Introduction

Summary

Workshop and follow up

List of Topics

Comments on some topics

Computing R&D Wokshop

- site: **Ferrara** is our best option at the moment
  - it's close to Bologna airport (40 Km),
  - cheap hotels, good food, few cars and many bikes on the streets
  - Univ. conference center available free of charge
  - can count on experienced local organizers

- Wed. Feb. 24th (9am) to Fri. Feb. 26th (5pm)
  - possible layout:
    - **initial plenary session** to get started
    - **four slots** of plenary sessions; presentations concentrated on those issues that require more detailed study
    - **four slots** of two to three parallel sessions
    - **two slots** for the final plenary sessions
  - options:
    - would it be more prudent to schedule the initial day on Thursday ?
    - nice to have all people in a single hotel (with sofas for after dinner)

Workshop goal

- Come to the WS with a **list of proposed issues** (and a bunch of physicst and comp. professionals that can be interested in joining the effort)
  - topics we need to address for being in a position to develop the SuperB computing model in 2011 (Computing TDR)
- Leave the WS with an **R&D program proposal**
  - prioritized list of R&D activities
    - quantification of benefits wherever possible
    - estimation of manpower needed and timescale
  - definition of responsibilities for those activities that can be started immediately
  - strategy for dissemination

R&D activity form

- Description, main goal
- Motivations
- Tasks for the workshop
- Work breakdown structure
  - manpower needed
- Collaborations
- Schedule
- Reference material

- available now (~ before the end of the year)
- available by the end of the WS

WBS

Articulation of the activity

activity	task	subtask		*manpower (man-months)* physicist w. comp. expertise	*junior comp. prof.*	*senior comp. prof.*	*total*
	1		identify the most data-volume demanding data processing applications foreseen for SuperB and their requirements	**1**			**1**
	2		develop models of alternative storage implementations that can satisfy the requirements, based on one or two approaches taken from current HEP experiments vs. a new model based on local disk storage with possible use of SSD trechnology	**1**		**1**	**2**
	3		develop a simulation application that provides quantitative estimates of the performance achievable for the various models		**3**	**3**	**6**
	4		Identify the aspects of the computing model that are affected by the new storage strategy and evaluate the impact	**1**		**1**	**2**
	5		evaluate development costs, TCO and performance, improve the models and finally present a comparison with an indication of the recommended choice for SuperB	**1**		**1**	**2**
			TOTAL	**4**	**3**	**6**	**13**

Workshop follow-up

- Writing the **second white paper** describing the R&D program
- Presenting the program at the SuperB collab. meeting and get it "approved"
- Scheduling:
  - a **mid-way WS** after ~ 6 months
  - a **final WS** after ~ 1 year
- Publicise it for getting **new collaborators**
  - presentation to conferences, seminars in main laboratories, etc.
  - not only among physicists but also in computing science departments

User interfaces

- GUI for running analysis
- access to computing applications and data
- code management tools
- collaborative tools

offline tools and infrastructure

(exploiting developments from LHC exp., etc.)

- general code quality issues: robustness, error handling, performance control, inline qualification
- code and build management
- integration of firmware code, scripts, configuration files, etc.
- release system
  - addressing special online needs
- geometry, conditions, framework
- persistency, event store

migration of BaBar legacy code base to SuperB

- migration of BaBar legacy code base to SuperB
- general code revision for enforcing higher quality standards
- rewrite packages (IFR, Dirc,Track pattern recognition, ...)
- modernize packages (Kalman fit, EMC reco, Beta)
- redesign data structures (MC Truth, ...)

exploitation of modern CPUs

- many-cores, multithreads
- vectorization
- deeper parallelism
- optimization

Storage efficiency and scaling

- de-centralized event store
- exploitation of SSD storage technology

distributed computing

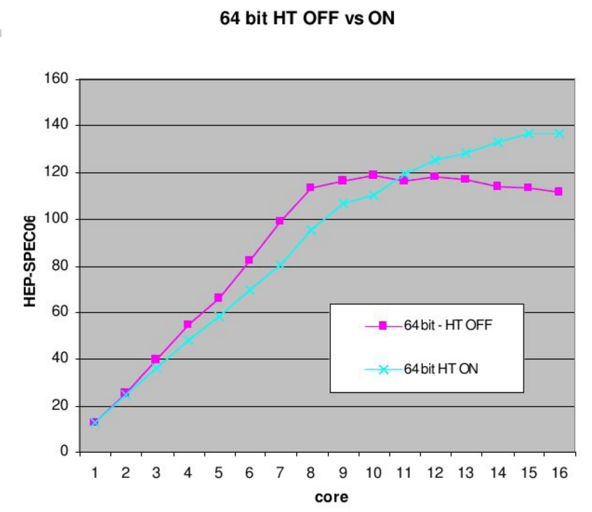
- develop a model defining the requirements
- evaluate the constraints for SuperB computing model and code development
- data bookkeeping
  - common system with online

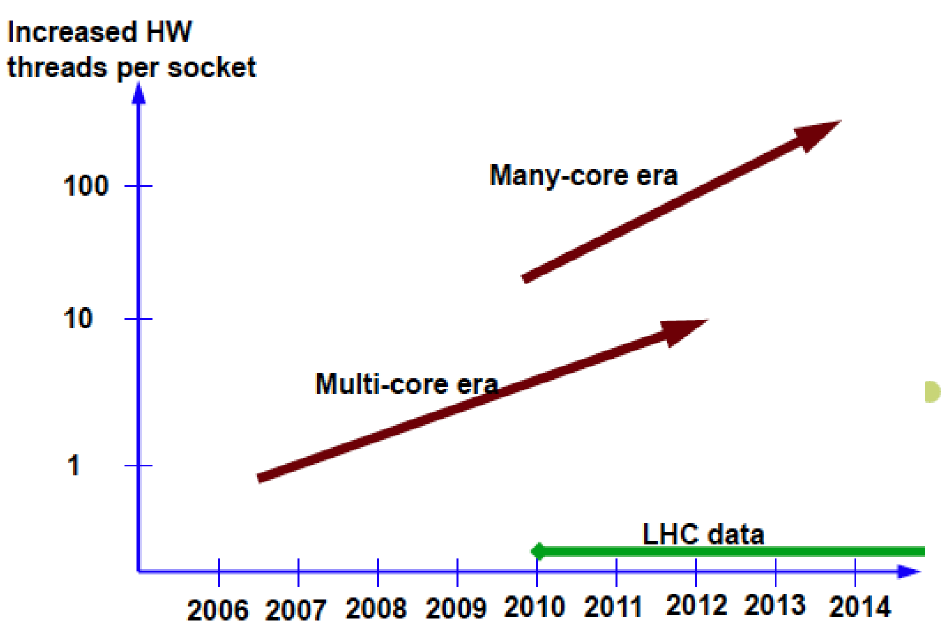
Online specific topics

- Support for Raw data versioning
- Decouple container size (e.g.: files) from event grouping (e.g.: runs)
- farm management: make sure of what machine are running and how they are configured
- design a flexible offline build/release/deployment system to mitigate the constraints on evolution of online data (format/content) and DB schema

core and threads

- transition from multy-core to many-core is underway
  - core = indipendent execution unit
    - CPU external channels may be shared
- new CPU also support the Symmetric Multi Threading
  - thread = only program counter ar register files are independent
    - execution logic and caches are shared



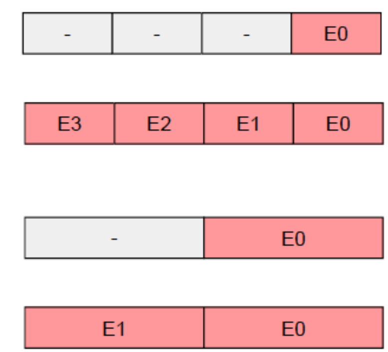


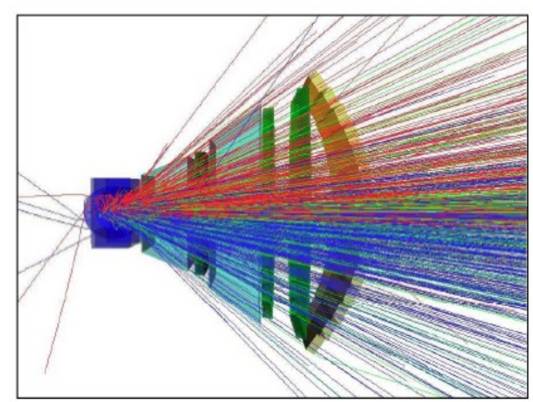
From "Platform 2015: Intel Platform Evolution for the Next Decade" (S.Borkar et al./Intel Corp.)

Vector instruction sets (SSE)

- CPU now have **128 bit** istructions/registers
  - not exploiting means a 2x t
    4x **peak capacity loss**
- next CPUs:
  - Advanced Vector eXtensions (256 bits)
- exmples of exploitations:
  - CBM/Alice track-fitting with 4-packed SP --> gain **4x**

- **Single precision**
  - Scalar single (SS)
  - Packed single (PS)
- **Double precision**
  - Scalar Double (SD)
  - Packed Double (PD)

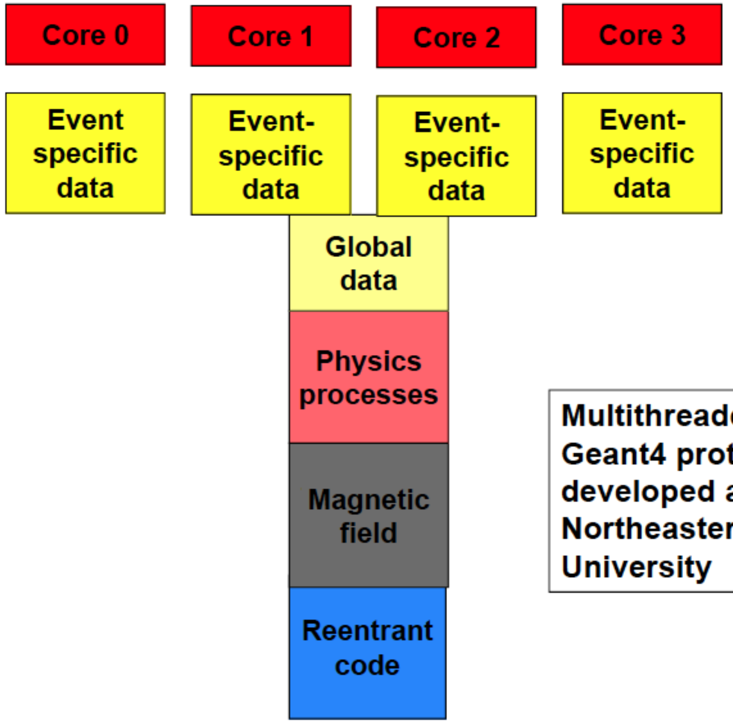




I.Kisel/GSI: “Fast SIMDized Kalman filter based track fit”
http://www-linux.gsi.de/~ikisel/reco/CBM/DOC-2007-Mar-127-1.pdf

Consequences

- natural parallelism based on event-by-event dispatching will not work:
  - I/O channels to RAM too slow
  - excessive amount of RAM
- one will have to
  - introduce parallel processing at a deeper level
  - share data and code stored in the RAM by different threads or different cores
- eg.: GEANT4 experience quite encouraging
    - only 22 MB per thread !

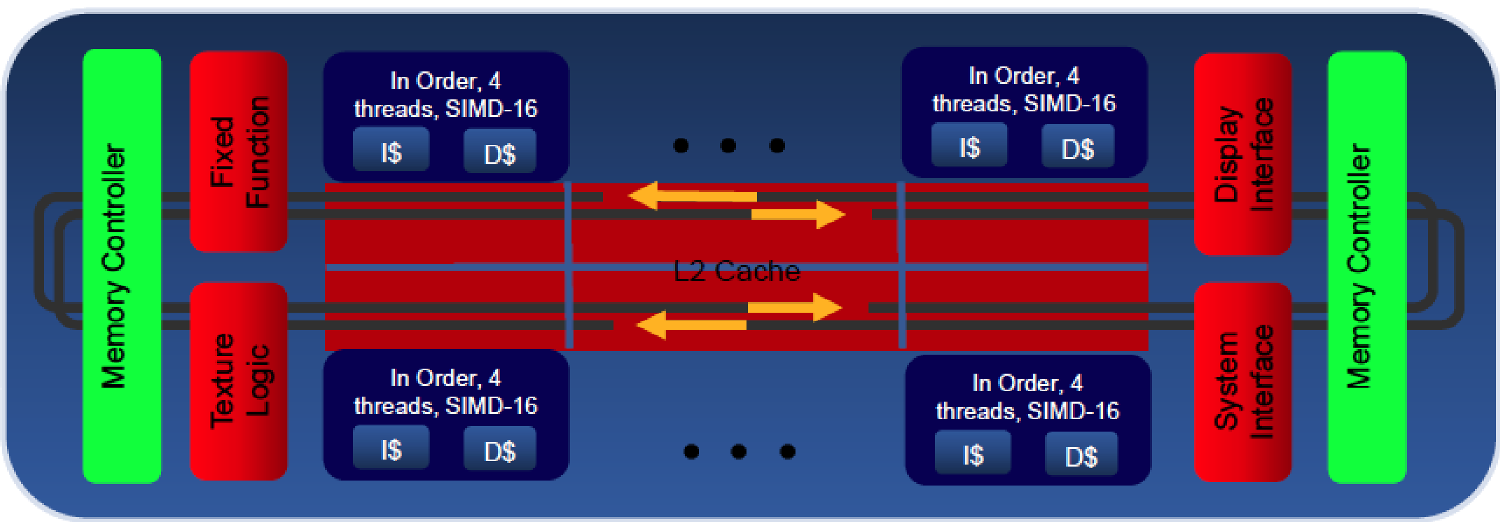


Multithreaded Geant4 prototype developed at Northeastern University

GPUs

Availability of GPUs based on x86 architectures will open up more possibilities

- **Intel's Larrabee:**
  - Already announced at SigGraph 2008!
  - Based on the x86 architecture
  - Many-core + 4-way multithreaded + 512-bit vector unit



Storage

Crucial area for the computing model:

- critical performance issues
- computing main cost driver

What topics should be address ?

- exploitation of new SSD technology
- new storage architecture: de-centralization ?

today...

eff. media 61%

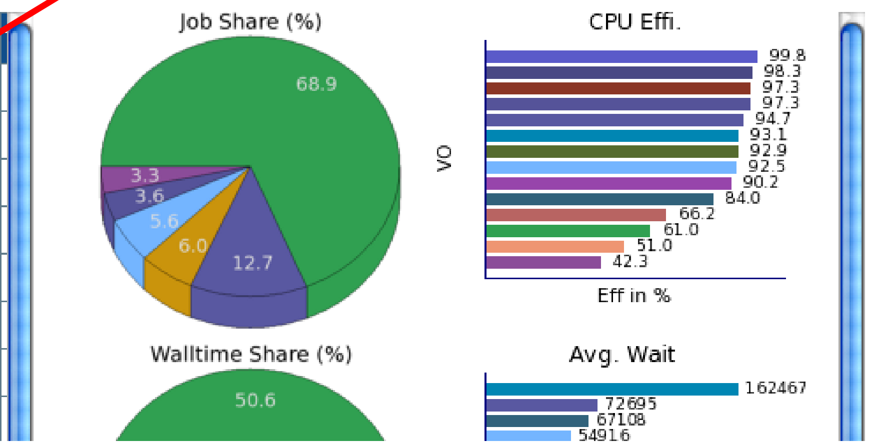
- ~ 85 % produzione
- ~ 25% analisi

LSF Monitoring at Pisa

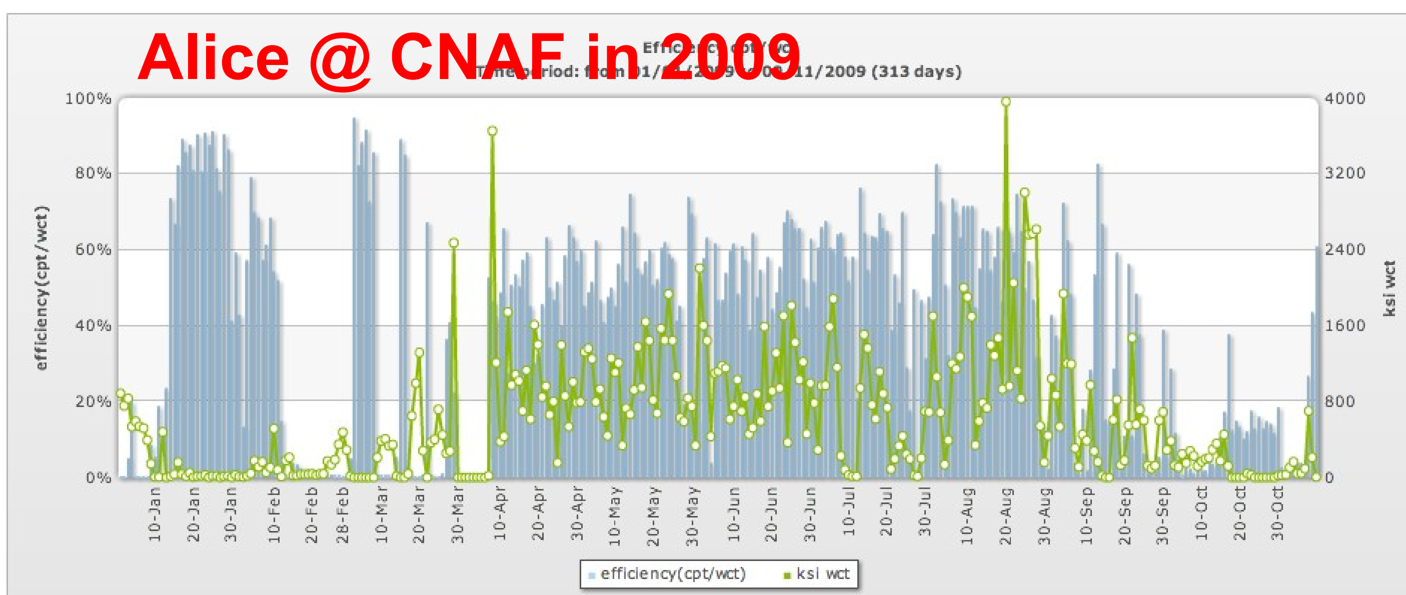
Current Status | Last 6 Hours | Last 12 Hours | Last Day | Last Week | Last Month | Last 3 Months | **Last 6 Months** | Last Year

Jobs completed during the last 6 months

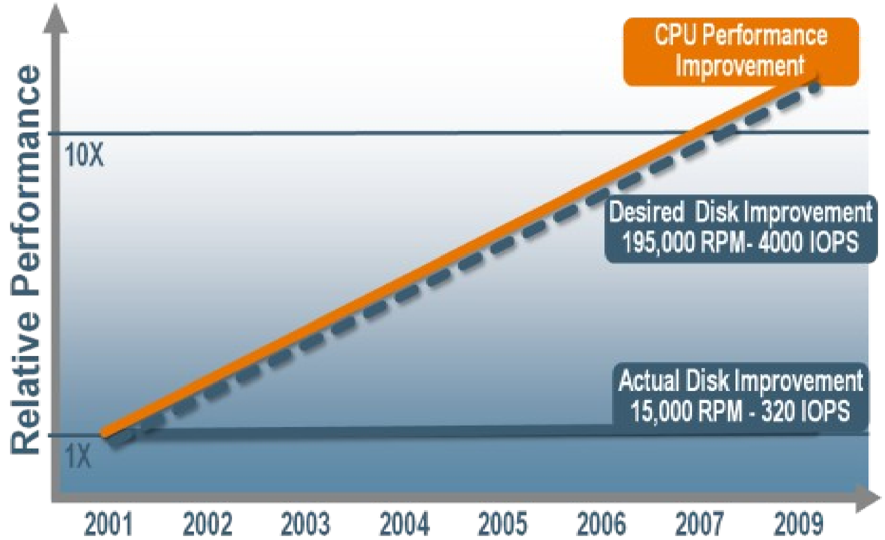
VO/Group	Total Jobs	Succ Jobs	Succ Rate(%)	Walltime (sec)	CPU Time (sec)	CPU Eff(%)	Walltime Share(%)
cms	758897	739886	97.49	10807365895	6596150199	61.03	50.61
theoinfn	139876	132312	94.59	6199161815	5868599471	94.67	29.03
theodip	39305	34665	88.19	2089834072	2032975455	97.28	9.79
theophys	14398	14164	98.37	822206252	807875270	98.26	3.85
glast	61524	60979	99.11	594504032	550109851	92.53	2.78
lhcb	16938	16731	98.78	324862637	301875985	92.92	1.52
biomed	8742	7291	83.40	203056714	36130934	17.79	0.95
cmsprt	8630	8371	97.00	159607801	15475423	9.70	0.75
cdf	947	939	99.16				
atlas	6109	5985	97.97				
theolong	750	678	90.40				
compchem	904	886	98.01				
gridit	799	794	99.37				
ops	35924	35643	99.22				
virgo	331	327	98.79				

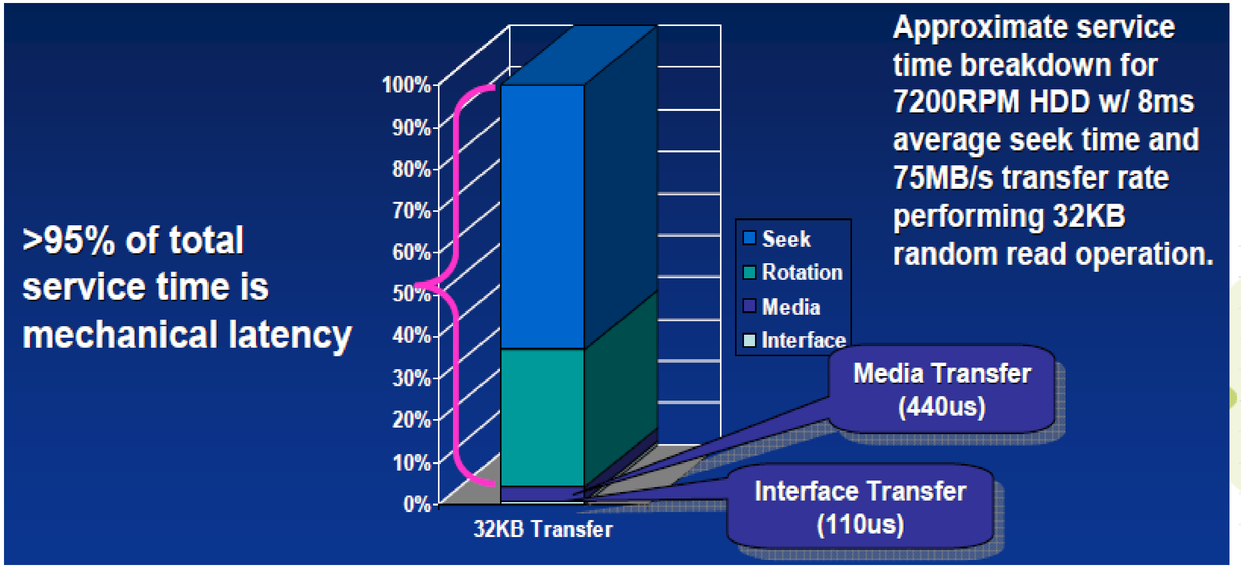


Last updated on 2009-11-10 14:10:58



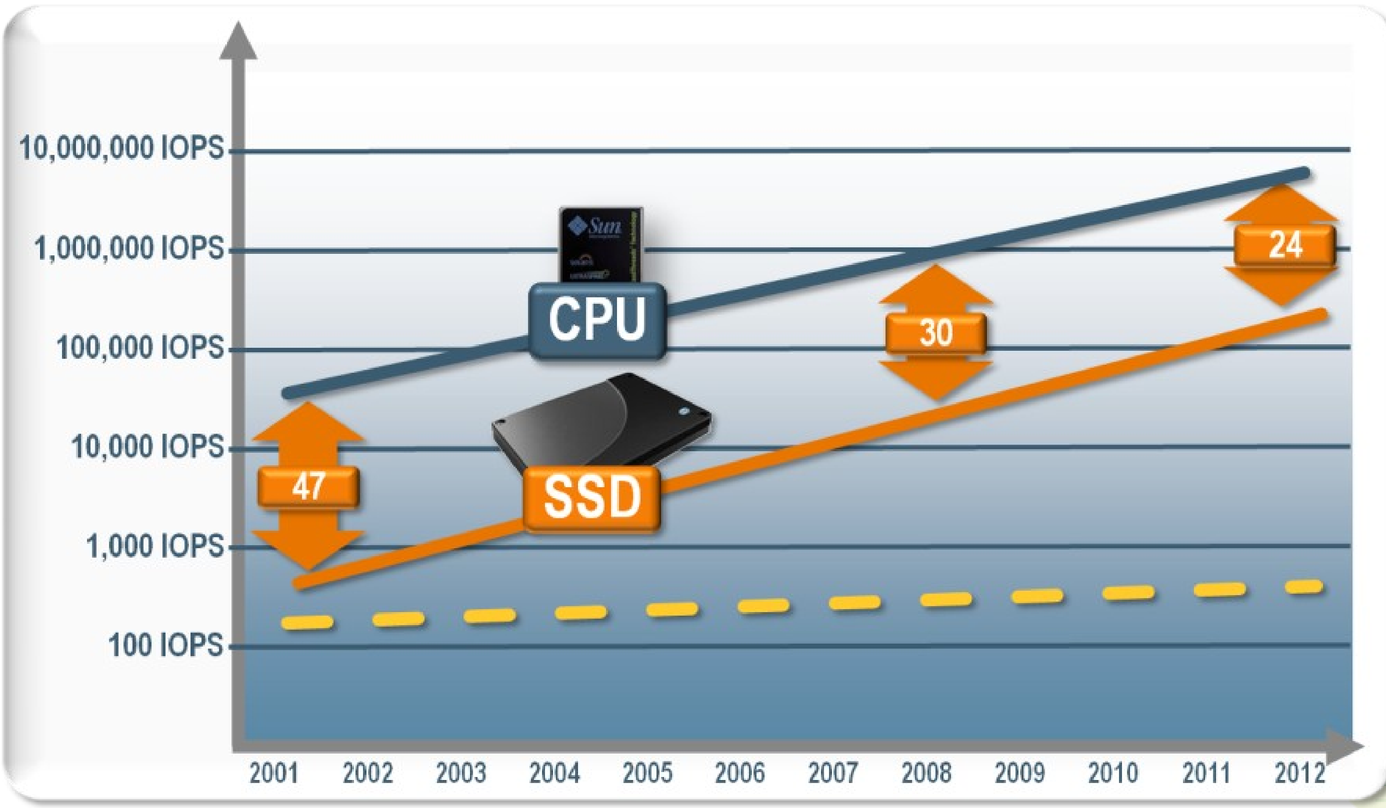
DISK/CPU performance mismatch





SSD

Le memorie persistenti a stato solido possono cambiare radicalmente il quadro

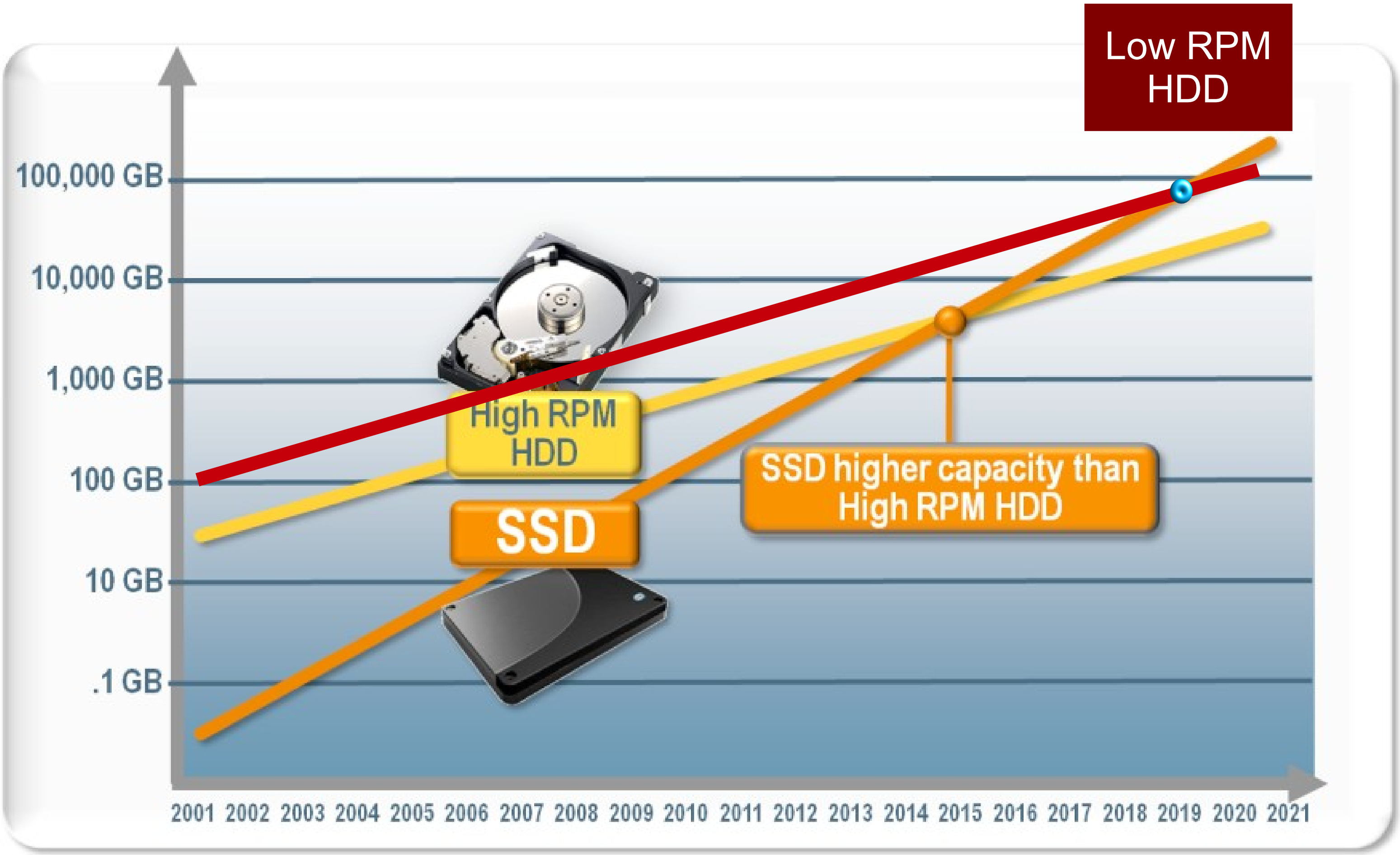


The old and the new

- Enterprise HDD
  - > 180 Write IOPS
  - > 320 Read IOPS
  - > 300 GB
  - > ~18W
- $ per IOPS: 2.43
- IOPS/W: ~14

- Enterprise SSD
  - > 7,000 Write IOPS
  - > 35,000 Read IOPS
  - > 32GB
  - > ~3W
- $ per IOPS: 0.04
- IOPS/W: ~7000

but HD will still be around for a while

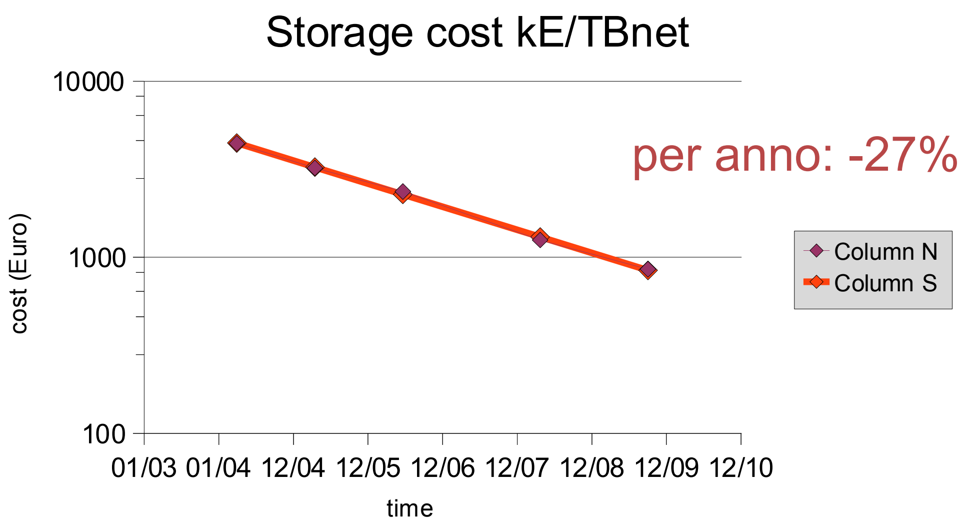


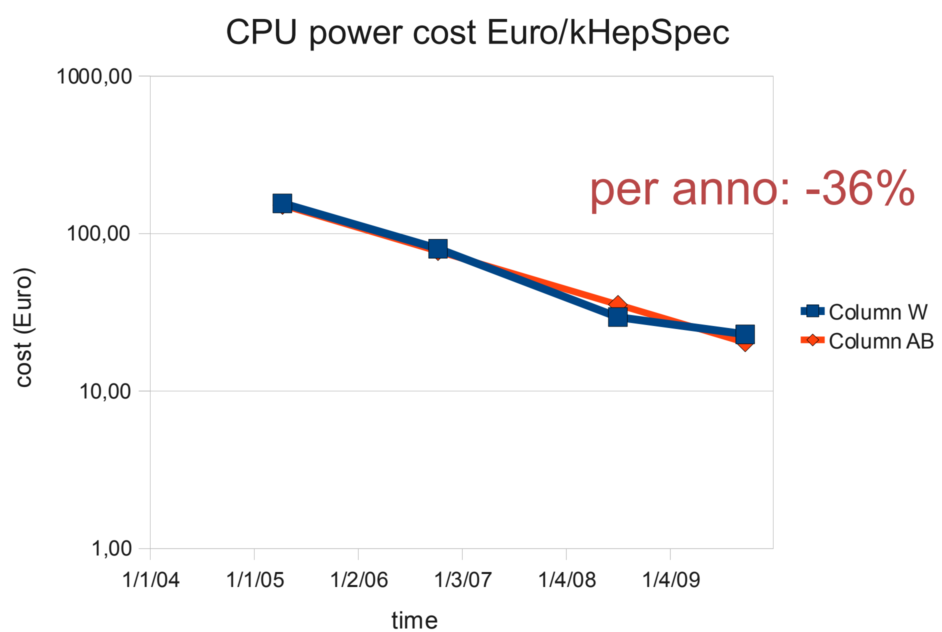
Meanwhile

- storage system will be thr
  SSD - HD - tape
- it is not clear that data intensive applications one can get optimal performance just using SSD as storage caches in a transparent way

Evolution of CPU vs. Storage costs

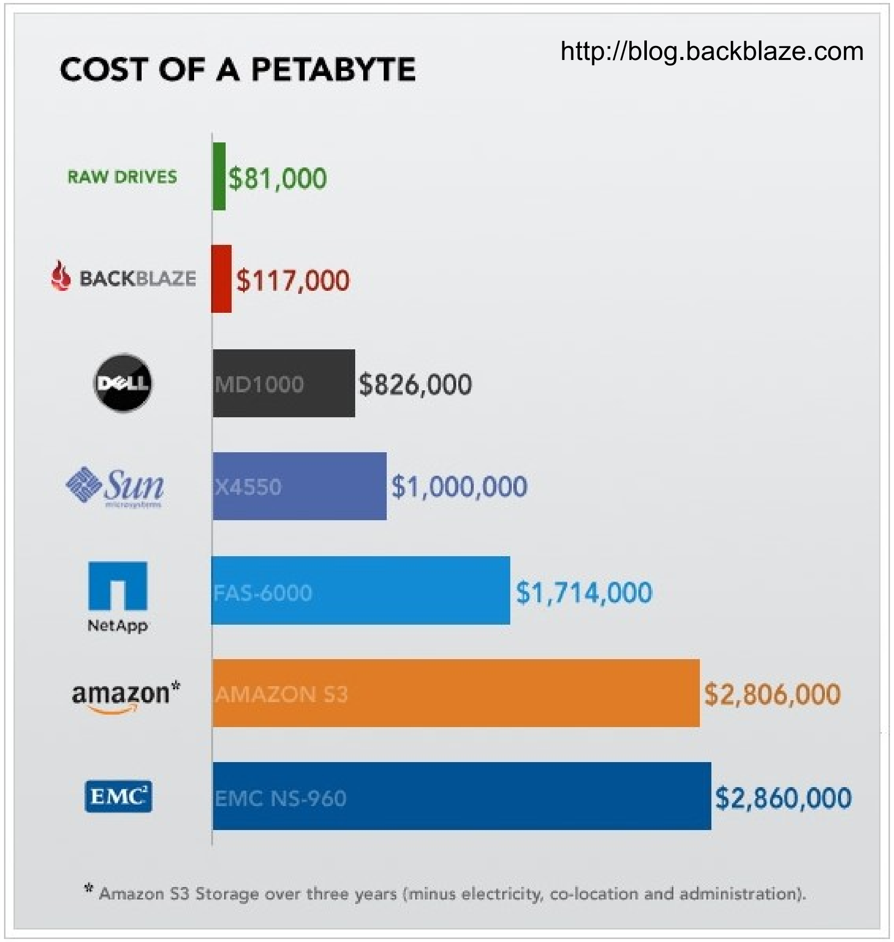
- if we assume that CPU power and storage space scale in the same way
  - tipically with int. lumin.
- storage cost is rapidly beoming dominant w.r.t. CPU
- in 5 years, per Euro:
  - CPU capacity x 9.5
  - Storage capacity x 4.5





Storage costs drivers

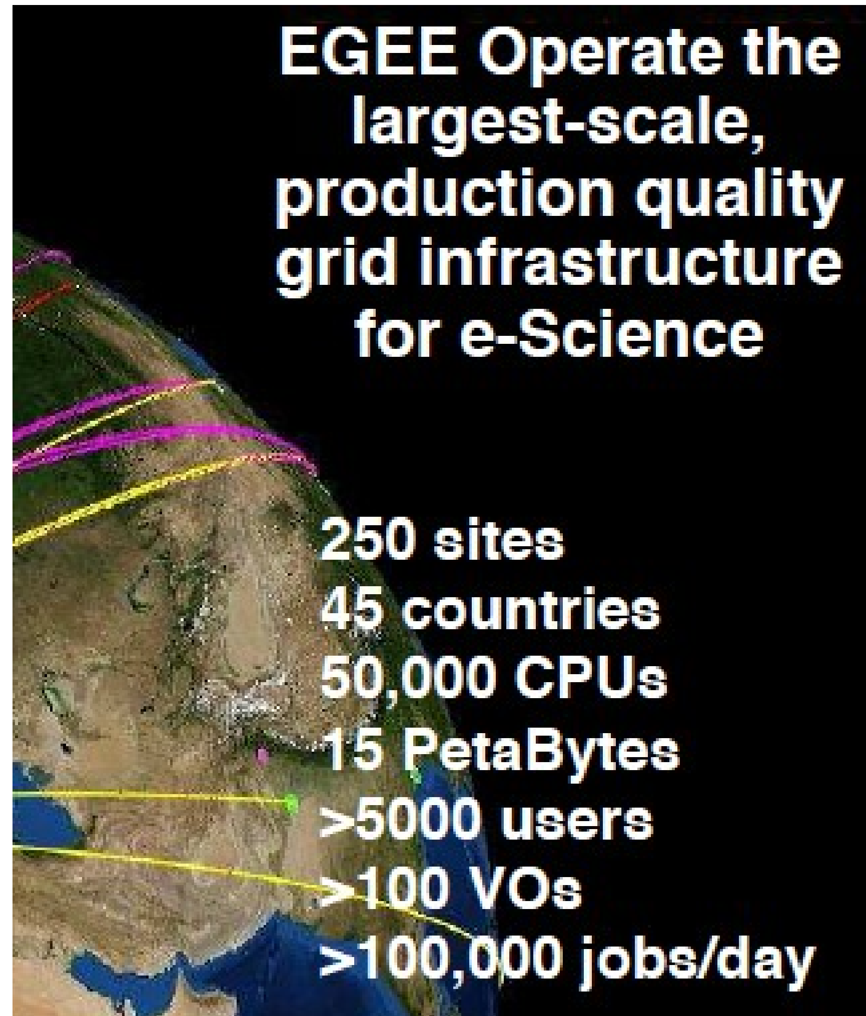
- disk drives costs < 10% total storage system costs
- due to:
  - hardware redundancy, high performance servers, interfaces and networks, caches, SAN infrastructures, ecc.
- but infrastructure costs don't seem to scale as disk drives do



A useful comparison

- 200 clusters
- per cluster:
  - 1000s machines
  - 4+ PB files system
  - 40 GB/s read/write load

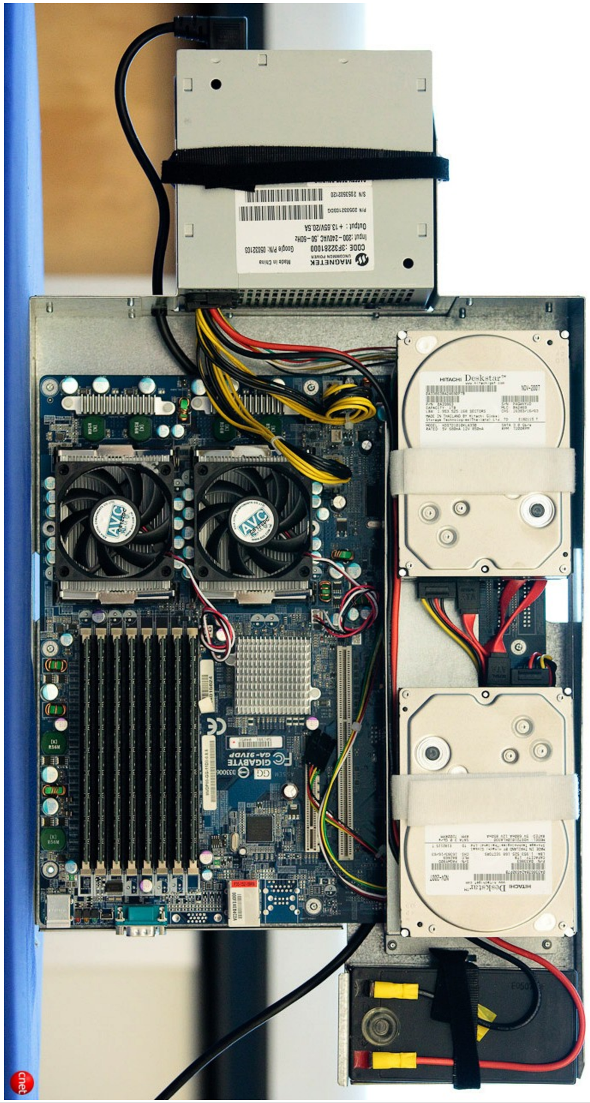
Google



approaching storage differently

- Google approach to computing:
  - **maximize performance per $**
  - **hardware fails, fix it by software**
    - no RAID, no expensive disks, no SAN, no special disk servers
    - data is replicated x3
    - energy saving too:
      - 12 V P.S., no UPS, lead battery in each server
  - **run the application as close as possible** to the data

the Google machine



Belle's implementation

High Performance Data Analysis for Particle Physics using the Gfarm le system
Journal of Physics: Conference Series 119 (2008) 062039

- Analysis data sets
  - mdst data sets for several categories of event: hadronic total sample: 30 TB of event data + 100 TB Monte Carlo
    - event are indexed by skimming
- Analysis farm
  - ~ 1140 nodes (2x3.6GHz Dual Xenon) w/ 72 GB disk
  - 1 PB disk storage on file servers
  - comp. nodes to file servers bandwidth 6+ GB/s
- The problem:
  - it takes a long time to go through the full data sample (one week! few hundred MB/s aggreg.)

Gfarm file system

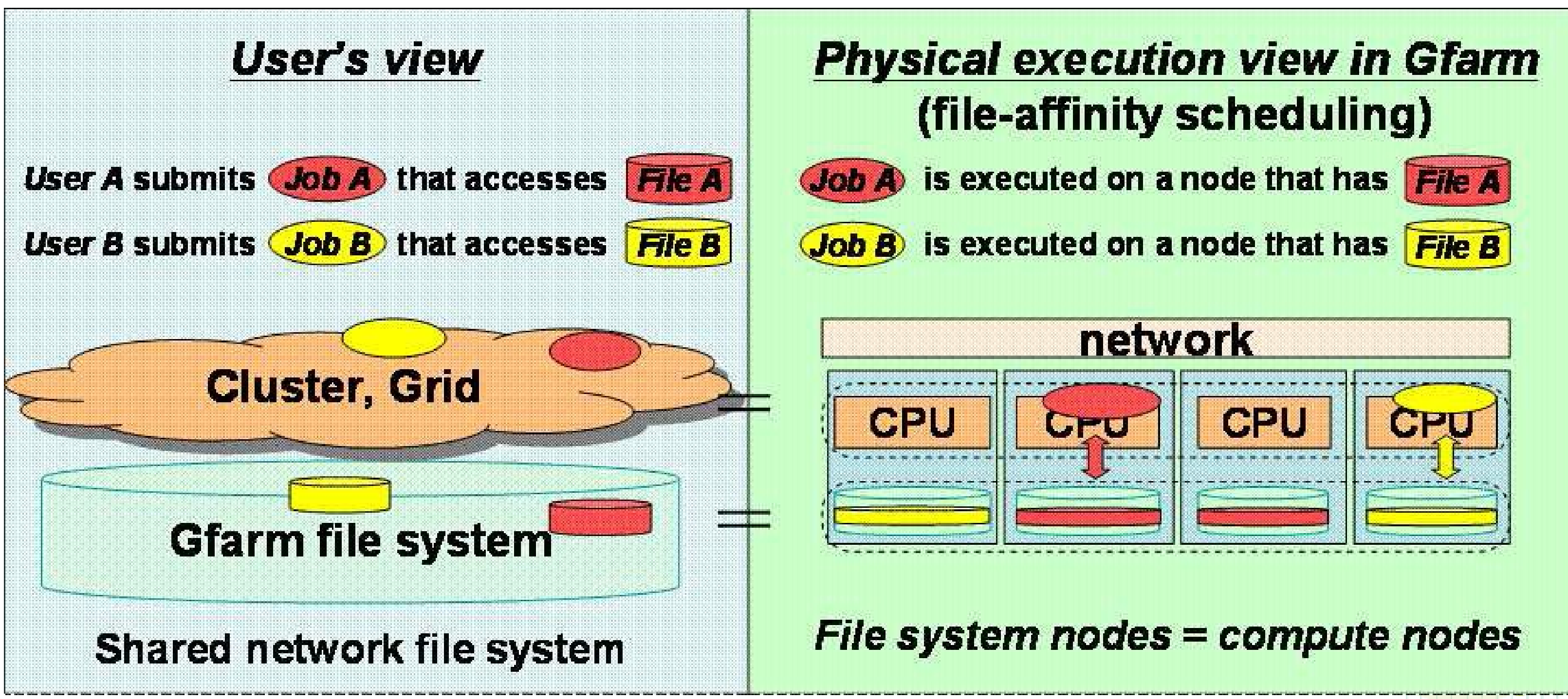
High Performance Data Analysis for Particle Physics using the Gfarm le system
Journal of Physics: Conference Series 119 (2008) 062039

- Wanted to move to a de-centralized file ystem
  - GFarm file system was selected because:
    - it federates multiple disk servers into a single namespace
    - it runs in user space (via Linux Fuse, no kernel mod.)
    - it handles replicas
    - it doesn't require modifications of user code
  - Gfarm writes and reads files where it's most convenient:
    - local disk, if possible
    - otherwise close and least busy node
  - File metadata are kept on a central server
    - metadata are cached in multiple copies for improving access performance

Scheduling

High Performance Data Analysis for Particle Physics using the Gfarm le system
Journal of Physics: Conference Series 119 (2008) 062039

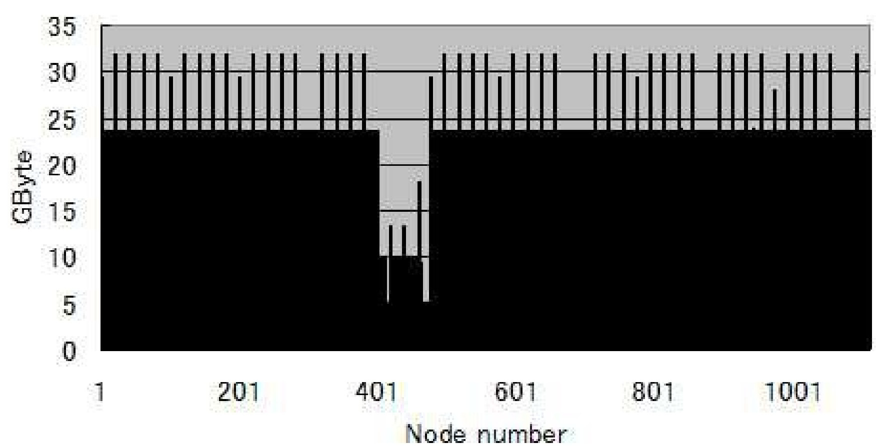
- Gfarm also provides "scheduling by affinity"
  - jobs run on the idlest node that keeps a local copy of the required file



Test setup

High Performance Data Analysis for Particle Physics using the Gfarm le system
Journal of Physics: Conference Series 119 (2008) 062039

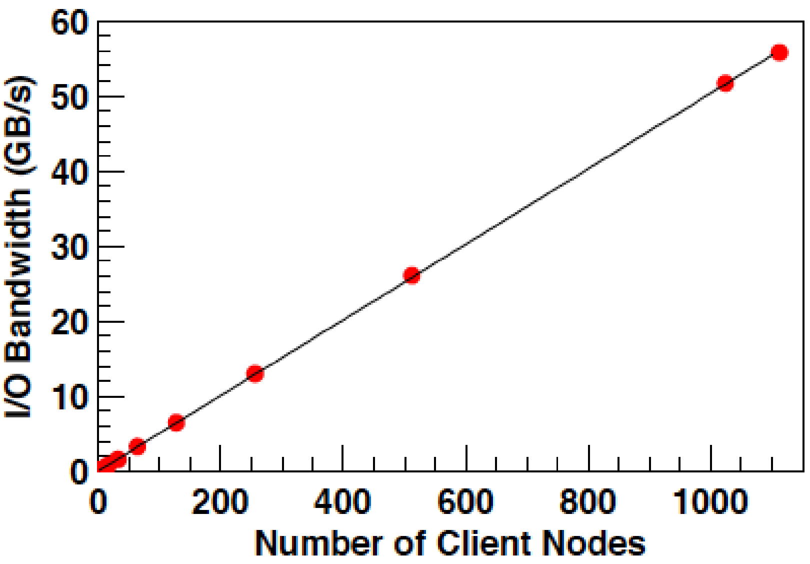
- 1112 nodes
  - + 1 metadata server
  - + 3 metadata cache server
- 24.6 TB of data on local disks
  - ~ 20000 files (runs), size from 100 MB to 23 GB
  - copying the files to the Gfarm file system, evenly distribute the files across the nodes
  - each node provides max 54 MB/s read throughput



Scalability

High Performance Data Analysis for Particle Physics using the Gfarm le system
Journal of Physics: Conference Series 119 (2008) 062039

- I/O benchmark
  - up to the physical limit 52 GB/s aggregated bandwidth



- Skimmink app.
  - looking for high energy gamma in *B* --> *sγ* events
  - 24 GB/s on 704 nodes
    - 34 MB/s average on each node
  - took 15 minutes instead of 3 weeks to run the skimming