

Computing at SuperB

Domenico Del Prete^{*,1}, Fabrizio Bianchi⁵, Vania Boccia¹, Vincenzo Ciaschini¹⁰, Marco Corvo¹³, Guglielmo De Nardo^{1,4}, Andrea Di Simone⁸, Giacinto Donvito², Armando Fella⁷, Paolo Franchini¹⁰, Francesco Giacomini¹⁰, Alberto Gianoli¹¹, Giuliano Laccetti^{1,4}, Stefano Longo¹³, Steffen Luitz⁹, Eleonora Luppi^{11,12}, Matteo Manzali¹², Leonardo Merola^{1,4}, Silvio Pardi¹, Alejandro Perez⁷, Matteo Rama³, Guido Russo^{1,4}, Bruno Santeramo², Roberto Stroili^{6,13}, Luca Tommasetti¹²

¹INFN Napoli, ²INFN Bari, ³INFN LNF Frascati, ⁴University of Naples Federico II, ⁵INFN Torino, ⁶University of Padova, ⁷INFN Pisa, ⁸University of Roma Tor Vergata, ⁹SLAC National Accelerator Laboratory, ¹⁰INFN CNAF Bologna, ¹¹University of Ferrara, ¹²INFN Ferrara, ¹³INFN Padova
E-mail: domenico.delprete@na.infn.it, fabrizio.bianchi@to.infn.it, vania.boccia@unina.it, vincenzo.ciaschini@cnafe.infn.it, marco.corvo@pd.infn.it, guglielmo.denardo@na.infn.it, andrea.di.simone@cern.ch, giacinto.donvito@ba.infn.it, armando.fella@pi.infn.it, paolo.franchini@cnafe.infn.it, francesco.giacomini@cnafe.infn.it, alberto.gianoli@fe.infn.it, giuliano.laccetti@unina.it, stefano.longo@pd.infn.it, luitz@slac.stanford.edu, eleonora.luppi@fe.infn.it, matteo.manzali@fe.infn.it, leonardo.merola@na.infn.it, silvio.pardi@na.infn.it, luis.alejandroperez@pi.infn.it, rama@slac.stanford.edu, guido.russo@unina.it, bruno.santeramo@ba.infn.it, roberto.stroili@pd.infn.it, luca.tommasetti@fe.infn.it

The development of a computing model for the next generation of Super Flavour Factories, like SuperB and SuperKEKB, presents significant challenges. With a nominal luminosity above $10^{36} \text{ cm}^{-2} \text{ s}^{-1}$, we estimate that, after few years of operation, the size of the data sample will be of the order of 500 PB and the amount of CPU required to process it will be close to 5000 KHep-Spec06 (the new HEP-wide benchmark for measuring CPU performance). The new many and multi core technologies need to be effectively exploited in order to manage very large data set and this has a potential large impact on the computing model for SuperB. In addition, the computing resources available to SuperB, as is already the case for LHC experiments, will be distributed and accessed through a Grid or a cloud infrastructure and a suite of efficient and reliable tools needs to be provided to the users. A dedicated research program to explore these issues is in progress and it is presented here.

*36th International Conference on High Energy Physics
July 4-11, 2012
Melbourne, Australia*

* Speaker

1. Introduction

The design and implementation of the SuperB machine is being carried out also on the basis of the experience and designs of BaBar [1] and Belle [2] detectors, both in terms of knowledge and in terms of reuse of some components still valid, as much as possible. A SuperB detector is possible with today's technology, reusing large (expensive) parts of Babar; some areas require moderate R&D and engineering developments to improve performances, such as: small beam pipe technology, photon detection for DIRC quartz bars, electronics and triggers, and a new computing model, able to handle a massive amount of data.

The design is based on a new collision scheme, that we call a “crabbed waist” [3]. Pantaleo Raimondi is the author of this new scheme to attain high luminosity in a storage ring, change the collision so that only a small fraction of one bunch collides with the other bunch (large crossing angle, long bunch length); this due to the large crossing angle the effective bunch length (the colliding part) is now very short so we can lower βy^* by a factor of 50. The beams must have very low emittance – like present day light sources, the x size at the IP now sets the effective bunch length. In addition, by crabbing the magnetic waist of the colliding beams we greatly reduce the tune plane resonances enabling greater tune shifts and better tune plane flexibility. This increases the luminosity performance by another factor of 2-3.

Therefore this new scheme will allow SuperB to reach a luminosity by overcoming some of the issues that have plagued earlier super electron ($e^+ e^-$) collider designs, such as very high beam currents and very short bunches. In fact in the conventional Super B Factory designs, the cost is dominated by the requirements for dealing with higher currents and shorter bunches: (for example, substantial additions to the RF system, engineering design for larger HOM power due to shorter bunches, and the cooling and vacuum challenges posed by larger synchrotron radiation power). Most of these problems do not exist in the SuperB design using the new collision scheme “crabbed waist”; In addition, the SuperB design allows the reuse of a great deal of PEP-II hardware, resulting in substantial savings for the project, even at a new site. Nevertheless SuperB can also be a good “light source”, in fact work is in progress to design synchrotron radiation beamlines through the collaboration with Italian Institute of Technology [4]. The area of Tor Vergata campus is the site chosen for the construction of the new accelerator. The size of this bed is about 100.000 square feet as shown in Figure 1.



Figure 1: On the left - the site of the Accelerator (Tor Vergata Campus area); On the right - distance between the site of the accelerator and the National Laboratories of physics at Frascati

2. The Computing Model

The main activities of the SuperB Computing Group are mainly focused on same key points, such as: the problems of data access in a distributed and efficient manner, and a software solutions that will provide faster and easy access on this data for the analysis, assuming also that datasets will be geographically distributed. In addition, the optimization of data processing both through the new technologies of parallel computing and through the optimization of legacy source code.

Moreover, in the preparation studies for the SuperB computing TDR, the collaboration is considering the possibility of creating an offlineTier0 (tapes) and a single distributed Tier1, based on sites interconnected by a network at $n \times 10$ Gbps, updatable to $n \times 40$ Gbps and later to $n \times 100$ Gbps, in order to minimize the latencies on all types of applications of the computing team: dynamic distributed data access, distributed parallel computing, monitoring and much more. The idea is an extrapolation of BaBar computing model to a luminosity 100 times larger. “Raw data” from the detector will be permanently stored, and reconstructed in a two-step process: i) a “prompt calibration” pass on a subset of events to determine calibration parameters (constants); ii) a full “event reconstruction” pass on all the events that use the parameters derived in the previous step. Then, iii) Monte Carlo data will be processed in the same way, and selected subset of detector and Monte Carlo data, the “skims”, will be made available for different areas of physics analysis. This approach is very convenient for the analysis, but it increases the storage requirement because the same events can be present in more than one skim.

Moreover, improvements in constants, reconstruction code, or simulation may require reprocessing of the data or generation of new simulated data, and this requires the capability of reprocessing in a given year all the data collected in previous years. Making an attempt to estimate the SuperB detector computing needs, we can say that (with a limited precision due to the many assumptions): Raw Event size ~ 100 kByte (= 3 x BaBar), Mini/Micro event size = 2x BaBar, CPU/unit lumi: 3x BaBar, Skim expansion factor: 5, Some fraction of Mini on disk (100% \rightarrow 10%), Equivalent amount of Monte Carlo “lumi”, 2 copies of raw data one of these stored on tape, then: Storage grows from O(50) PB to O(600) PB in 6 years, and CPU grows from 500 to 12,000 KHepSpec in 6 years.

2.1 The Infrastructure and the Network

The sites involved in the future SuperB computing infrastructure are shown in the picture below, a part from the LNF-Tor Vergata site, close to the future collider, which should host the tier0 site (but no decision has been made so far). The sites in the circle were funded by European Union under the PON 2007-2013, with a project named ReCaS (PONa3_00052) [5]. The work for realizing the infrastructure of the ReCaS project have already started, at the date of November 2012. These sites are already part of the national grid infrastructure, they will handle the increased amount of data produced by the detectors of the experiment. The network of tier1 will be distributed as shown in Figure 2.

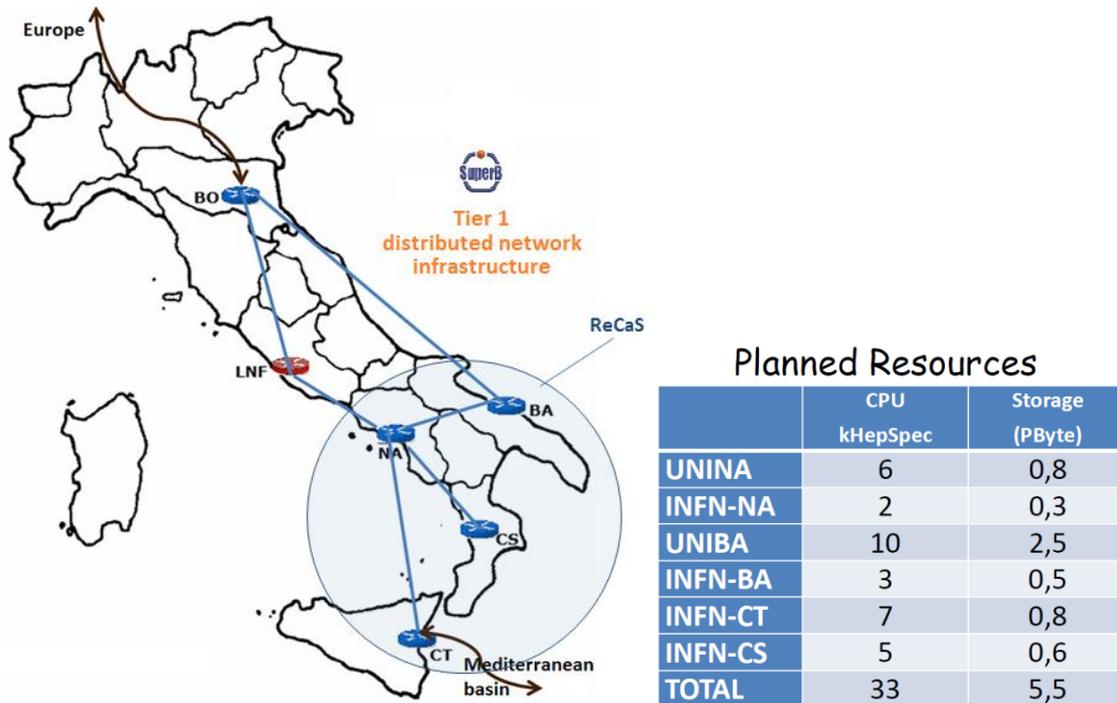


Figure 2: The distributed Tier1 network infrastructure planned for SuperB

In detail, the specification for storage, server and infrastructure for UNINA, INFN-NA, are completed. UNIBA, INFN-BA and INFN-CT require more detailed specifications for the power and cooling subsystems. The first tenders (NA & BA) will start early next year.

The computing model requires an efficient network among sites; to fulfill this basic requirement, all sites are migrating to the new GARR-X network [6]; the Consortium GARR, the Italian NREN, is driving the installation of dark fibers between major scientific and academic institutions in Italy, including SuperB sites. Since the end of 2012, a dedicated 10 Gbps network between the SuperB Data Centers sites will be operational. When more speed will be necessary, the lines will be moved to 40 Gbps with a change in the optics, and multiplication via DWDM [7] will ensure faster data rates.

2.2 The Currently Available Resources

The next table in *figure 3* summarizes all distributed resources already available in SuperB Virtual Organization over 27 sites: Canada, France, Italy, Poland, UK and USA. The current total distributed resources, in terms of dynamic allocation of cores, up to 11500, and for nearly half petabyte of distributed storage. The details of the available resources is shown in Figure 3.

The availability of these resources plays a very important role in the preliminary phase of the project. In fact, thanks to the GRID Virtual Organizations can be carried out simulations on the future production code, and hence the refining of a new generation of framework. In this phase it is interesting assess the impact of future use of grid infrastructures geographically distributed, in order to solve computational problems in a HEP experiment that produce a massive amount of data in the next few years.

Site	Min (cores)	Max (cores)	Disk (TB)	SRM layer	Grid Org.	Site contacts
RAL(T1)	200	1000	25	Castor	EGI	F. Wilson, C. Brew
Ralpp	50	500	5	dCache	EGI	F. Wilson, C. Brew
Queen Mary	300	2000	150	StoRM	EGI	A. Martin, C. Walker
Oxford Univ.	50	200	1	DPM	EGI	K. Mohammad, E. MacMahon
IN2P3-CC(T1)	500	1000	16	dCache	EGI	N. Arnaud, O. Dadoun
Grif	50	300	2	DPM	EGI	N. Arnaud, O. Dadoun
in2p3-lpsc	50	100	2	DPM	EGI	J.S. Real
in2p3-ires	50	100	2	DPM	EGI	Y. Patois
CNAF(T1)	500	1000	180	StoRM	EGI	A. Fella, P. Franchini
Pisa	50	500	0.5	StoRM	EGI	A. Ciampa, E. Mazzoni, D. Fabiani
Legnaro	50	100	1	StoRM	EGI	G. Maron, A. Crescente, S. Fantinel
Napoli	500	2000	15	DPM	EGI	S. Pardi, A. Doria
Bari	160	260	0.5	StoRM/Lustre	EGI	G. Donvito, V. Spinoso
Ferrara	10	50	0.5	StoRM	EGI	L. Tomassetti, A. Donati
Cagliari	10	50	1	StoRM	EGI	D. Mura
Perugia	10	50	1	StoRM	EGI	R. Cefala'
Torino	50	100	2	DPM	EGI	S. Bagnasco, R. Brunetti
Frascati	30	100	2	DPM	EGI	E. Vilucchi, G. Fortugno, A. Martini
Milano	50	100	2	StoRM	EGI	N. Neri, L. Vaccarossa, D. Rebatto
Catania*	?	?	?	StoRM	EGI	G. Platania
Slac	400	400	10	NFS	OSG	S. Luiz, W. Yang
Caltech	200	400	4.5	NFS	OSG	S. Lo, F. Porter, P. Ongmongkolkul
Fnal*	50	400	1	dCache	OSG	M. Slyz
OhioSC*	?	?	?	dCache	OSG	R. Andreassen, D. Johnson
Victoria	50	100	5	dCache	EGI	A. Agarwal
McGill*	100	200	1	StoRM	EGI	S. Robertson, S.K. Nderitu
Cyfronet	100	500	10	DPM	EGI	L. Flis, T. Szepienie, J. Chwastowski
Total	3570	11510	440			

* VO enabling procedure in progress

Figure 3: Distributed resources already available in SuperB Virtual Organization

3. Research and Development

3.1 Many-Core computing technologies

One of the R&D activity of great interest for the SuperB project is the exploit of parallel computing through the use of both the GPGPU (General Purpose Graphic Processing Unit) technology and the Many-Core technology, which both provide a high density of cores for faster atomic processing. The main goal of the R&D activity is to understand the impact, the benefits and the limits of these technologies: The first step is the use of the GPGPU architecture through test algorithms that simulate problems of high energy physics, thus delegating part of the code that can be parallelized to the GPGPU as a math co-processor, and to measure the achieved performances, with respect to the CPU-only version. The goal is to investigate if the computational problems coming from the SuperB experiment can be parallelized with this architecture. Although the use of the GPGPU only as a math co-processor limits the use of the GPGPU, we are confident that one order of magnitude can be achieved in timing of the application, a fact which should guarantee us on the usefulness of integrating CPU and GPGPU in the reconstruction algorithms for SuperB. As a by-side product, we will be disentangle the natively parallel algorithms form the rest of the code, thus being able to separately investigate, as a second step, the use of Multi-Core technologies, that is CPUs with dozens of core each. As shown in Figure 4, the main difference between CPU and GPGPU lies in the different way that

they have taken in the composition of hardware. To date there are two main techniques for the development of microprocessors: **multi-core** and **many-core**.



Figure 4: Structural difference CPU – GPU

A multi-core CPU is optimized with respect to the performance during the execution of sequential code. It makes use of a sophisticated control logic to allow instructions from a single thread of execution to be performed in parallel or even in a different sequential order (out-of-order), while maintaining the logic of a sequential execution. The technology makes use of large, multi-level caches to reduce latency when accessing data and instructions. An important aspect for the performance evaluation is the bandwidth for the memory access. The graphics chips have the bandwidth 10 times greater than that available on-chip CPU. The difficulty in increasing the bandwidth for the CPUs is due to the limitations posed by operating systems, applications and system I/O. On the contrary, designers of GPGPU can achieve a higher bandwidth due to reduced limitations inherited (legacy constraints) and simplified memory models. The design philosophy of the GPGPU is modeled after the videogame industry pressing to get high speed for floating point operations.

The hardware is designed so that a large number of threads of execution continues to work even when some threads are waiting for data from the memory, minimizing the control logic for each thread. The use of small cache memories makes it possible to prevent multiple threads simultaneously accessing data from DRAM. Most of the chip area is dedicated to floating-point operations. GPGPUs are designed as engines for the numerical calculation. It is also true that the CPU and GPGPU are designed to perform different tasks and do not give good performance when used for tasks other than those they were planned for. Many applications need both good CPU and GPGPU technologies, so as to perform parts of the CPU intensive numerical computation on GPUs. The model currently used in the SuperB experimental code is the nVIDIA CUDA programming model (Compute Unified Device Architecture) designed to support the running of an application as a whole, on both CPU and GPGPU, and this CUDA environment is what we are using to optimize the model of data processing for the SuperB project requirements.

The SuperB applications that can take full advantage from this new technology have to fit well with the SIMT parallelism model (Single Instruction Multiple Thread); each thread represents a fundamental unit of computations, e.g. in the reconstruction algorithms, and has to be executed on a single GPGPU core, also called SP (Streaming Processor) on nVIDIA boards.

This template creates and manages the threads in groups of 32, called warps, and these groups of threads are launched along the same line of code, even if they can “handle” parts of the different algorithm. The introduction of the concept of the warp is a implementation choice for the SIMT model and cannot be changed by the programmer; this technique, however, guarantees maximum efficiency with the least possible use of thread.

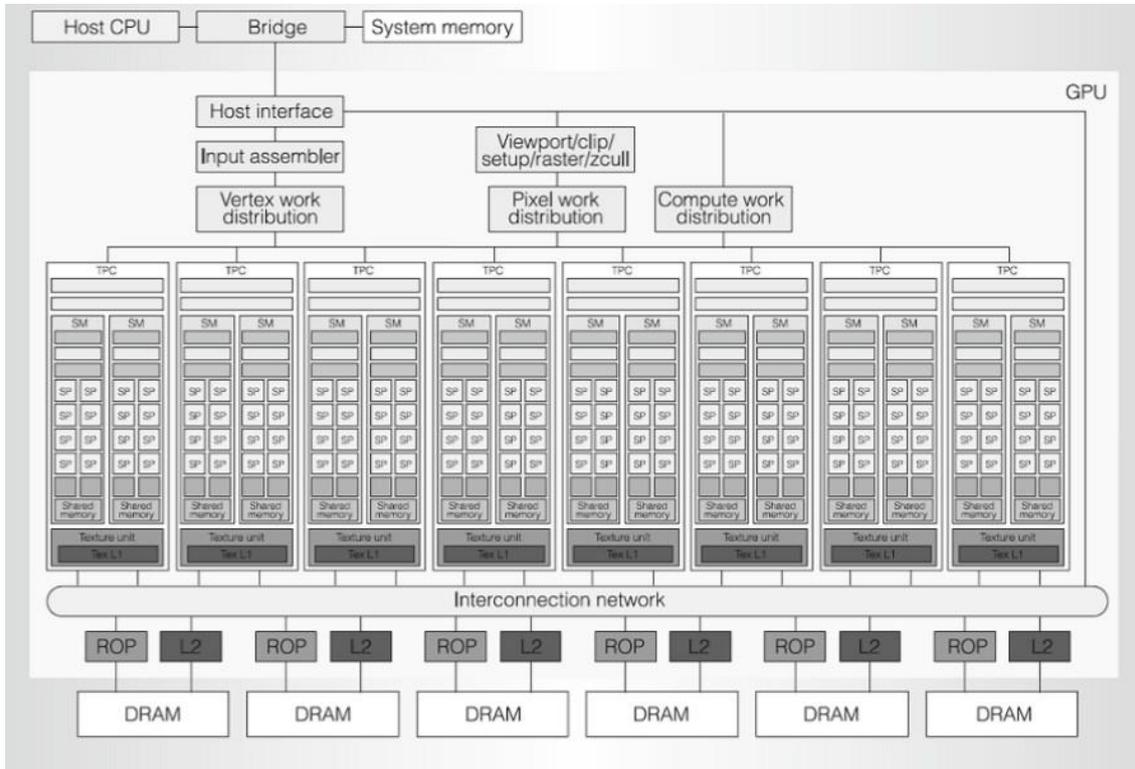


Figure 5: GPU nVIDIA Tesla architecture

The architecture used is composed by arrays of multiprocessors SM (Streaming Multiprocessor) each of which is constituted by a series of streaming processor that form the logic control and share the instructions in the cache (Figure 5). A GPGPU has a dedicated memory, the GDDR (Graphics Double Data Rate), also referred to with the name of *global memory*. Flanked in this type of memory there are other two major kinds of memory, the *shared memory* and the *local memory*. The *shared memory* is a memory shared by the threads during the execution of an application on the device. This memory has a capacity of about 64 KB (on average) between the various tabs of nVIDIA, and serves to reduce the transfer of data from the global memory to the various SP. The *local memory* has a private area of memory for each running thread and is used to manage logs and the execution stack.

In our experimental environment we used as hardware the nVIDIA Tesla series [8] and C-CUDA [9] as a programming language, which allows us to use C/C++ without need to use graphical programming interfaces. An important aspect that has made the GPU most interesting to us, is the adoption of the IEEE standard for floating point representation. This implies that the results obtained with a GPGPU are easily comparable with those obtained with a standard CPU.

The main drawback we had to cope with, was the use of single precision data, since applications that need double precision could not run on the GPGPU. To assess whether the model Many-Core GPGPU usage was better than the CPU architecture, we wrote an algorithm using benchmarks that could solve a typical problem of particle physics. We started with an algorithm (toy model) for the reconstruction of the decay of neutral mesons π^0 from the combination of pairs of photons input data as shown in Figure 6. We first wrote a normal sequential code in C, then we evaluated which parts of the code could be parallelized to be implemented in the CUDA C language. The implementation logic was the following: the parts of the code which looked like *poorly repetitive* were handled by the CPU, and when we recognized necessary to carry out the same activities on an enormous amount of data (SIMD parallelism) we routed the task of computation to the GPGPU. The main goals of this naïve approach are: *i)* understand the impact, benefits and limits of using the GPGPU architecture and *ii)* determine the impact on the performance.

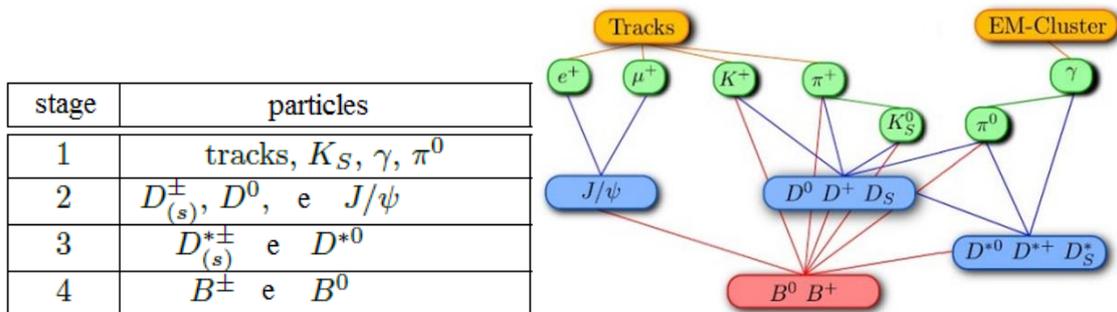


Figure 6: The four stages of the simulation algorithm for the reconstruction of the B meson (toy-model)

We have prepared various types of tests in order to compare three different approaches, which solve in different ways the same type of problem: *a)* a sequential algorithm for CPU, *b)* a parallel algorithm for multi-core CPU, *c)* a parallel algorithm for many-core platform Fermi GPGPU. The results were very interesting in all three types of approach; in particular, as we expected on the basis of the parallel environments used, we confirm that the user must resort to the massively-parallel methods only when necessary, *i.e.* when working on a substantial number of input data. Preliminary tests have directed us to use the coprocessing CPU / GPU, so as to exploit the parallelism within the source code.

It is precisely the “coexistence and integration” of the various approaches the main achievement of this new paradigm, that is the programmer has to determine, for each practical case, which parts of the algorithm has be delegated to the CPU and which parts to the GPGPU, in order to take the most from both subsystems, and thus reaching the maximum gain from the point of view of performance . Therefore we can say say that the CPU-GPU co-processing can be realistically considered the future with regards to the high-performance computing, especially for scientific experiments.

3.2 Distributed storage access

Another challenge for the SuperB computing model concerns the testing of failover capabilities and scalability of storage access over wide area network, using a new distributed file system -such as *Hadoop* [10] and *GlusterFS* [11]- that provides a shared store mechanism and automatic geographic data replication; of course, this implies the building of a software library that will easy access to data for analysis application. We are testing both file systems and, in addition, we are testing remote data access to datasets stored with these file systems, using HTTP protocols. Usually, end-users access data over WAN, with tasks such as datasets transfers from different distributed sites and the dynamic jobs execution. Therefore try to decrease the performance loss on remote data access, so all problems due to concurrent applications that works on distributed dataset. Finally the integration of all user interfaces of analysis application in a centralized software solution for single distributed Tier1, like integrated web portal.

Today, in the production Grid infrastructure there is a strong separation between storage and computing facilities following the MONARC [12] hierarchical computing model. The data are stored in disk array and made available through a Storage Element protocol, by means of services like SRM (Storage Resource Manager) and gsiftp/gridftp transfer protocols (on left of figure 7).

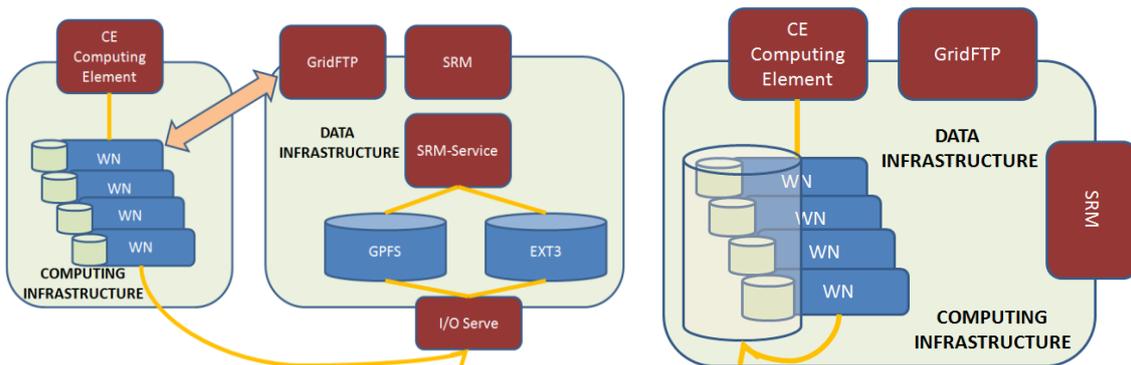


Figure 7: On the left: Typical Scenarios of a Tier X facility within the MONARC computing model currently used – On the right: New scenario, worker node can hold the data to analyze in our local disk

The realization of the typical scenario requires a complex infrastructure in terms of switches, storage processors and servers, in order to export the storage. During operations, a typical analysis job downloads the required datasets from the appropriate storage element to a local disk, then starts the computation. This approach introduces a systematic overhead in analysis jobs over the storage system capacity in terms of network and read/write disk operations, and suffers from scalability problems when the number of computational nodes grows. In Figure 7 is shown the new solution at the right side. The solution we propose in SuperB computing is the integration of computational and storage resource by investigating the performance of a system with Worker Node with embedded disks, overcoming the data-centric model. In particular our model provides a farm of nodes that store the data directly in the local disk shared among all the cluster component through distributed cluster file system. From the

announced roadmap of the main hardware constructors, we expect that future technologies allow to grow the core and disk density for single servers. Currently (year 2012) is a single server with 48 core and 18 TB of disk space in 2 rack unit. The possibility to port (at least part of) the standard HEP code on GPUs allows us to further reduce the space needed. We expect to have 128 core per node and high density disks (6 TB each) in commodity hardware. This approach allows us to move the computing power to the data, following a file affinity scheduling approach at cluster level. In other words, we do not want anymore to have separated computing (WN) and storage (SE) nodes, but to have stand-alone servers that accommodate all the data they can work on, performing the computations with their own cores and with dedicated attached GPUs as math co-processors for standard math library calls. The main issue in supporting this model is the design of the right cluster configuration, that is the correct balance between network interconnection, number of nodes and disk space, as well as number of GPUs.

Experimental measurements show that a typical SuperB Job is able to process a data rate between 20MB/s with a peak at 50MB/s, moreover the CPU occupation for a single job is about 90%. In other words, a server with 8 cores can run 8 jobs concurrently, each using an average of 90% of the core potentiality, and each using an amount of RAM memory of the order of 2-3 Gbytes. Following the future trend of CPUs, we expect an increase in the ratio cores/socket instead of an increase in the core clock, so we expect that the computation will pose the upper limits in performance, and the GPU will show in the next future. However, the present model cannot scale because of the need of dedicated memory, which cannot scale in parallel with the number of cores: if nowadays a server with 8 cores uses 16-24 Gbytes of memory for 8 concurrent jobs, each accessing data over the network, in the future a server with 128 cores will not have 256-384 Gbytes of memory for 128 jobs, as the cost of memory would be too high, and will not have access to a proportional quantity of disk spaces over the network, as the network bandwidth is not scaling up with the number of cores. Therefore, in designing our cluster we have taken into account this value in order to avoid to create additional bottleneck provided by network and disk access. The hardware parameters that need to scale are cpu, disk and network. Considering N boxes with M cores, we need a data access rate of $M*20\text{MB/s}$. This value corresponds to the network file system bandwidth occupation between the nodes. This solution, in the absence of file system overhead and with the assumption of a statistically homogeneous access to each disk, can scale linearly to respect the number of server and the storage occupation. Otherwise, the standard grid setup is limited by the performance of the storage system and by the storage processor limits.

3.3 Distributed monitoring solutions

A distributed computing system that will support the SuperB project will need a valid software tool for the management and monitoring. It is estimated that the new accelerator will have to bear an average production of around 30TB a day, these data should be analyzed and made available in a fully site-independent way. For this reason the use of advanced services at the network layer is an important prerogative for all relevant data are available, in the shortest time possible and with maximum reliability, all Tier involved in the analysis and processing.

The INFN locations in south Italy will participate in the computing model of SuperB, making available a facility for calculating class Tier-1 facilities, multi-regional, currently in the final stage of design. The need to monitor the status of these services configured in a distributed system requires an advanced monitoring system [13], capable of controlling all services of the stack, from machines to applications. This allowed the user to abstract the underlying hardware in order to use all services in a transparent manner, underestimating all the dynamic aspects of the case.

In next picture shown below (Figure 8) we show the manager site architecture (cloud based) that monitors itself and also captures all information from each distributed site. The Central site (orange modules) will handle the entire monitoring through an *enterprise information portal* (EIP), Liferay based [14], which in turn will allow the end user to see in a practical, simple and centralized way all information of different sites. The portal is the most dynamic module, because in it one can also integrate and accommodate new systems not covered now, and which could also reside in different sites. This centralized monitoring system will be able to monitor all services, starting from basic layer to application layer. A Layer2-like network will ensure the visibility of monitoring remote nodes between sites over a cloud distributed infrastructure.

Using Liferay as a portlet container, we could integrate several heterogeneous tools, allowing an integrated vision of all sites.

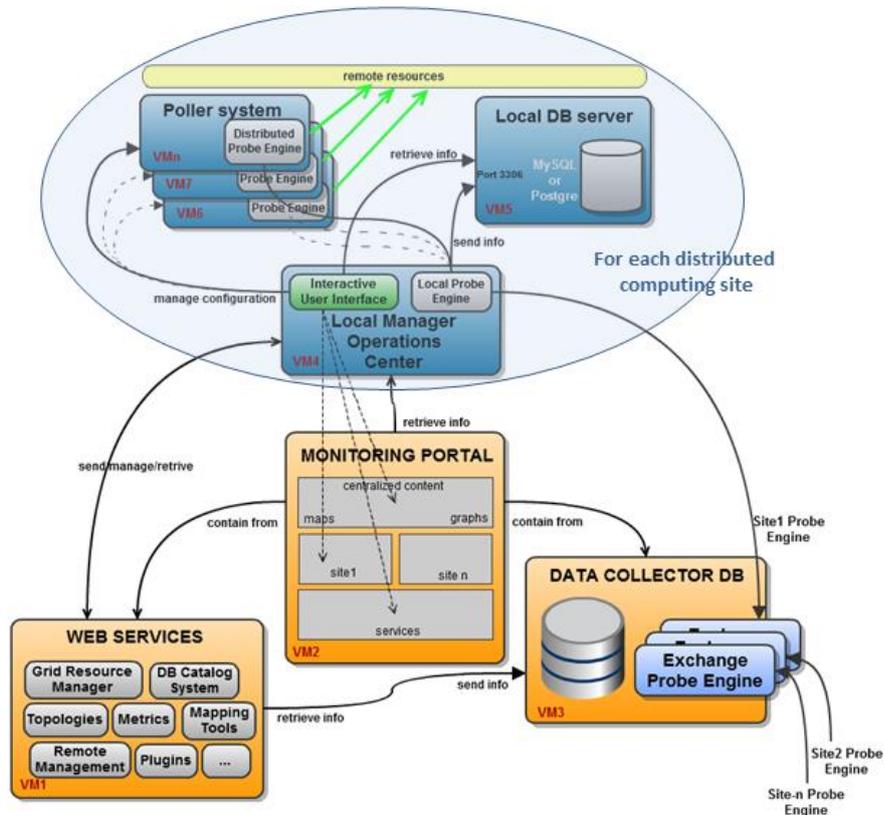


Figure 8: Distributed Monitoring Architecture

The interconnection between the converged sites of the project implies that each datacenter inherits a portion of storage to offer, in a shared way, within the distributed infrastructure. So each site will have several Pbyte of storage, and all together the sites will make up the entire distributed file system (Figure 9). On this Distributed Infrastructure we are installing and configuring a cloud operating system (OpenStack) [15] that will provide the management of all cloud resources, and that will be controlled by the monitoring system. In this way, the environment will be unique and shared between sites, distributed, but managed as if it were a local environment. This type of solution is currently in the final phase of testing; it appears as the most flexible from the point of view of usability of resources over distributed wide area network. So, every Data Center manager will have a set of virtual machines to monitor his own site, but they will pass information to the central site, through a Layer2-like network provided by the Consortium GARR.

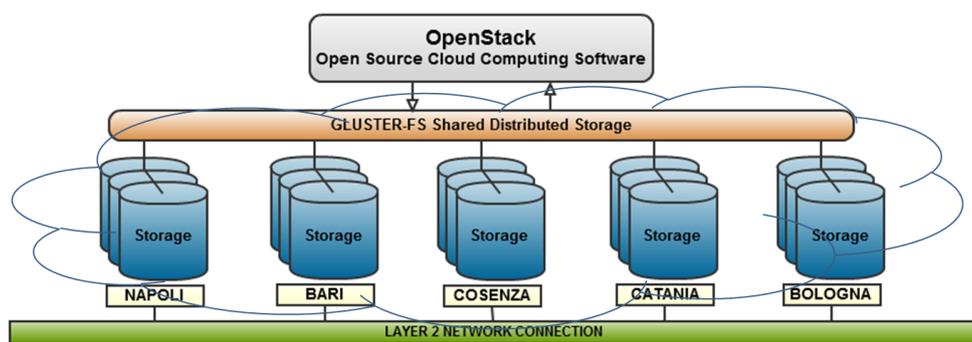


Figure 9: Shared Storage over a Cloud Operating System

The features provided by this type of cloud software may be of fundamental importance for the management, control and data reliability, thanks to the integrated vision (Figure 10).

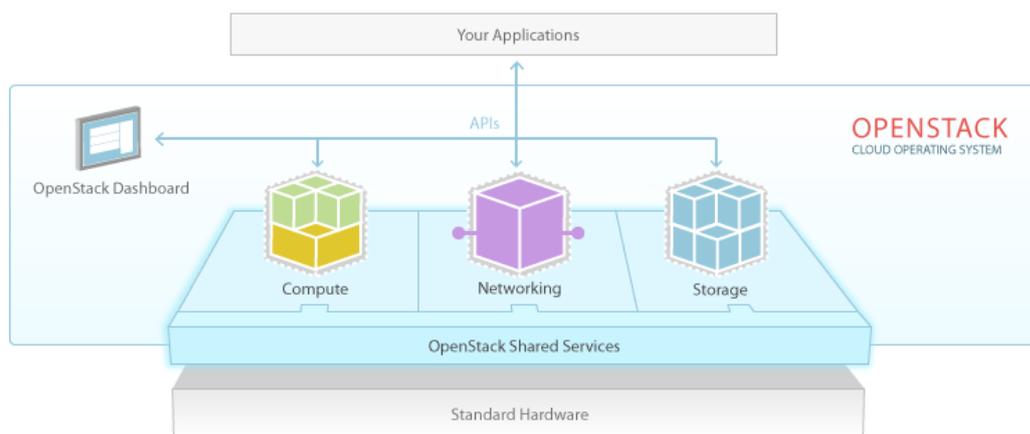


Figure 10: The modules of the cloud operating platform

Compute module allows users to create virtual computers on which to run their own computer applications and to create, launch, and terminate server instances as needed (monitoring servers, web services, user interface, web portal and more). A very important

feature provides users with control over the geographical location of instances that allows for latency optimization and high levels of redundancy.

Storage module provides storage through web services interfaces, Object and Block storage for use with servers and applications (virtual disks, shared storage, databases, synchronization services, filesystems and more).

Networking module provide “network connectivity as a service” between interface devices; all network features are seen as a services (pluggable, scalable, API-driven network and IP management and all network-features in a cloud view). Moreover the *dashboard module* that provide to manage all component of entire infrastructure.

4. Conclusions

The main goal of SuperB experiment is the search for new particles, which may be divided into two categories: those that can be detected directly in processes at low energies, and those that can be detected only indirectly. Both types of research require the analysis of huge data samples, which certainly far exceed the amount of data gathered from past experiment of the same type, like for example Babar. This new era of particle physics poses strong constraints on computing and storage availability for data analysis and data distribution. All the proposed solutions contribute to develop a new generation of framework and evaluate solutions as a follow up to what already used in the experiments of the same type. The Research and Development activity planning look forward to what will be available in a few years from now, in order to predict and estimate the technical expectations for new resources and the corresponding price, the value of which can also change a lot over a period of 24 months. This represents a new challenge not only in physics but also in the design of new technologies that make it possible to process the information acquired by the detectors as efficiently as possible, so that scientists can analyze the data in a site independent manner. These considerations and work activities in progress can already give us an increasingly clear picture of what are the real needs of the SuperB machine, and compare the results with the technologies currently used that allow to support the computing model of SuperB.

References

- [1] The BaBar Experiment Website, <http://www.slac.stanford.edu/BF/>, 2008.
- [2] The Belle Experiment Website, <http://belle.kek.jp/>, 2010.
- [3] M. Zobov (INFN LNF), P. Raimondi (INFN LNF), D. Shatilov (IYF, Novosibirsk), K. Ohmi (KEK), *Crab Waist Collision Studies for e+e- Factories*, [arXiv:0802.2667](https://arxiv.org/abs/0802.2667), Feb 2008
- [4] The Italian Institute of Technology Website, <http://www.iit.it/>, 2012.
- [5] ReCaS, *Rete di Calcolo per SuperB ed altre applicazioni*, Website, <http://www.pon-recas.it/>, 2012.
- [6] The GARR-X Network, Consortium GARR Website, <http://www.garr.it/rete/garr-x>, 2012.
- [7] F. Palmieri, *GMPLS-based Service Differentiation for scalable QoS support in all-optical GRID applications*, Future Generation Computer Systems 22(6), [pp. 688-698, 2006](#), may 2006.

- [8] GPGPU Fermi architecture nVIDIA Website, <http://www.nvidia.com/object/fermi-architecture.html>, 2011.
- [9] Software Development Tools for CUDA C, http://www.nvidia.com/object/tesla_software.html
- [10] The GlusterFS, Website, <http://www.gluster.org/>
- [11] The HDSF, Website, <http://hadoop.apache.org/hdfs/>
- [12] M. Aderholz et. al., *Models of Networked Analysis at Regional Centres; MONARC Phase 2 Report*, CERN-LCB-2000-001, March 2000.
- [13] D. Del Prete, S. Pardi, G. Russo, *A Centralized, Extensible, Multilayer Monitoring System for Distributed Infrastructures: the Atlas Tier2-Naples experience*, E-ISBN: 978-0-7695-4528-8, Print ISBN: 978-1-4577-1458, DOI: [10.1109/CCP.2011.35](https://doi.org/10.1109/CCP.2011.35), June 2011.
- [14] OpenStack Open Source Cloud Computing Software, Website, <http://www.openstack.org/>
- [15] Liferay Enterprise Information Portal, Website, <http://www.liferay.com/>