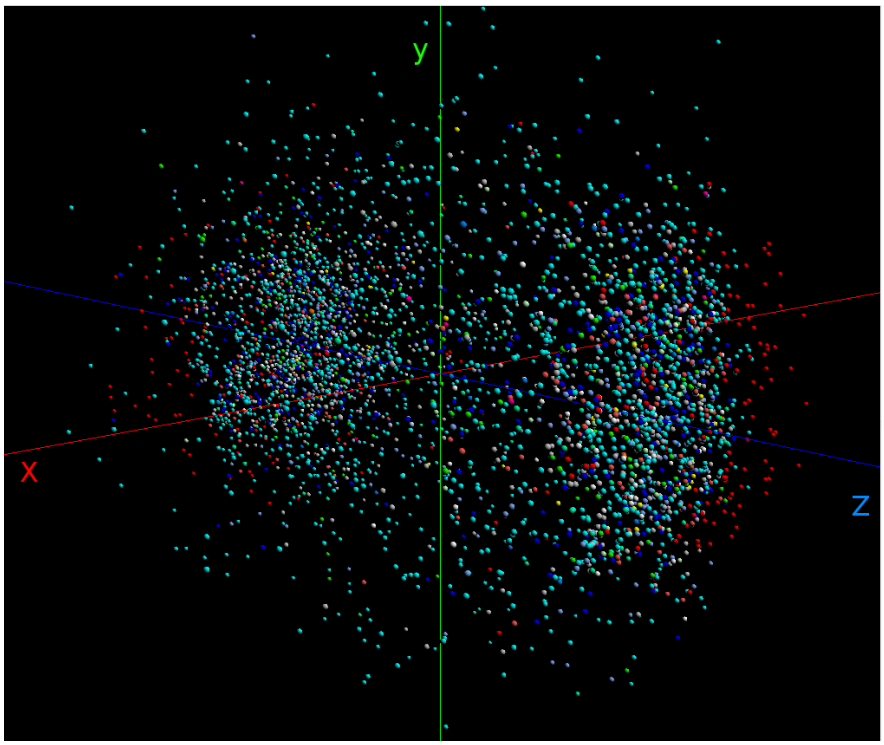

Figure 6.3: A 3D perspective view of UrQMD - The reaction plane angle is zero. It can be seen a Au-Au collision at $\sqrt{s_{NN}} = 200GeV$, $b = 4.66fm$ at time-step t=10fm/c

6.1.4 Comparison of elliptic flow from UrQMD with ALICE results

Here we present the analysis results of elliptic flow from Case Data set, see section 6.1.1 and compare with elliptic flow obtained by ALICE for Pb-Pb at $\sqrt{s_{NN}} = 2.76$ TeV, published ALICE results in [56].

This study was done in the pseudorapidity interval $\eta \in [-1, 1]$ and transverse momentum $p_t < 5.0$ GeV in the central (0%-5%) and peripheral (30%-40%) centrality classes. The impact parameter for these centrality classes is determined according to the discussion in Section 6.1.1. The differential flow v_2 is done using event plane method as described in Section 3.5.

In order to have a gain in resolution and to enhance the contribution of particles with higher flow, the weight coefficients used in the calculation were set according to:

$$w_i(p_t) = \begin{cases} 0.5 \cdot p_t & \text{if } p_t < 2 \text{ GeV} \\ 1 & \text{if } p_t \geq 2 \text{ GeV} \end{cases} \tag{6.5}$$

Figure 6.4 shows the differential elliptic flow v_2 as a function of p_t for central collisions while Figure 6.5 for peripheral collisions. The errors are statistical $v_2^i/\sqrt{N_{bin}}$, shown as vertical bars in the figures, while the horizontal lines represent bin errors. Comparing the obtained results Figure 6.5 with the ideal hydrodynamics model calculations from ALICE, Figure 6.6, the differential v_2 from UrQMD shows a suppression from ideal hydrodynamics up to 25%. While comparing with ALICE results, elliptic flow in UrQMD for midcentral collisions is underestimated presenting a suppression up to 11 %. This is more pronounced in central collisions (0-5%), where UrQMD underestimates v_2 up to 21%.

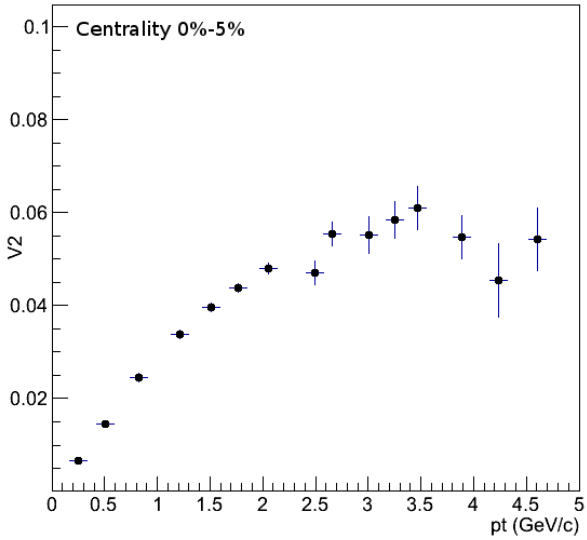

Figure 6.4: Elliptic flow as a function of transverse momentum - for Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV and $|\eta| < 1$

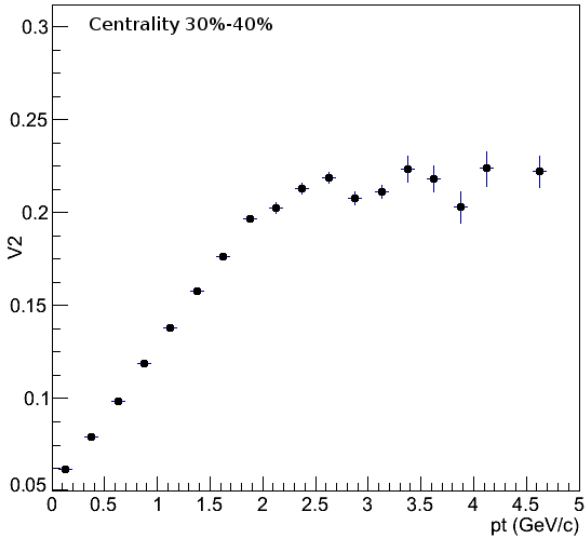

Figure 6.5: Elliptic flow as a function of transverse momentum - for Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV and $|\eta| < 1$

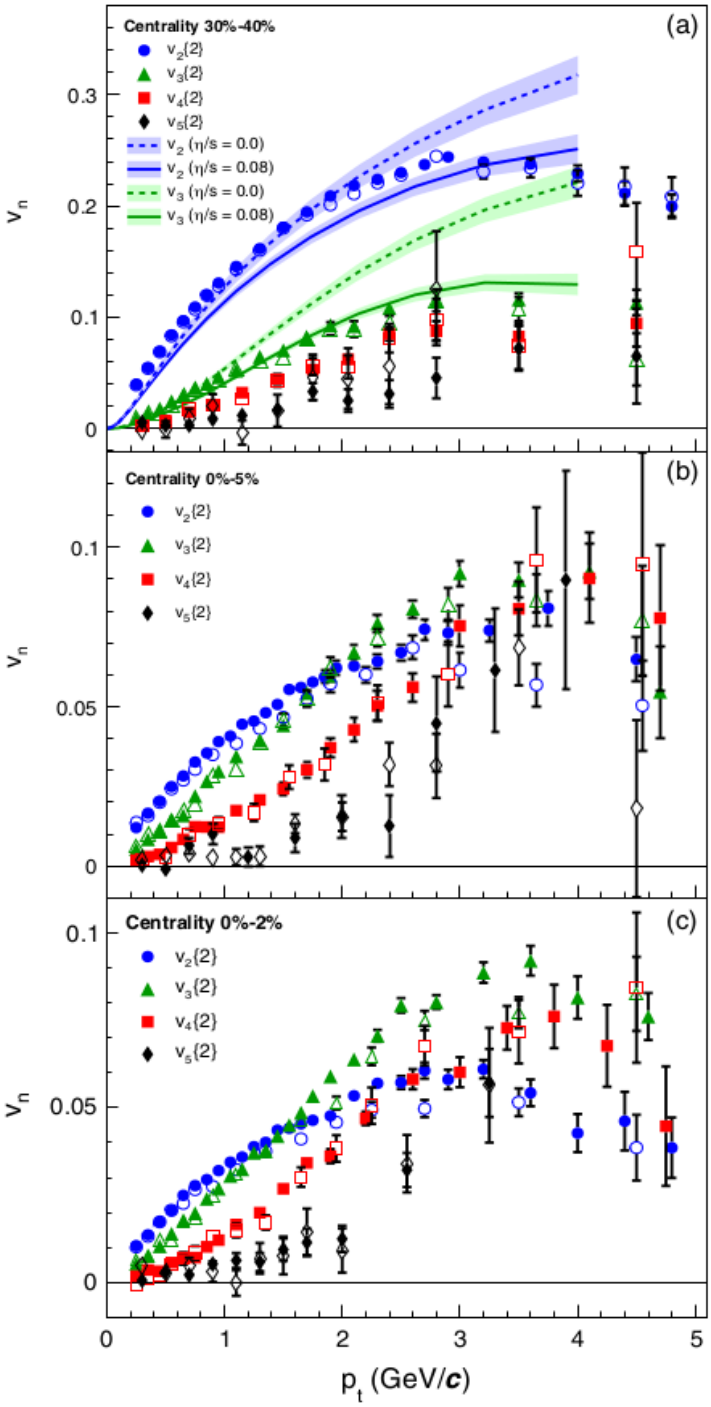

Figure 6.6: Flow harmonics v_2, v_3, v_4, v_5 at ALICE as a function of transverse momentum for three centralities - for Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76$ TeV. Fig.(a) compared to ideal hydrodynamics. Taken from [56]

6.2 Performance benchmarks

In this section we present the benchmark results of flow analysis implementation in CUDA versus the same algorithm on CPU.

The benchmarks were done on a Alienware M17 laptop running Ubuntu 12.04. The general features of the laptop's components, used in benchmarking:

- **CPU** Intel Core i7 2670QM @ 2.2GHz, 8 processors each with 4 cores, HyperThreading enabled
- **GPU** NVIDIA GeForce GTX 580M, 2 GB DRAM, 384 CUDA cores

Timing was implemented for different parts of the code, a class timer based on C timing functions - used for the CPU part, while timing on GPU was achieved using CUDA streams. The total execution time was measured using the **time** command from bash terminal. Table 6.2 shows the obtained results between CPU versus GPU running times for the flow analysis algorithm.

	Run (s)	**Data load (s)**	**Transfer (ms)**	**Computation (ms)**
CPU	8.238	7.43	n/a	430
GPU	8.193	7.43	48.207	3.2567

Table 6.2: Benchmark of flow algorithm CPU vs GPU - Amdahl's law in action

The obtained speedup, measured as time on GPU/CPU is shown in Table 6.3. As expected the gain in computation on GPU is bigger than on CPU, but the question that rises is why the this is not seen in the total run time of the application.

	Total	**Computation**
Speedup (GPU/CPU)	1.00549x	132x

Table 6.3: Speedup gain of flow analysis algorithm - as expected GPU computation is faster than CPU

One answer is given by Amdahl's law, Equation 2.1: the parallel part of the code is smaller than the serial part. This explanation is already presented quantitative in Table 6.2, where column **Data load** represents actually Input/Output

operations done by ROOT, which are serial, while the computation time is parallel part (when on GPU). In order to better understand this, another test was done using a tool specialized in performance profiling for parallel applications: Intel's Advisor XE Beta 2012 [87]. The *Survey Report* generated with this tool shows places in the code where the applications spends time. The report is shown in Figure 6.7.

The survey report confirms that the application is IO bound and furthermore, it shows exactly the functions that spend most time. To be more precise, the places where the application spends most time is in de-serialization and unzipping of the objects from ROOT's class TTree.

It is clear that in order to gain a bigger total speedup, more parallel code must be implemented and precisely IO must be parallelized. However, this is not easy to achieve in ROOT since TTree class in not parallelized and the way ROOT handles IO using global variables. Thus, this becomes an impossible task.

In spite of the fact that IO can not be parallelized, and even if the time spend in IO is reduced using other type of files (database, xml, binary files, etc.), the total speedup would remain about the same because of Amdahl's law. Following Amdahl's law suggestion is to improve the ratio *Computation/IO*. In other words, another solution would be that the algorithm must do more computation.

Figure 6.7: Advisor Survey report of flow analysis algorithm -

Chapter 7

Summary and Conclusions

In this thesis we tried to present the impact of the new technologies in the large field of heavy ion collisions. As a case study, we implemented on GPU the event plane method and using the developed code we compared the elliptic flow generated by UrQMD with ALICE published data.

We started with the context in which the subject of this thesis was born, the high computing requirements of high energy physics experiments. We seen how big is the scale size of the computing requirements in high energy physics experiments and how they are satisfied at ALICE experiment, as presented in Chapter 1. Since the experiments at LHC tend to increase the frequency of acquisition rate and beam luminosities this will mean more events need to be stored and more data needs to be processed. As the computing requirements grow, software must keep the step with technology changes.

In Chapter 2, we seen the reason behind processors evolution into multi-core processors and how the software and hardware industry was forced to adapt to this change. One of the most remarkable processors, the graphic processor brought new hopes for high performance computing on commodity hardware. This created interest in scientific community in all research fields and high energy physics was not left outside of this trend. A small survey, but relevant, of utilization of the graphic processor in HEP was made and presented in Section 2.2. We have seen how the use of GPU can bring a speedup gain of almost 137x as in the case of NA62 experiment. This survey was useful to understand how these massive parallel architectures are already used in HEP and what expectations are there.

We raised the question if the GPUs can be used in HEP for offline data analysis and to get this answer we decided it is best to use a real world data analysis example. Due to the wonderful physics behind the theory of anisotropic flow, we took this subject as a case study. An introduction to Heavy Ion Physics and the theory of anisotropic flow was given in Chapter 3. Also, the flow analysis method used in this thesis is presented in section 3.5.

Since, the simulation code used to obtain data, UrQMD is able to simulate the expansion of the fireball at different stages, or time frames, and the analysis program is already running on GPU, we thought it is just a step closer in order to write a 3D visualization application of the particles from time frames. Other applications needed to avoid the hassle of working with text files generated by UrQMD.were written, presented in Chapter 5. The f14tosqlite file converter was a good decision for the 3D visualization application since it was easy to interrogate the database with SQL queries and to provide a standalone application without the use of other external toolkits except Qt. While the other file converter, f14toROOT was useful in order to provide ROOT files much like the real world ESDs files. This was useful for the performance tests presented in Chapter 6.

In Chapter 6, we present the results of the elliptic flow analysis and the performance benchmarks. We have seen that elliptic flow in UrQMD for mid-central collisions is underestimated presenting a suppression up to 11 %. This is more pronounced in central collisions (0-5%), where UrQMD underestimates v2 up to 21%. But in general, UrQMD predictions are within 5-10%.

On the other side, GPU vs CPU benchmark shows an insensible total speedup, although the computation part on GPU is 132x faster than on CPU. We blame Amdahl for this and say the application is IO bound. In order to get a better total speedup, it must be necessary to parallelize the IO part of the code. Since this is not possible with the current version of ROOT another solution would be to do more computation inside the CUDA kernel. Let's say the algorithm should compute in a single kernel more flow harmonics v_1, v_2, v_3, v_4, v_5 calculated with different flow analysis methods, beside event plane method.

As a final conclusion, a single and simple algorithm as the event plane method on a highly parallel architecture as the GPU is far too less work for the graphic processor compared with the time CPU needs to load the data. The main debate

is not CPU vs GPU, but this comparison shows what can be achieved with parallelization.

References

[1] Leon Levitchi. **English translation of M. Eminescu - First Epistle**, 2012. Available from: `http://www.fa-kuan.muc.de/SCRISO1.HTML`. iii

[2] CERN web. **LHC at CERN**, 2012. Available from: `http://public.web.cern.ch/public/`. vi, 39

[3] Particle Data Group. **PDG webpage**, 2012. Available from: `http://pdg.lbl.gov/`. 2

[4] ATLAS. **ATLAS website**, 2012. Available from: `http://www.atlas.ch/news/2012/latest-results-from-higgs-search.html`. 2, 38

[5] CMS. **CMS website**, 2012. Available from: `http://cms.web.cern.ch/news/observation-new-particle-mass-125-gev`. 2, 38

[6] ALICE Collaboration. **Technical Design Reports**, 2005. Available from: `http://aliceinfo.cern.ch/Documents/TDR/Computing.html`. 4, 5, 6

[7] LHC Computing Review. **CERN/LHCC/2001-004**, 2001. 4

[8] ALICE Collaboration. **Technical Design Report of Trigger, Data Acquisition, High-Level Trigger and Control System**. 4

[9] ALICE Collaboration. **Physics Performance Report, Volume 1**. *J. Phys. G: Nucl. Part. Phys. 30*, 2004. 4

[10] ROOT System. **ROOT webpage**, 1995. Available from: `http://root.cern.ch/`. 6, 7, 54

[11] ALICE Offline. **ALICE Offline page**, 1998. Available from: http://aliweb.cern.ch/Offline/. 6

[12] ROOT Team. **PROOF webpage**, 2012. Available from: http://root.cern.ch/drupal/content/proof. 8

[13] ALICE Analysis Facilities. **List of CAFs**, 2012. Available from: http://alimonitor.cern.ch/stats?page=PROOF/list. 9

[14] I. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, C. Dobre, A. Muraru, A. Costan, M. Dediu, and C. Stratan. **MonALISA: An agent based, dynamic service system to monitor, control and optimize distributed systems**. *Computer Physics Communications*, **180**(12):2472 – 2498, 2009. 40 YEARS OF CPC: A celebratory issue focused on quality software for high performance, grid and novel computing architectures. Available from: http://www.sciencedirect.com/science/article/pii/S0010465509002410. 10

[15] MonALISA. **MonALISA webpage**, 2012. Available from: http://alimonitor.cern.ch. 10

[16] Globus Team. **Globus webpage**, 2012. Available from: http://www.globus.org/toolkit/. 10

[17] gLite Team. **gLite webpage**, 2012. Available from: http://www.glite.cern.ch. 10

[18] ARC Team. **ARC webpage**, 2012. Available from: http://www.nordugrid.org/arc/. 10

[19] Gordon E. Moore. **Cramming more componentsonto integrated circuits**. *Electronics*, **38**(8), 1965. Available from: http://download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf. 12

[20] Andrzej Nowak and OpenLab. **OpenLab webpage**, 2012. Available from: http://openlab.web.cern.ch/. 13

[21] M.Flynn. **Some Computer Organizations and Their Effectiveness**. *EEE Trans. Comput.*, **21**(398), 1972. 12

[22] Greg Ippolito. **POSIX thread (pthread) libraries**, 2002-2012. Available from: `http://www.yolinux.com/TUTORIALS/LinuxTutorialPosixThreads.html`. 14

[23] GNU team. **The GNU Fortran Compiler**. Available from: `http://gcc.gnu.org/onlinedocs/gfortran/OpenMP.html#OpenMP`. 15

[24] ANL. **The Message Passing Interface (MPI) standard**. Available from: `http://www.mcs.anl.gov/research/projects/mpi/`. 15

[25] CRAY. **The Chapel Parallel Programming Language**. Available from: `http://chapel.cray.com/index.html`. 15

[26] INTEL. **Intel Threading Building Blocks**. Available from: `http://threadingbuildingblocks.org/`. 15

[27] OpenHMPP Consortium. **OpenHMPP Webpage**, 2012. Available from: `http://www.openhmpp.org/en/OpenHMPPConsortium.aspx`. 15

[28] NVIDIA. **NVIDIA Cuda C programming guide**, 2012. Available from: `http://www.nvidia.com/object/cuda_home_new.html`. 15, 16, 18

[29] AMD/ATI. **AMD Stream webpage**, 2012. Available from: `http://developer.amd.com/tools/hc/AMDAPPSDK/Pages/default.aspx`. 15

[30] Khronos Group OpenCL Consortium. **OpenCL webpage**, 2012. Available from: `http://www.khronos.org/opencl/`. 15, 16

[31] M.Niculescu and S.I. Zgura. **Computing trends using graphic processor in high energy physics**. 2011. `arXiv:1106.6217`. 17

[32] NA62 Collaboration. **NA62 experiment webpage**, 2007. Available from: `http://na62.web.cern.ch/na62/Home/Home.html`. 17

[33] Gianluca Lamanna, Gianmaria Collazuol, and Marco Sozzi. **GPUs for fast triggering and pattern matching at the CERN experiment NA62**. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **639**(1):267–270, 2011. Available from: http://dx.doi.org/10.1016/j.nima.2010.09.167. 19

[34] PANDA Collaboration. **Panda experiment webpage**. Available from: http://www-panda.gsi.de/. 20

[35] FairROOT. **FairROOT webpage**. Available from: http://fairroot.gsi.de/. 20

[36] M Al-Turany, F Uhlig, and R Karabowicz. **GPU's for event reconstruction in the FairRoot framework**. *Journal of Physics: Conference Series*, **219**(4):042001, 2010. Available from: http://stacks.iop.org/1742-6596/219/i=4/a=042001. 20

[37] CBM Collaboration. **CBM experiment webpage**. Available from: http://www.fair-center.eu/en/fair-users/experiments/cbm.html. 21

[38] Matthias Bach. **SIMT Kalman Filter - High throughput track fitting**, 2010. Available from: https://www.gsi.de/documents/DOC-2010-Jun-125-1.pdf. 21

[39] G. S. Bali. **QCD forces and heavy quark bound states**. *Phys. Rept.*, **343**(1):1–136, 2001. 23

[40] RHIC at Brookhaven National Laboratory. **RHIC experiment webpage**, 2012. Available from: http://www.bnl.gov/rhic/. 23

[41] J. C. Collins and M.J. Perry. **Superdense Matter: Neutrons or Asymptotically Free Quarks?** *Phys. Rev. Lett.*, **34**(1):1353, 1975. 24

[42] F. Karsh. **A698**. *Nucl. Phys.*, **34**(199), 2002. 26

[43] M. Stephanov. **Phase diagram of QCD: the critical point**, 2009. Available from: http://www-alt.gsi.de/documents/DOC-2009-Mar-47-1.pdf. 26

[44] Y.Aoki et al. *Nature*, **443**(675), 2006. 27

[45] I.Arsene et al. (BRAHMS Collaboration). *Phys. Rev. Lett.*, **91**(072305), 2003. 28

[46] J.Adams et al. (STAR Collaboration). *Phys. Rev. Lett.*, **91**(072304), 2003. 28

[47] J.W. Cronin et al. *Phys. Rev. Lett.*, **D11**(3105), 1975. 27

[48] (ALICE Collaboration). *Phys. Rev. Lett.*, **B**(696):30–39, 2011. 28

[49] L.D.Landau. *Izv.Akad. Nauk Ser. Fiz.*, **17**(51), 1953. 29

[50] J.Sollfrank P.F.Kolb and U.Heinz. *Phys. Rev.*, **C**(054909), 2000. 29

[51] U.Heinz. **Early collective expansion: Relativistic hydrodynamics and the transport properties of QCD matter**. 2009. arXiv:0901.4355. 30, 32

[52] J.Brachmann, S.Soff, A.Dumitru, H.Stocker, J.A. Mahruhn W.Greiner, L.V.Bravina, and D.H.Rischke. *Phys. Rev C*, **61**(024909), 2000. 32

[53] STAR Collaboration. **Elliptic Flow in Au-Au Collisions at $\sqrt{s_{NN}} = 130GeV$**. *Phys.Rev.Lett*, **86**(3), 2001. 33

[54] S.S. Adler, et al, and PHENIX Collaboration. **Elliptic Flow of Identified Hadrons in Au+Au Collisions at $\sqrt{s_{NN}} = 200GeV$**. *Phys.Rev.Lett*, **91**(182301), 2003. 33

[55] S.S. Adler, et al, and PHENIX Collaboration. **Measurements of Higher Order Flow Harmonics in Au+Au Collisions at $\sqrt{s_{NN}} = 200GeV$**. *Phys.Rev.Lett*, **107**(252301), 2011. 33

[56] ALICE Collaboration. **Higher Harmonic Anisotropic Flow Measurements of Charged Particles in Pb-Pb Collisions at $\sqrt{s_{NN}} = 2.76TeV$**. *Phys.Rev.Lett*, **107**(032301), 2011. 33, 59, 64, 66

[57] ALICE Collaboration. **Elliptic Flow of Charged Particles in Pb-Pb Collisions at $\sqrt{s_{NN}} = 2.76TeV$**. *Phys.Rev.Lett*, **105**(252302), 2010. 33

[58] ALICE Collaboration. **Anisotropic flow of charged hadrons, pions and (anti-)protons measured at high transverse momentum in Pb-Pb collisions at $\sqrt{s_{NN}} = 2.76TeV$**. `arXiv:1205.5761v1`. 33

[59] G.Aad, et all, and ATLAS Collaboration. *Phys.Rev.Lett B*, **707**(330), 2012. 33

[60] S. Chatrchyan, et al, and CMS Collaboration. `arXiv:1204.1409`. 33

[61] Burak Han Alver, Clément Gombeaud, Matthew Luzum, and Jean-Yves Ollitrault. **Triangular flow in hydrodynamics and transport theory**. *Phys. Rev. C*, **82**:034913, Sep 2010. Available from: `http://link.aps.org/doi/10.1103/PhysRevC.82.034913`. 33

[62] A.M.Poskanzer and S.A.Voloshin. **Methods for analyzing anisotropic flow in relativistic nuclear collisions**. *Phys. Rev C*, **58**(3), 1998. 33, 34, 35, 37

[63] N.Borghini, P.M. Dinh, and J.Y.Ollitrault. *Phys. Rev C*, **63**(054906), 2001. 37

[64] CERN web. **ALICE website**, 2012. Available from: `http://aliceinfo.cern.ch/Public/Welcome.html`. 39

[65] CERN web. **CMS website**, 2012. Available from: `http://cms.web.cern.ch/cms/index.html`. 39

[66] CERN web. **ATLAS website**, 2012. Available from: `http://atlas.ch`. 39

[67] CERN web. **TOTEM website**, 2012. Available from: http://totem.web.cern.ch/Totem/. 39

[68] CERN web. **LHCb website**, 2012. Available from: http://lhcb-public.web.cern.ch/lhcb-public/. 39

[69] CERN web. **LHCf website**, 2012. Available from: http://public.web.cern.ch/public/en/LHC/LHCf-en.html. 39

[70] ALICE Collaboration. **ALICE Figures repository**, 2012. Available from: https://aliceinfo.cern.ch/Figure/general_fig. 40

[71] ALICE Collaboration. **The ALICE experiment at the CERN LHC**. *JINST*, **3**(S08002), 2008. Available from: http://iopscience.iop.org/1748-0221/3/08/S08002/. 40, 41, 42, 43, 45, 46, 47, 48, 49, 50

[72] W.R.Leo. **Techniques for Nuclear and Particle Physics Experiments: A How-to Approach**. *Springer*, 1987. 44

[73] ALICE Collaboration. **The ALICE experiment at the CERN LHC**. *Journal of Instrumentation*, **3**(08):S08002, 2008. Available from: http://stacks.iop.org/1748-0221/3/i=08/a=S08002. 45

[74] ALICE Collaboration. **PMD technical design report**. *CERN EDMS*, **ALICE TRD**(6), 1999. Available from: https://edms.cern.ch/document/398931/1. 47, 48

[75] ALICE Collaboration. **Forward Detectors**. *CERN EDMS*, **ALICE TRD**(498253 v1), 2004. Available from: https://edms.cern.ch/document/498253/1. 48, 49

[76] A.Jipa, C.M. Mitu, M.Niculescu, and S.I. Zgura. **DATA ANALYSIS AND 3D EVOLUTION IN HIGH ENERGY PHYSICS USING GRAPHIC PROCESSOR**. *Romanian Reports in Physics*, **63**(4), 2011. 51

[77] UrQMD. **UrQMD webpage**. Available from: http://urqmd.org/. 51

[78] M.Bleicher, E.Zabrodin, C.Spieles, and et al. **Relativistic Hadron-Hadron Collisions in the Ultra-Relativistic Quantum Molecular Dynamics Model**, 1999. arXiv:9909407v1. 51

[79] UrQMD. **UrQMD User Manual**, 2009. Available from: http://urqmd.org/documentation/urqmd-3.3p1.pdf. 52, 58

[80] SQLite. **SQLite webpage**, 2012. Available from: http://www.sqlite.org/. 52

[81] M.Niculescu. **f14tosqlite webpage**, 2009. Available from: http://sourceforge.net/projects/f14tosqlite/. 53

[82] M.Niculescu. **f14toROOT webpage**, 2011. Available from: http://sourceforge.net/projects/f14toroot/. 54

[83] Qt. **Qt - crossplatform application and UI toolkit**, 2012. Available from: http://qt.nokia.com/. 54

[84] OpenGL ARB. **OpenGL The industry foundation for high performance graphics**, 2012. Available from: http://www.opengl.org/. 54

[85] M.Niculescu. **Video of UrQMD Event Viewer**. Available from: http://www.youtube.com/watch?v=KW52eHWD_so. 54

[86] D. d'Enerria. **Hard scattering cross sections at LHC in the Glauber approach: from pp to pA and AA collisions**. arXiv:0302016v3. 59

[87] Intel. **Intel Advisor**, 2012. Available from: http://software.intel.com/en-us/articles/intel-parallel-advisor/. 68