Computing Summary

A. Di Simone
INFN Tor Vergata

On behalf of the computing team

Outline

- General remarks
- Physics tools
  - Fast sim
  - Full sim
- Distributed computing
- R&D

General remarks

- It has been an extremely fruitful meeting
- If you followed the computing sessions, you must have noticed that the computing group is successfully coping with a twofold problem
  - Provide support for ***present*** needs of the collaboration
  - Foresee ***future*** needs, and plan developments
- In spite of the very limited manpower, a lot of work is being done
  - Much more than I can present here
- I'll show you only a selection of relevant contributions

Physics tools: bkg frames and EMC simulation

A. Perez

background frames: a way to overlap the machine background over the physics event simulated with FastSim

- **A first look at the Bkg-mixing-frame code seems to show that is working properly, but still more checks need to be done**

Next steps:

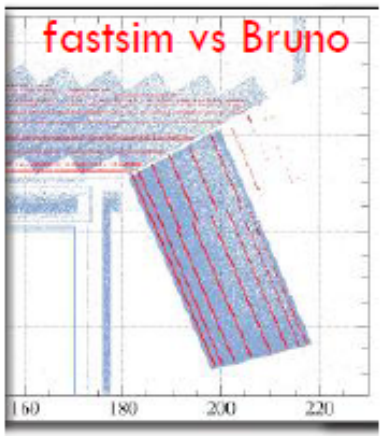
-)include bkg frames for beam gas and Touschek backgrounds

-)develop bkg frame QA code

C. Cheng

Intense development activity since last meeting

- Improved clustering algorithm to deal with the increased bkg rates
- Improved geometry description and alternative geometries
- More realistic shape model
- Studies of radiative Bhabha background.



New background frame production needed (without energy cut on photon energy)

Physics tools: skims and vertexing tools

M. Rama

A skim is a subset of data defined by a given set of selection criteria.

Skims allow to make physics production significantly more effective.

- Concept of *skim* now implemented in FastSim
- Code committed (V0.3.1+Patches_devel) and validated. More validation planned

G. Inguglia

- Vertexing tools subgroup in place. Contact person: Gianluca Inguglia (QMUL)
- Main short term goals:
  - Maintain the existing vertexing tools in FastSim
  - Take care of the documentation
  - Perform vertexing studies at the Psi(3770) resonance
- Preliminary time-dependent studies at charm threshold presented. More results expected for the Elba meeting

Physics tools: status reports from subsystems

Status reports from subystems

SVT

N. Neri

- Detector response reasonably modeled
- Few adjustments necessary for TDR studies
- Quality monitor code for MC productions essentially ready and documented.

DCH

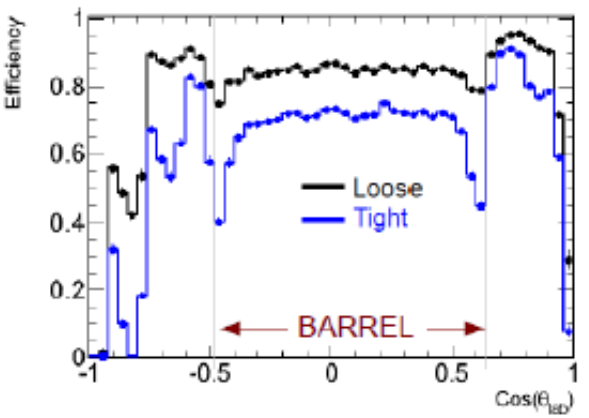
J.-F. Caron

- Proposed changes to DCH geometry in FastSim

IFR

M. Rotondo

- Two muon selectors now available in FastSim
  - Loose
  - Tight
- Quality monitor code for MC productions essentially ready



FullSim

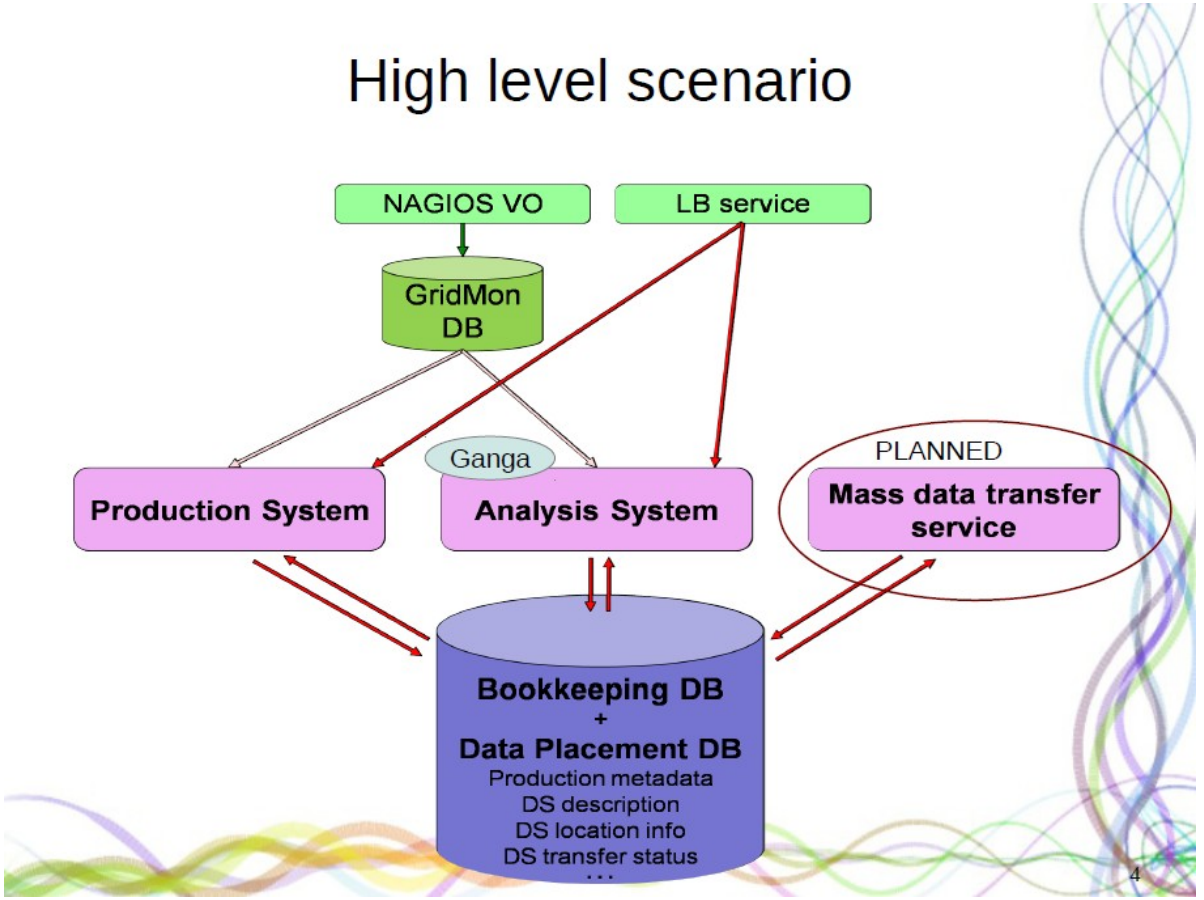
- A few things were implemented in view of the next round of production
  - More precise tuning of the bgframes
  - More generator info in metadata
  - Geantinos in Guinea Pig
  - BFieldDumper
  - Radmon
  - Optical photons for FTOF
- In addition, there are a number of more general, medium-term, longstanding issues we need to start dealing with
  - Event display
  - Event structure
  - Runtime configuration
  - Geometry handling
- There are MANY MORE issues we should be working on, but one must be realistic, and with present manpower they'll have to wait

http://agenda.infn.it/materialDisplay.py?contribId=312&sessionId=43&materialId=slides&confId=4441

Distributed computing

- Use of the tools discussed in the previous slides is ***already*** happening in a distributed environment (GRID)
- To the non-expert this may sound obvious, but a lot of work is needed, under the hood, to make it happen
  - Definition of a Computing Model: What kind of operations will we perform with our hardware?
  - Definition of a bookkeeping DB: what configuration was used to produce this particular file? What files were produced with this configuration?
  - Definition of a data model: where are the input files I need? How do I get them? Where will my results be stored?
  - Development of a software to protect the user from the underlying complexity of the above mentioned points (and from much more...)

Bookkeeping DB



- Recently changed the underlying DB (MySQL → PostgreSQL)
- System has been stress tested, to ensure proper operation in “real-life” use scenarios
  - Up to about 900 DB operations per second were easily performed

http://agenda.infn.it/getFile.py/access?contribId=206&sessionId=6&resId=0&materialId=slides&confId=4441

Data Model

Distributed Storage

- Activities progess on:
  - Computing Model survey
    - Please provide input on: http://mailman.fe.infn.it/superbwiki/index.php/Distributed_Computing/Distributed_storage_portal
  - Technology tracking
    - HADOOP, TEG groups
      - New technologies/solutions could improve the work of SuperB community and could interesting in order to write the Computing TDR
  - HTTP Remote data access
- Many others activities are pending for lack of Man Power
  - We really need new people joining the group with significant effort dedicated

Computing Model - Survey

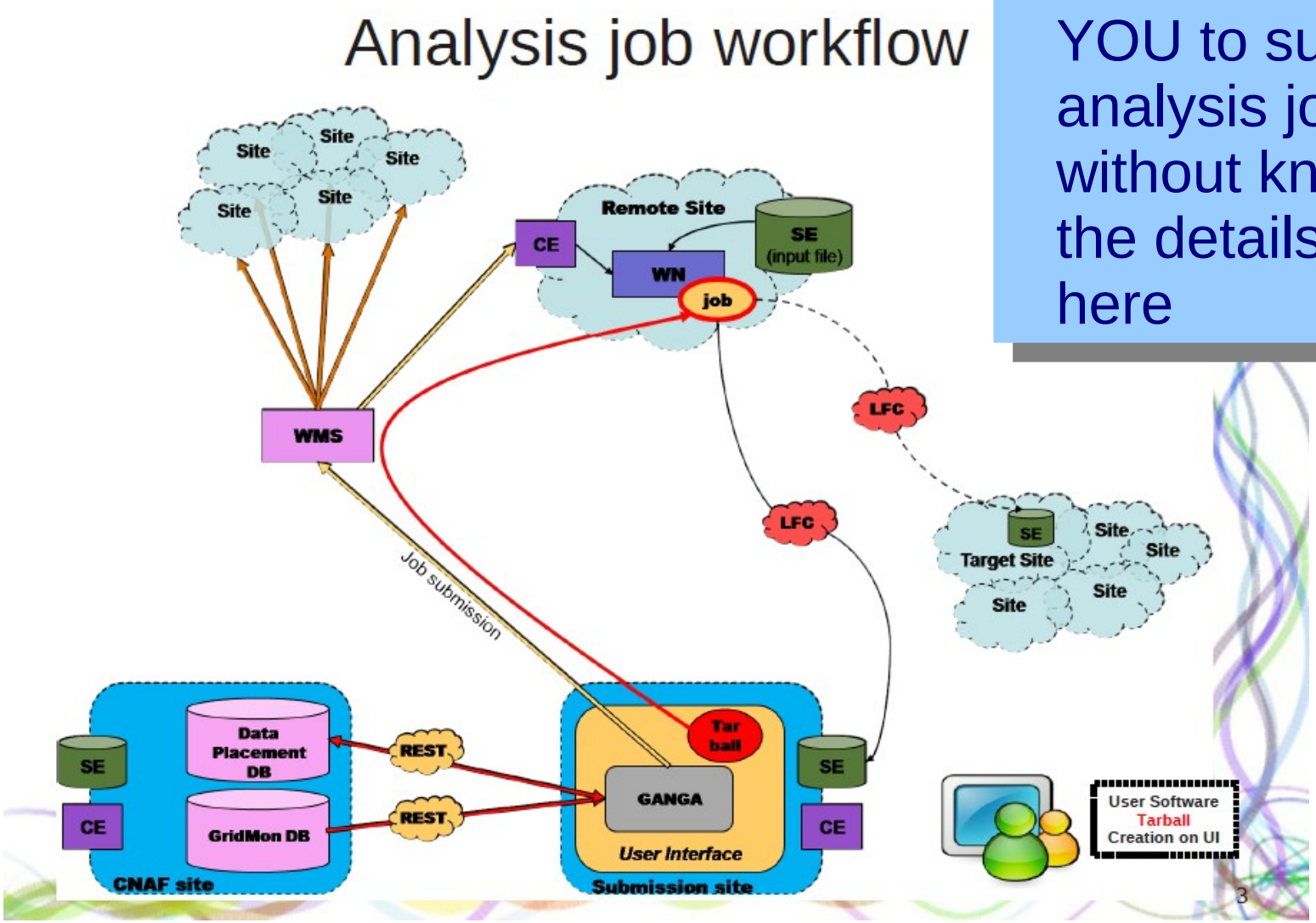
Computing Model survey

- Organized Processing
  - Frequency of reprocessing, IO definition
  - Organized physics groups productions
- Calibrations
  - Are there dedicated calibration/alignment samples?
  - Frequency? Latency?
- Analysis
  - Accessing any possible data format?
  - Is "Sparse" data access possible?
- Quantitative information
  - What is the amount of MC to be produced?
  - Do the following (by Steffen) need to be reviewed? http://agenda.infn.it/getFile.py/access?resId=0&materialId=0&confId=4678

- A Survey has been prepared by the distributed computing group
- WE NEED INPUT FROM ALL OF YOU
- Note that these are NOT "geeky" details
  - They will really affect the everyday life of all collaborators, and the effectiveness of the collaboration as a whole

Distributed Tools - Ganga

- Goal is to allow YOU to submit an analysis job without knowing all the details shown here



Distributed Tools - Ganga

Distributed Analysis System

- Analysis of next production output data should be performed in a distributed environment
- SuperB Ganga Plugin is under heavy development, an alpha release is ready
- Implemented use cases and features:
  - Analysis: official and personal FastSim and FullSim dataset analysis
  - Personal simulation production
- User feedback is essential to ensure Ganga has all required functionalities for distributed analysis.
  - We are proposing to form a “Focused group” of users interested in collaborating in Ganga SuperB framework testing
- Reference SuperB ganga list:

  superb-ganga@lists.infn.it

- Some documentation ready, a tutorial can be organized upon request

R&D

- Remember the two R&D workshops we had in Ferrara?
  - I'll show you here some follow-up studies

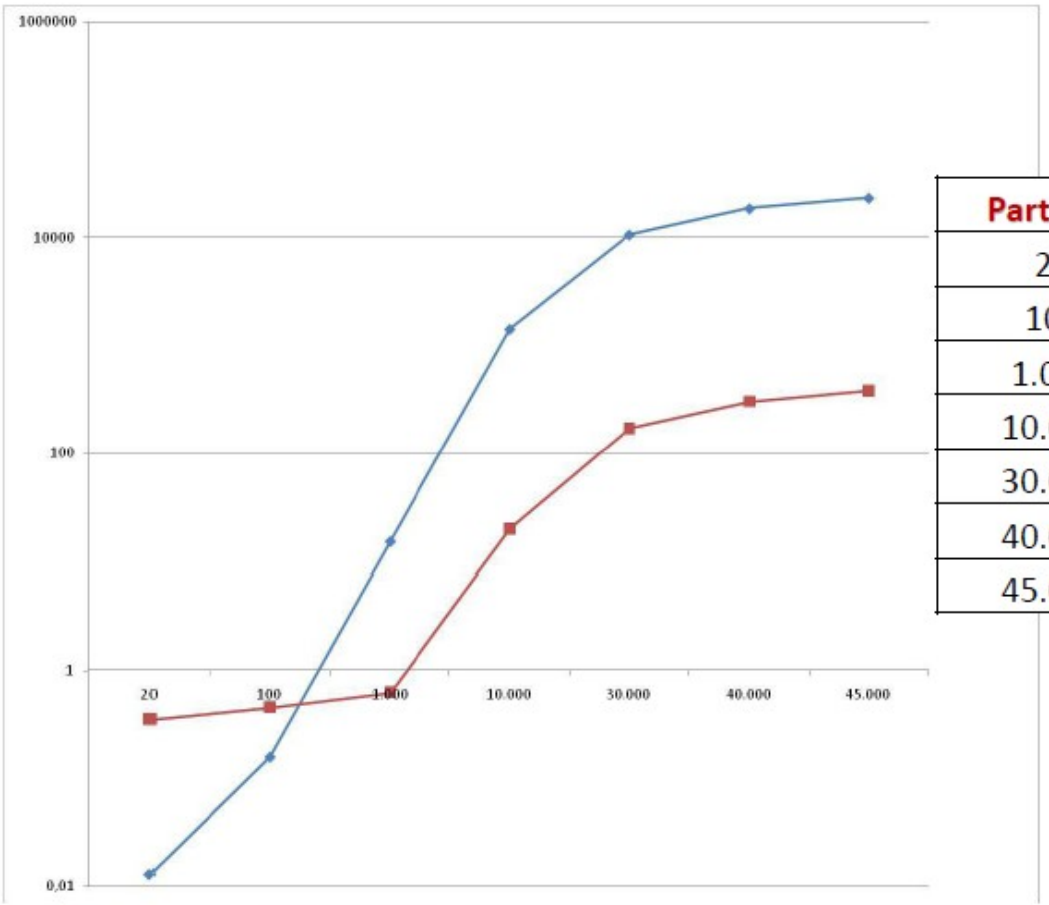
GPU tests

- GPUs are the graphical accelerators we all have in our laptops/desktops
- Started benchmarking a rather general problem:
  - Given a set of 4-vectors, find all possible pairs and calculate invariant mass
  - Idea is to exploit parallelization in combinatorics
    - Many pairs are processed at the same time
- All of us know how to do it in a normal, non parallel environment
- Some of us have also learnt how to do it on a GPU
  - Requires non-trivial core restructuring
  - In general, it is worth stressing that naïve, home-made programming is no more a viable coding strategy
- Requires all of us to revisit our approach to HEP code
  - Either we all undergo a serious training in “good” programming
  - Or, we delegate the most critical aspects of our code to real experts

http://agenda.infn.it/materialDisplay.py?contribId=343&sessionId=13&materialId=slides&confId=4441

GPU tests

Testing Algorithm



Particles	Seq	Parallel
20	0,013 ms	0,351 ms
100	0,159 ms	0,457 ms
1.000	15,211 ms	0,621 ms
10.000	1.422,742 ms	19,699 ms
30.000	10.637,217 ms	170,23 ms
40.000	18.613,341 ms	301,756 ms
45.000	23.266,988 ms	380,989 ms

Configuration parameters
512 threads for block.
Advanced tuning can improve these results

The Framework

- Why do we need a framework?
- A few common functionalities across all our applications
  - Input/Output from/to disk
  - Efficient (and safe) memory management
  - Parallelization capabilities
  - Runtime configurability
- Instead of having all applications implement their own solutions (as we do now), we can use a common pool of code
  - Initial manpower needed to setup and migrate all existing code
  - Gain in the medium term from removal of redundant maintenance/developments
  - Immediate HUGE gain in usability
    - User can move from one application to another within the same general setup

The Framework

Issues

- In case we take Art as the new Framework, should we use it just as a baseline for future development, or should we use it also for existing software?
- The question is delicate as a (still to be quantified) amount of work has to be done to eventually adapt the existing software
- In general, moving to a concurrent programming paradigm has many implications
  - evident in a simple parallelization exercise where standard containers (`vector`) have to be changed with concurrent ones (`tbb::concurrent_vector`)
  - This is not always possible due to a different interface of `concurrent_vector` w.r.t. `std::vector` (e.g. the lack of the `erase()` method)

Refurbishing legacy code

Parallelism in FastSim [5/5]

Some observations and questions:

- Even if the analysis is still partial, FastSim exposes some parallelism at the module level. We should try to exploit it

Regarding the module definition:

- Does the proposed definition of a module with requirements and products fit our needs?
- In FastSim some modules modify event properties (ex: BtaSelectCandBase). Is this a requirement we can avoid?
- Is there any module written being aware of its execution position in a sequence?

- How much parallelism can we inject into the execution of BaBar legacy code?

http://agenda.infn.it/getFile.py/access?contribId=322&sessionId=1&resId=0&materialId=slides&confId=4441

Conclusions

- An amazing amount of material has been shown at this meeting
  - My apologies to those colleagues whose contributions I could not mention here
- We must all revisit our approach to HEP software
  - It does not grow spontaneously on hard disks
  - It requires a lot of hard work, and a good mix of design, programming skills, ***user feedback***
  - Our input as experienced users of HEP software is crucial in this design/R&D phase
  - Manpower is still an issue
    - If you are willing to join us, drop us an email