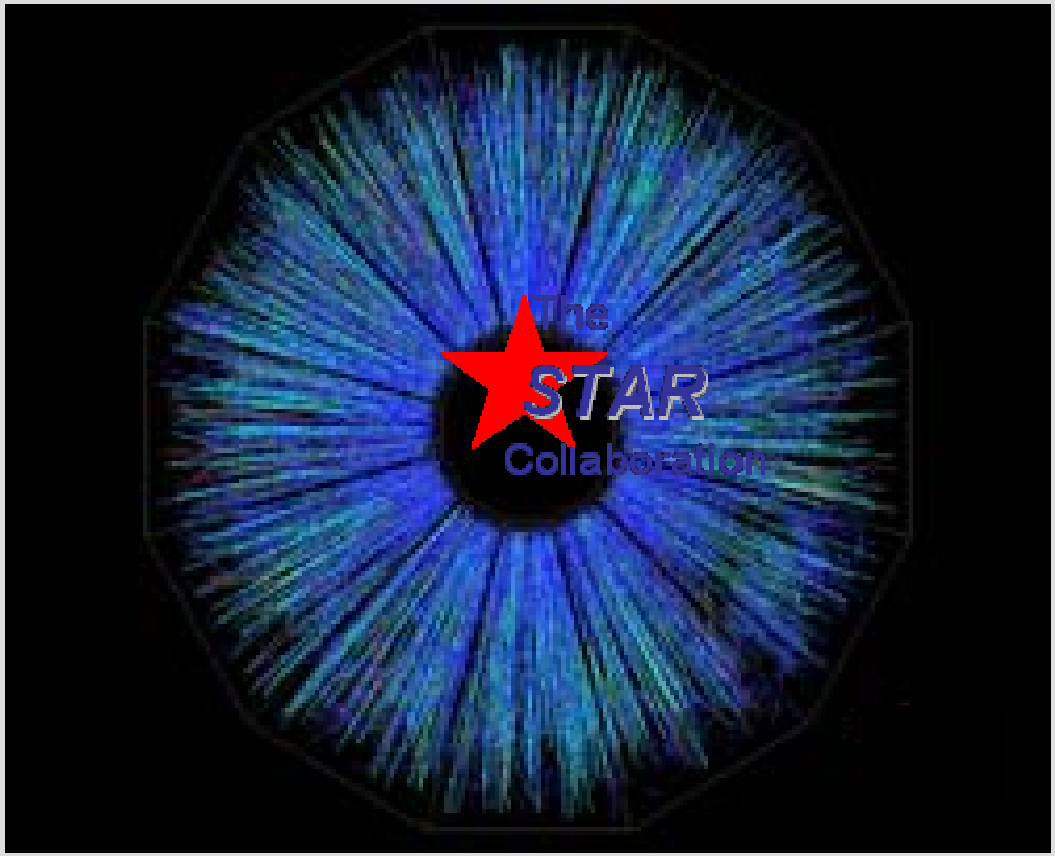
STAR Computing

Doug Olson for:
Jérôme Lauret

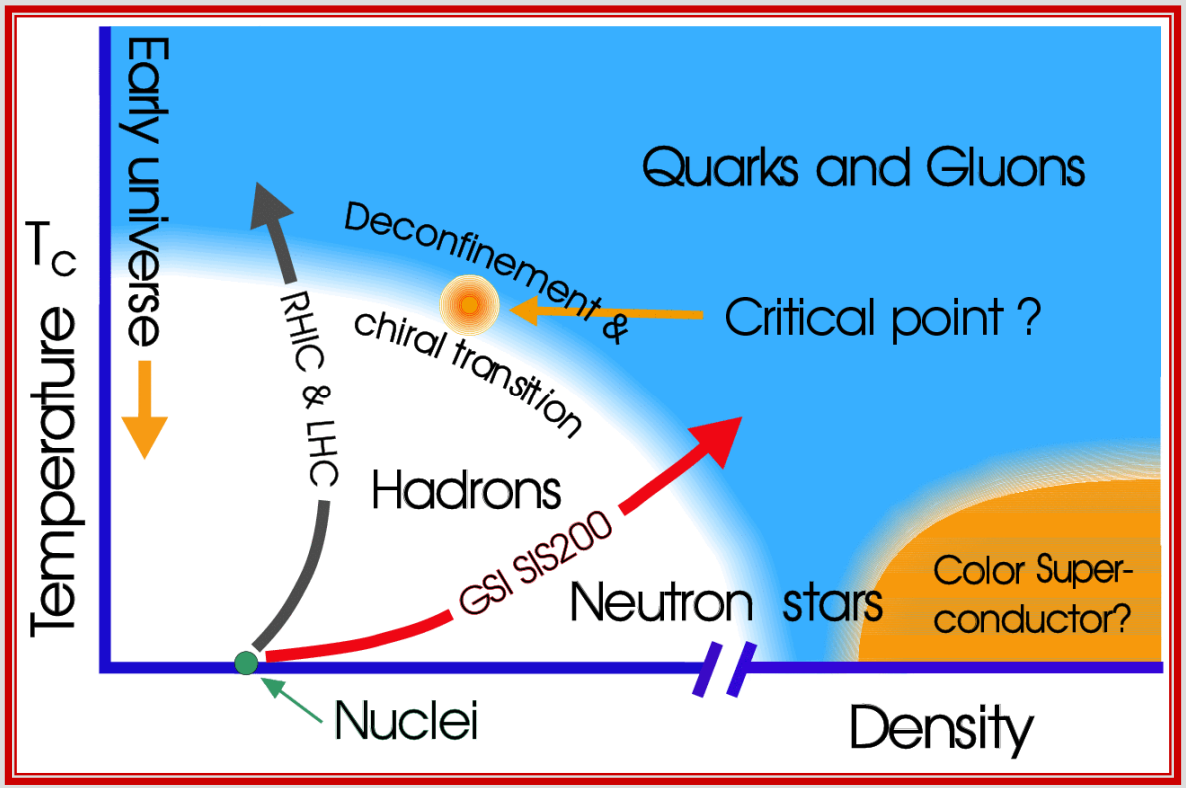


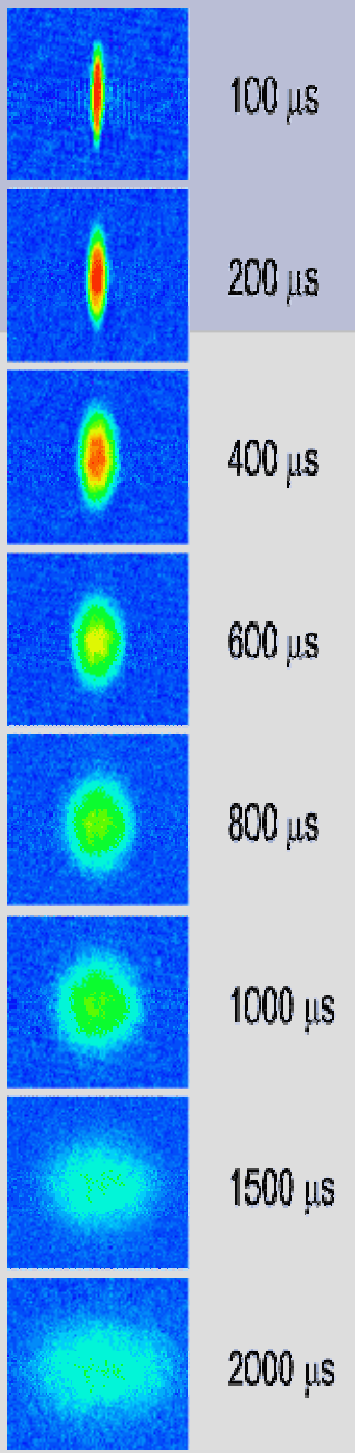
STAR experiment ...

- **The Solenoidal Tracker At RHIC**
  - http://www.star.bnl.gov/ is an experiment located at the Brookhaven National Laboratory (BNL), USA
  - A collaboration of **586 people** wide, spanning over **12 countries** for a total of **52 institutions**
  - A Pbytes scale experiment overall (raw+reconstructed) with several Million of files

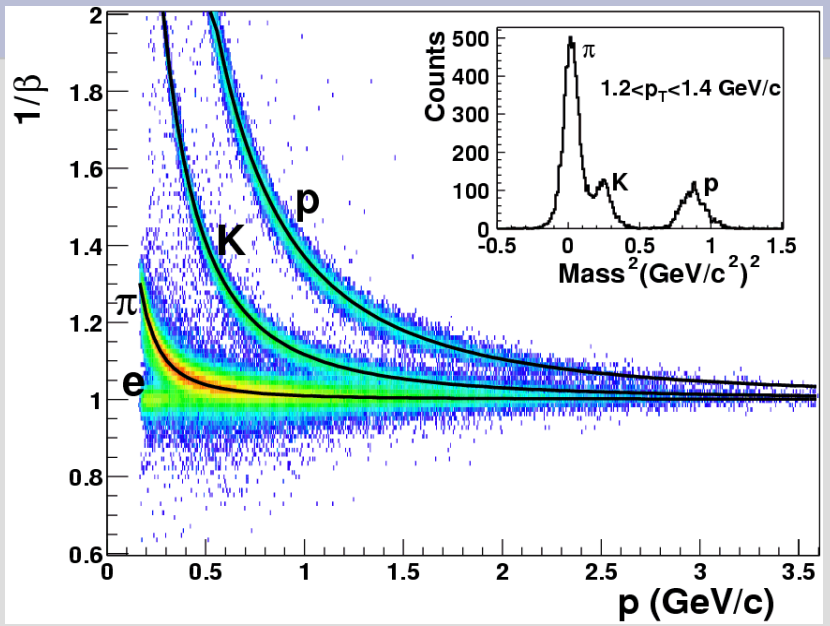
The Physics …

- **A multi-purpose detector system**
  - For Heavy Ion (Au+Au, Cu+Cu, d+Au, …)
  - For Spin program p+p

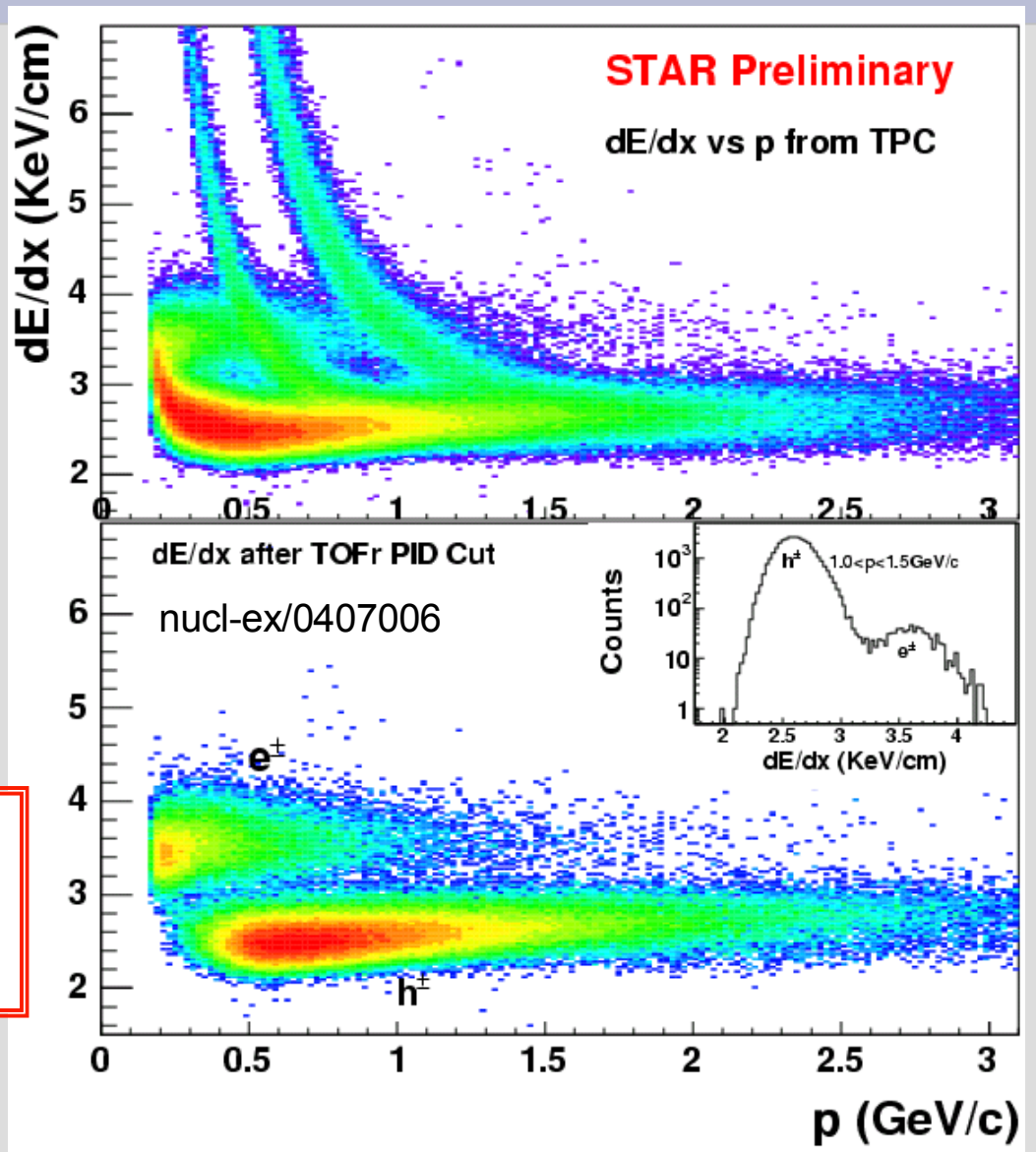




Zhangbu Xu, DNP2004



Hadron identification:
STAR Collaboration, *nucl-ex/0309012*

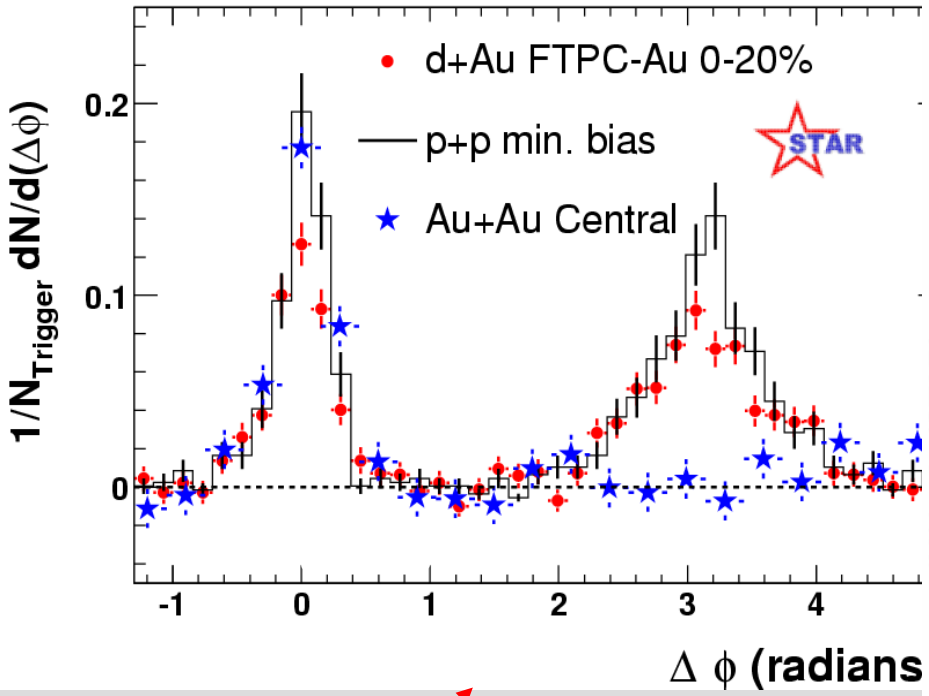


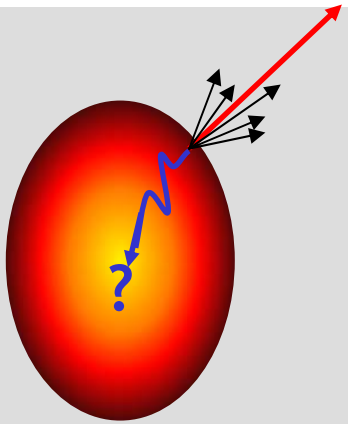
Electron identification:

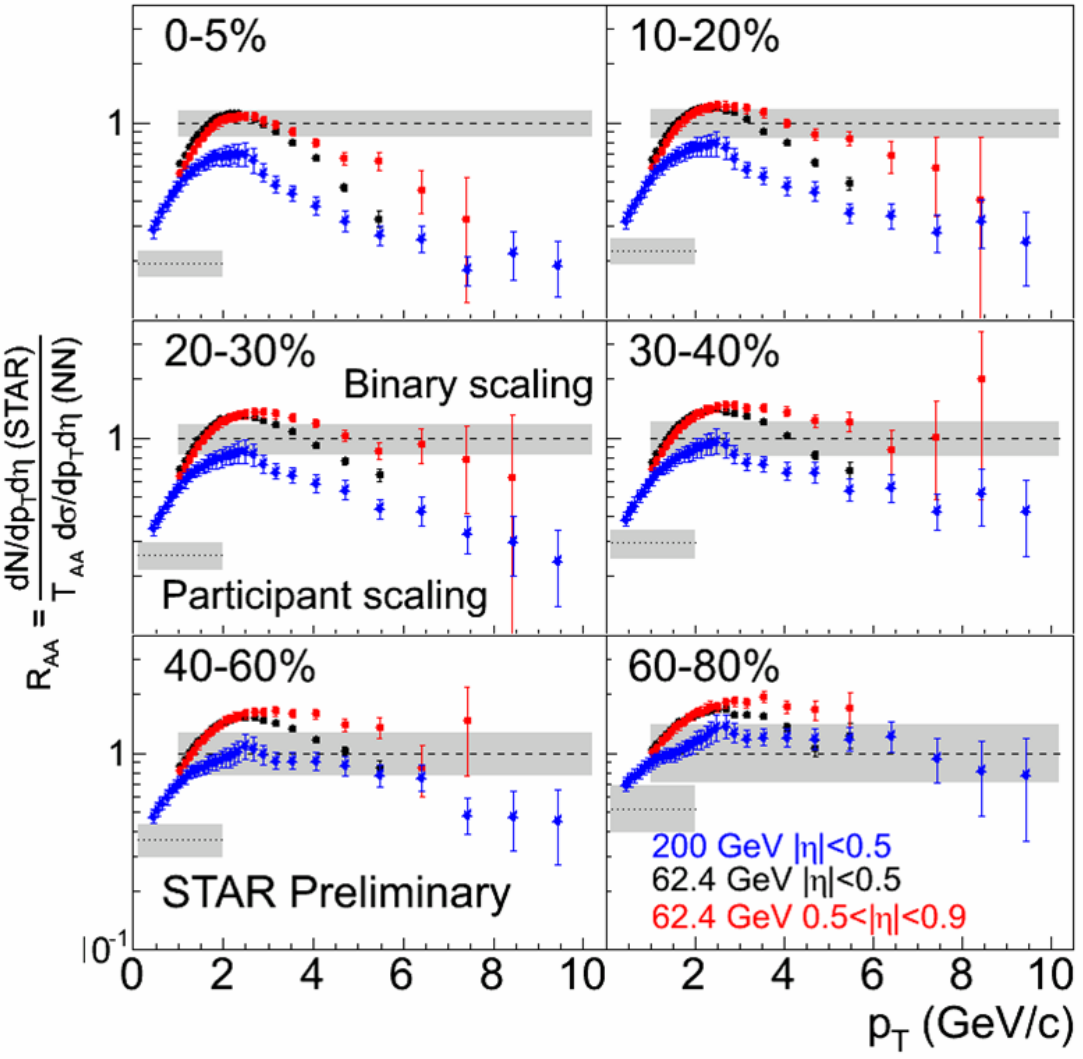
TOFr $|1/\beta - 1| < 0.03$

TPC dE/dx electrons!!!

Carl Gagliardi, Hard Probes 2004

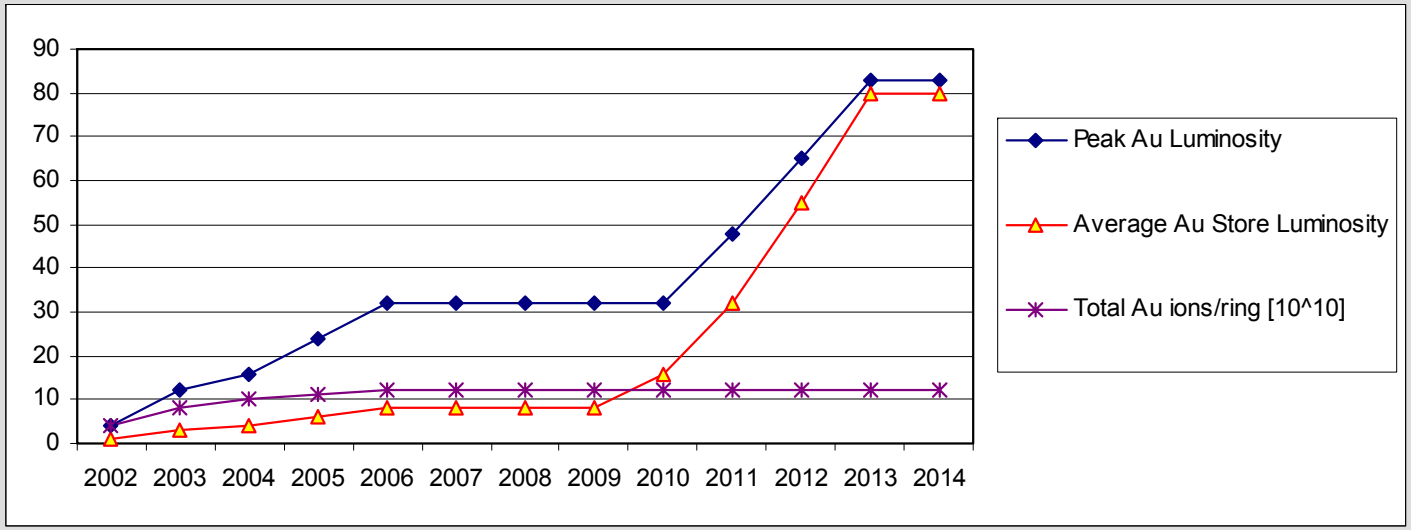






Data Acquisition Prediction

- **150 MB/sec**
  - 60% Live, 3-4 months running => 1+ PB of data / run
- **Possible rates x10 by 2008+**
  - x2 net output requested, the rest will be trigger
  - Is needed to satisfy the Physics program …
  - But pose some challenges ahead (RHIC-II era)



Data Sets sizes - Year4

- **Raw Data Size**
  - <> ~ 2-3 MB/event - All on Mass Storage (HPSS as MSS)
  - Needed only for calibration, production – Not centrally or otherwise stored
- **Real Data size**
  - Data Summary Tape+QA histos+Tags+run information and summary: <> ~ 2-3 MB/event
  - Micro-DST: 200-300 KB/event
- **Total Year4**

Total num events	138260234
GB total	357369,72
TB total	348,99
MuDst	34,9

Data analysis

- **Offline**
  - A single framework (root4star) for
    - Simulation
    - Data mining
    - User analysis

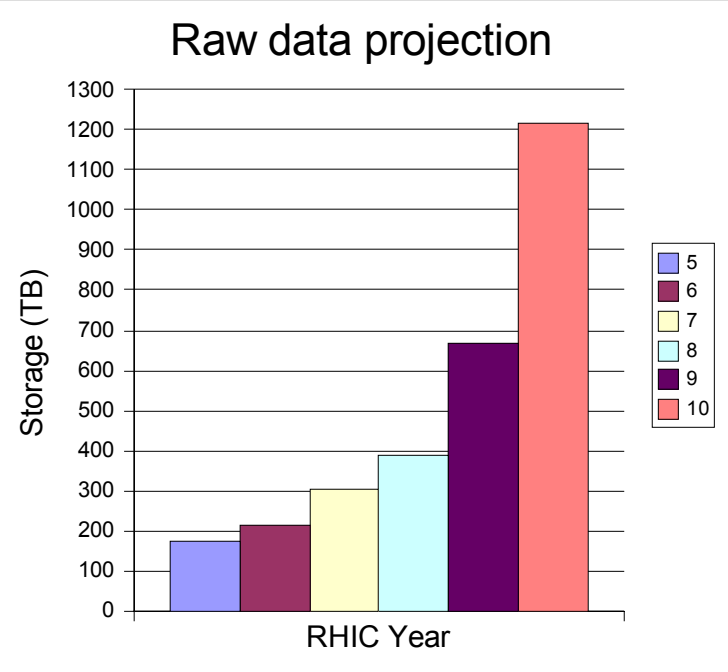
- **Real-Data Production**
  - Follows a Tier0 model
  - Redistribution of MuDST to Tier1 sites
- **Simulation production**
  - On the Grid …

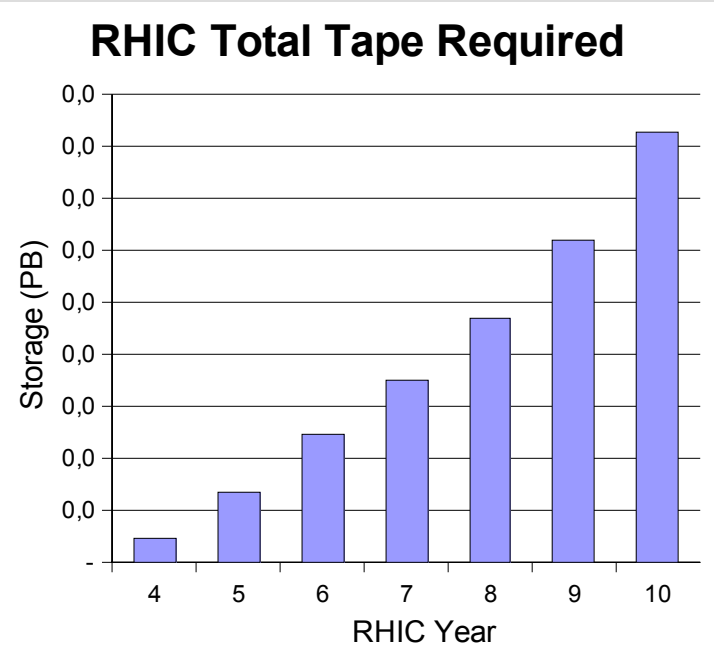
How much data – What does this mean ??

Data Sets sizes
Tier0 Projections

2003/2004 data

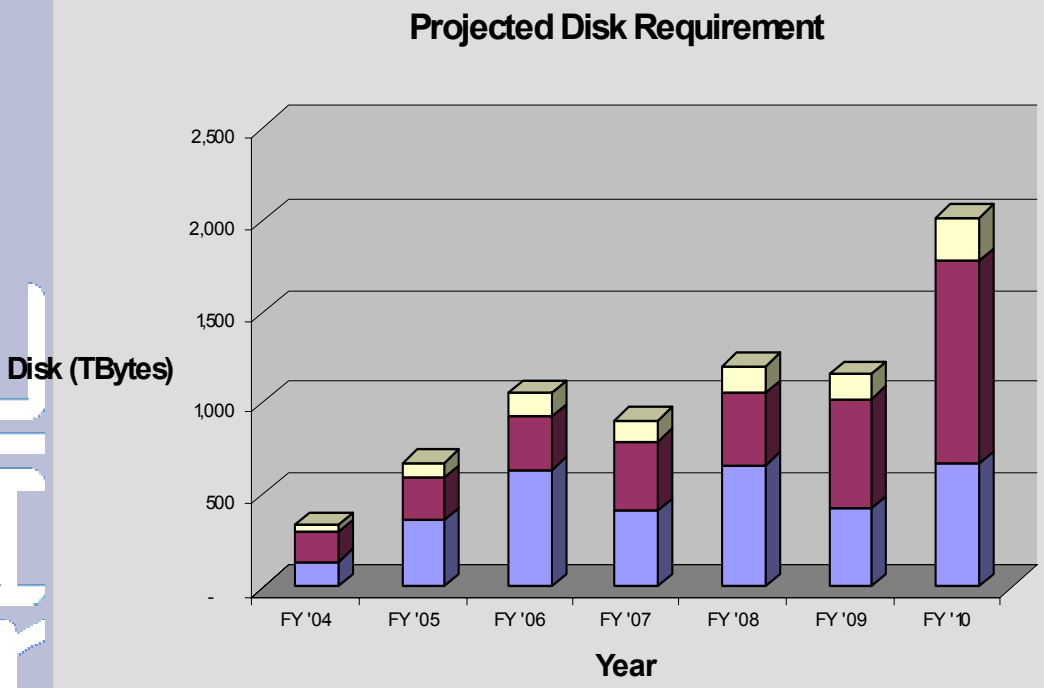
Experiment	Raw (TB)	Pass1 (TB)	# events (M)	#of files
PHENIX	250	800	2000	160000
STAR	200	400	215	399000
PHOBOS	36	72	360	36000

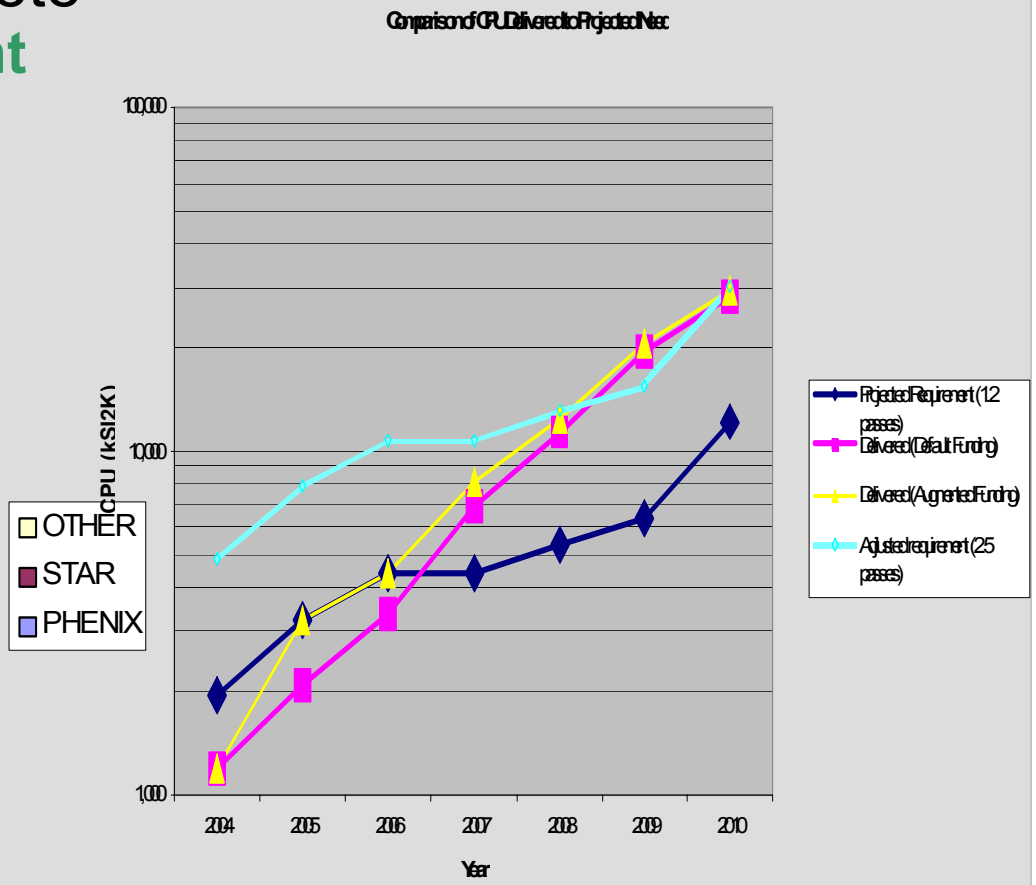




CPU need projections

- **An evolution and projections for the next 10 years (tier0)**
  - All hardware becomes obsolete
    - **Includes a 1/4 replacement every year**





How long ?

Trigger	Total month	Remains	FF (* DF)
production62GeV	128,06	19,11	*0,49*
pp	10,33	0,84	*0,04*
ppMinBias	13,76	0	*0,05*
ProductionPP	74,91	1,41	*0,28*
ProductionPPnoBarrel	8,29	0,51	*0,03*
ProductionPPnoEndcap	1,53	0	*0,01*
ProductionCentral	18,42	17,09	*0,07*
ProductionHalfHigh	23,52	1,38	*0,09*
ProductionHalfLow	161,75	2,52	*0,61*
productionMinBiasHT	0,55	0,55	*0,00*
ProductionMinBias	395,24	69,63	*1,50*
ProductionHigh	267,50	246,15	*1,02*
ProductionLow	1362,49	1306,73	*5,17*
ProductionMid	328,13	317,49	*1,25*
			9,36
			10,54

Year scale production cycles

This is “new” since Year4 for RHIC experiments accustom to fast production turn around …

NON-STOP data production and data acquisition

Consequences on overall strategy

Needs & Principles

- **Cataloguing important**
  - Must be integrated with framework and tools
  - Catalogue MUST be
    - The central connection to datasets for users
    - Moving model from PFN to LFN to DataSets, cultural issue at first
  - STAR has a (federated) Catalog of its own brew…
- **Production cycles are long**
  - Does not leave room for mistakes
  - Planning, phase, convergence
  - **Data MUST be available ASAP to Tier1/Tier2 sites**
- **Access to data cannot be random but optimized at ALL levels**
  - Access to MSS is a nightmare when un-coordinated
    - Is access to “named” PFN still an option ?
    - Need for a data-access coordinator, SRM (??)

Data distribution
As immediately accessible as possible

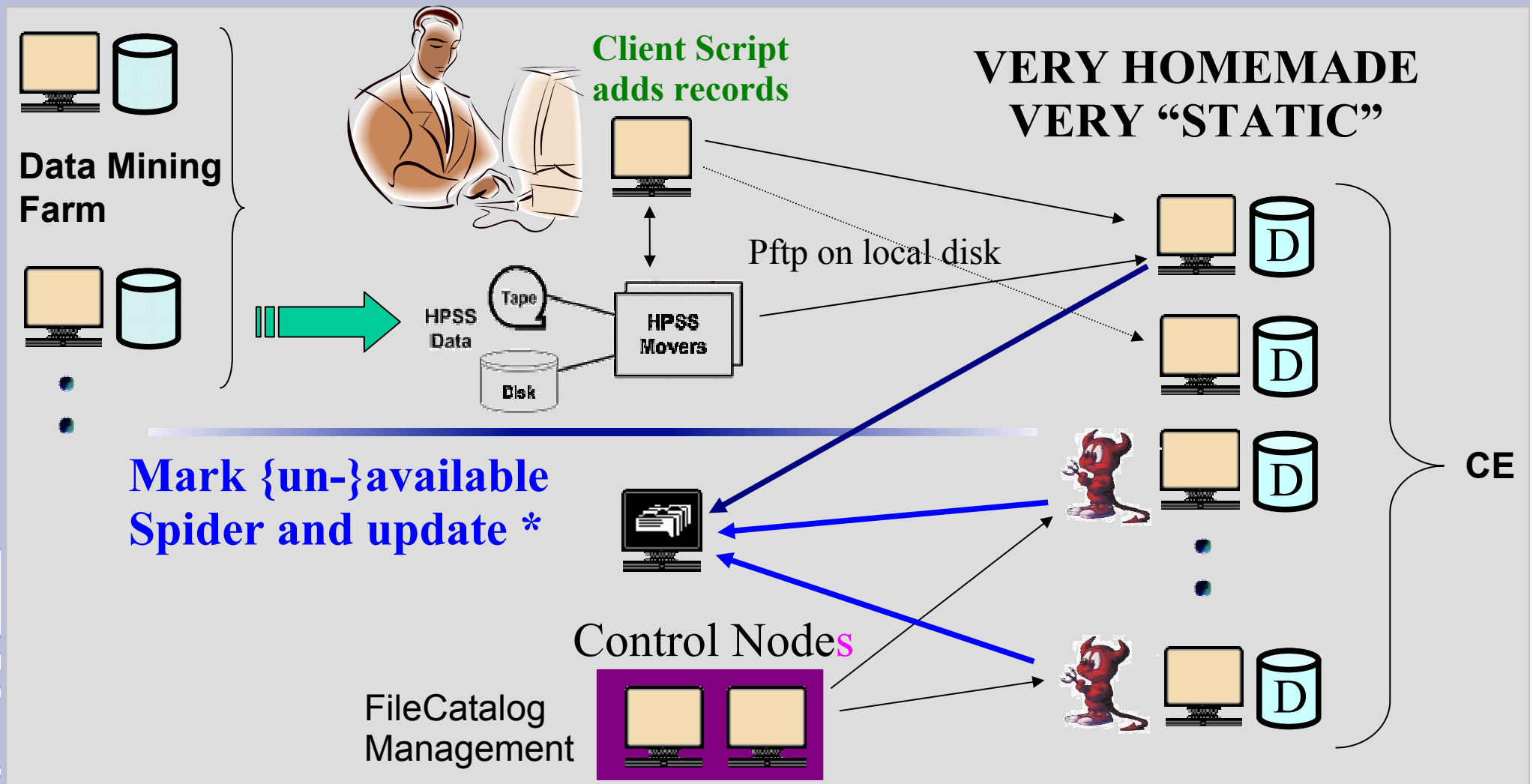
- ***Tier0* production**
  - ALL EVENT files get copied on MSS (HPSS) at the end of a production job
  - Strategy implies dataset IMMEDIATE replication
    - As soon as a file is registered, it becomes available for “distribution”
    - 2 Levels of data distributions – **Local** and **Global**

- **Local**
  - All analysis files (MuDST) are on disks
  - Ideally: One copy on centralized storage (NFS), one in MSS (HPSS)
  - Practically: Storage do not allow to have all files “live” on NFS
    - Notions of distributed disk – Cost effective solution

- **Global**
  - **Tier1 (LBNL) -- Tier2 sites (“private” resources for now)**

local/global relation through SE/MSS
strategy needs to be consistent
Grid STARTS from your backyard on …

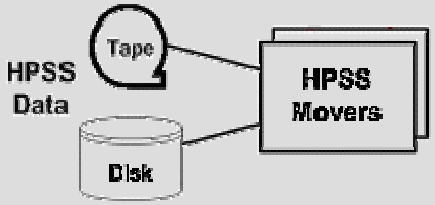
Distributed disks
SE attached to specific CE at a site



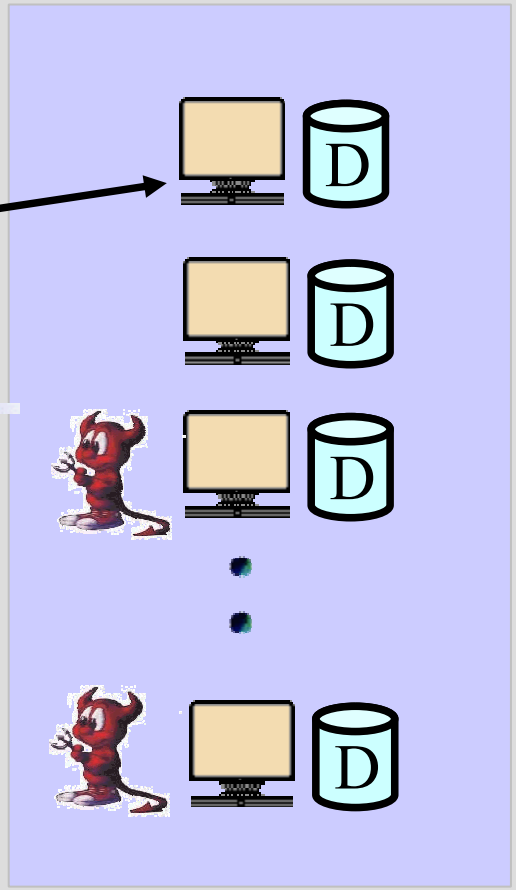
Distributed disks, possible model?

XROOTD

- load balancing + scalability
- a way to avoid LFN/PFN translation (Xrootd dynamically discovers PFN based on LFN to PFN mapping) ...



Seeking to replace this with XROOTD/SRM

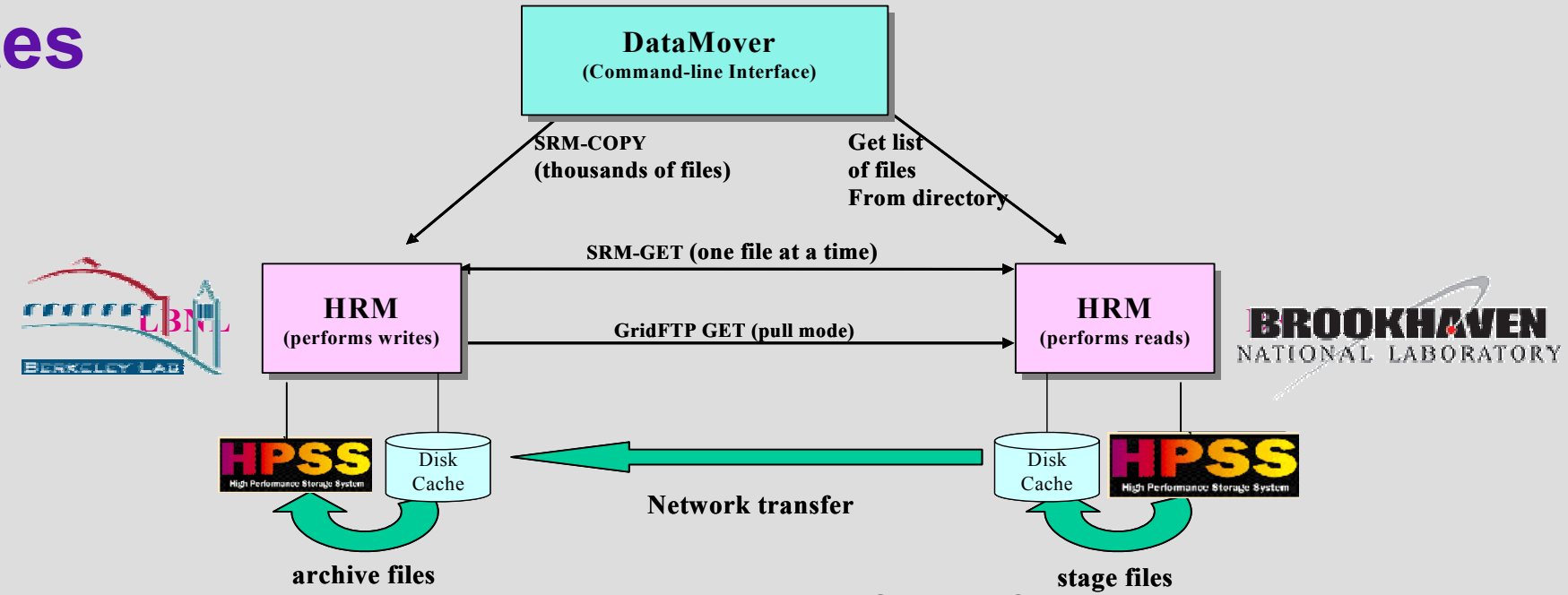


Coordinated access to SE/MSS STILL needed - “A” coordinator would cement access consistency by providing policies, control, …

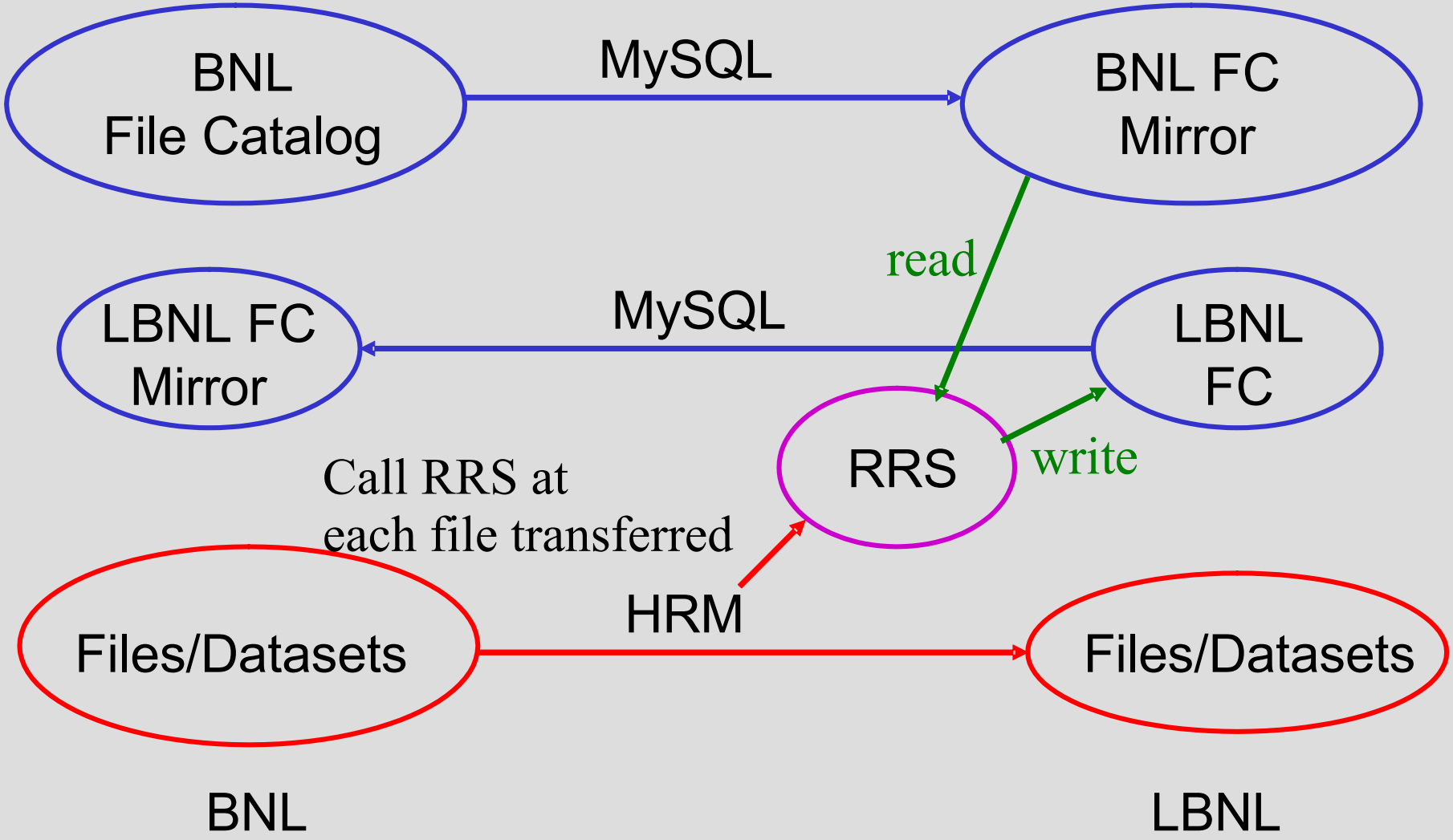
Could it be DataMover/SRM ???

Data transfer Off-site in STAR - SDM Data-Mover

- **STAR started with**
  - A Tier-0 site - all “raw” files are transformed into pass1 (DST), pass2 (MuDST) files
  - Tier-1 site - Receives all pass2 files, some “raw” and some pass1 files
- **STAR is working on replicating this to other sites**



Data transfer flow



Experience with - SRM/HRM/RRS

- **Extremely reliable**
  - Ronko's rotisserie feature "*Set it, and forget it !*"
  - Several 10k files transferred, multiple TB for days, no losses
  - Project was (IS) extremely useful, production usage in STAR
  - Data availability at remote site as it is produced
    - We need this NOW (resource constrained => distributed analysis and best use of both sites)
    - Faster analysis yield to better science sooner
    - Data safety
- *Since RRS (prototype in use ~ 1 year)*
  - 250k files, 25 TB transferred AND Cataloged
  - 100% reliability
  - Project deliverables on-time

Note on Grid

- For STAR, Grid computing is EVERY DAY Production used
  - Data transfer using SRM, RRS, ..
  - We run *simulation* production on the Grid (easy)
  - Resource reserved for DATA production (still done traditionally)
    - No real technical difficulties
    - Mostly fears related to un-coordinated access and massive transfers
  - Did not “dare” to touch user analysis
    - Chaotic in nature, requires more solid SE, accounting, quota, privilege, etc …

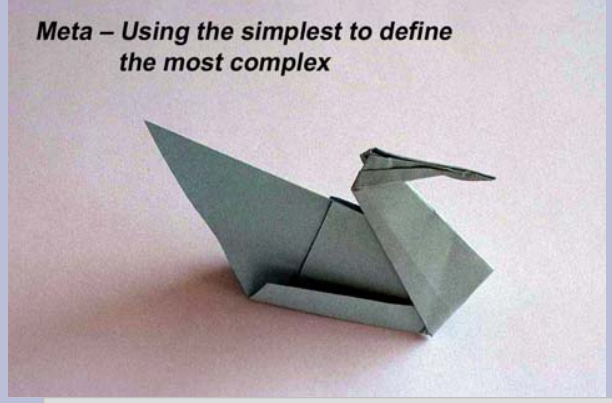
More on Grid

SUMS

The STAR Unified Meta-Scheduler, *A front end around evolving technologies for user analysis and data production*

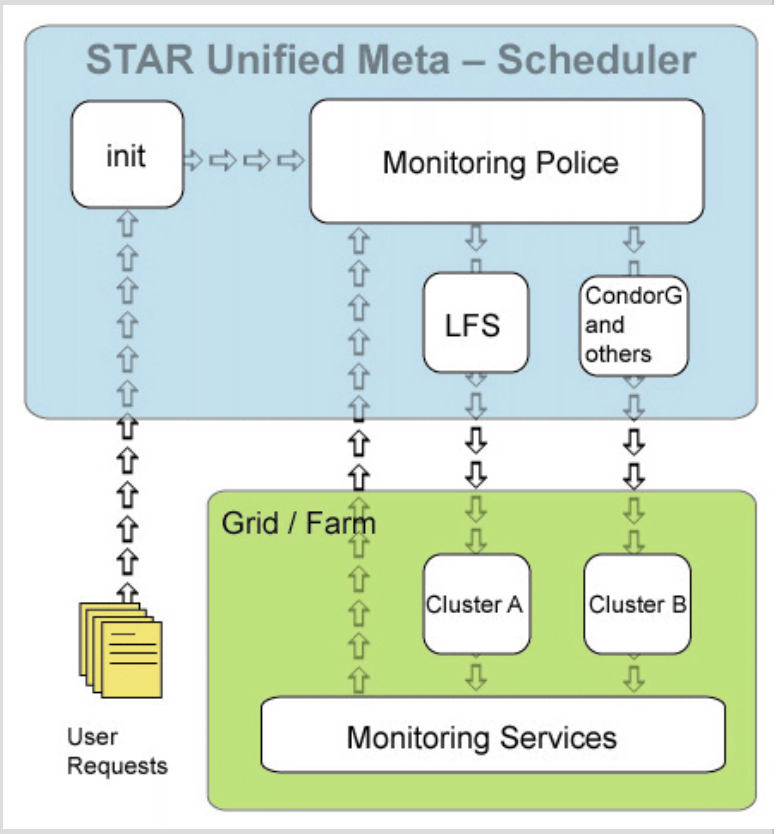
GridCollector

a framework addition for transparent access of event collection

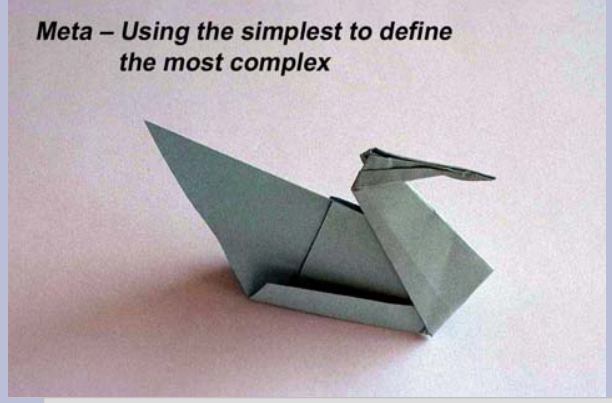


SUMS (basics)

- STAR Unified Meta-Scheduler
  - Gateway to user batch-mode analysis
  - User writes an abstract job description
  - Scheduler submits where files are, where CPU is, ...
  - Collects usage statistics
  - User DO NOT need to know about the RMS layer
- Dispatcher and Policy engines
  - DataSet driven - Full catalog implementation & Grid-aware
  - Used to run simulation on grid (RRS on the way)
    - **Seamless transition of users to Grid when stability satisfactory**
  - Throttles IO resources, avoid contentions, optimizes on CPU
  - Most advanced features include: self-adapt to site condition changes using ML modules



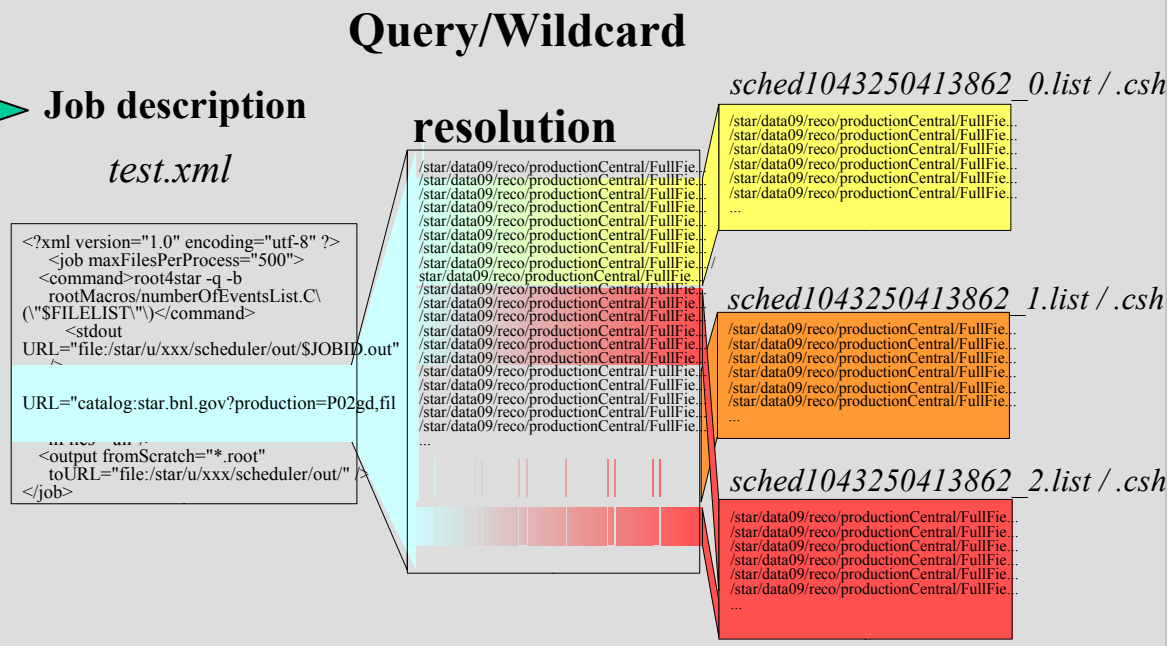
Makes heavy use of ML



SUMS input

From U-JDL to RDL

- SUMS: a way to unify diverse RMS
- An abstract way to describe jobs as input
  - Datasets, file lists or event catalogues lead to job splitting
  - A request is defined as a set or series of “operations”
    =
    A dataset could be subdivided in N operations

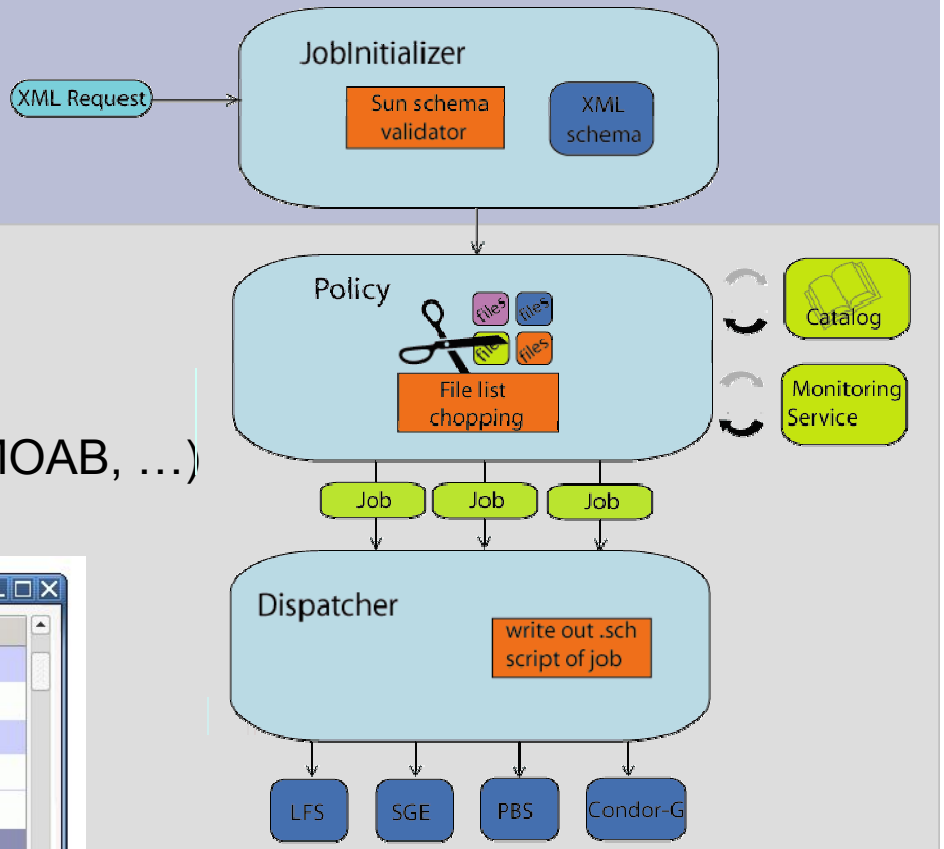


User Input … () … Policy …. dispatcher

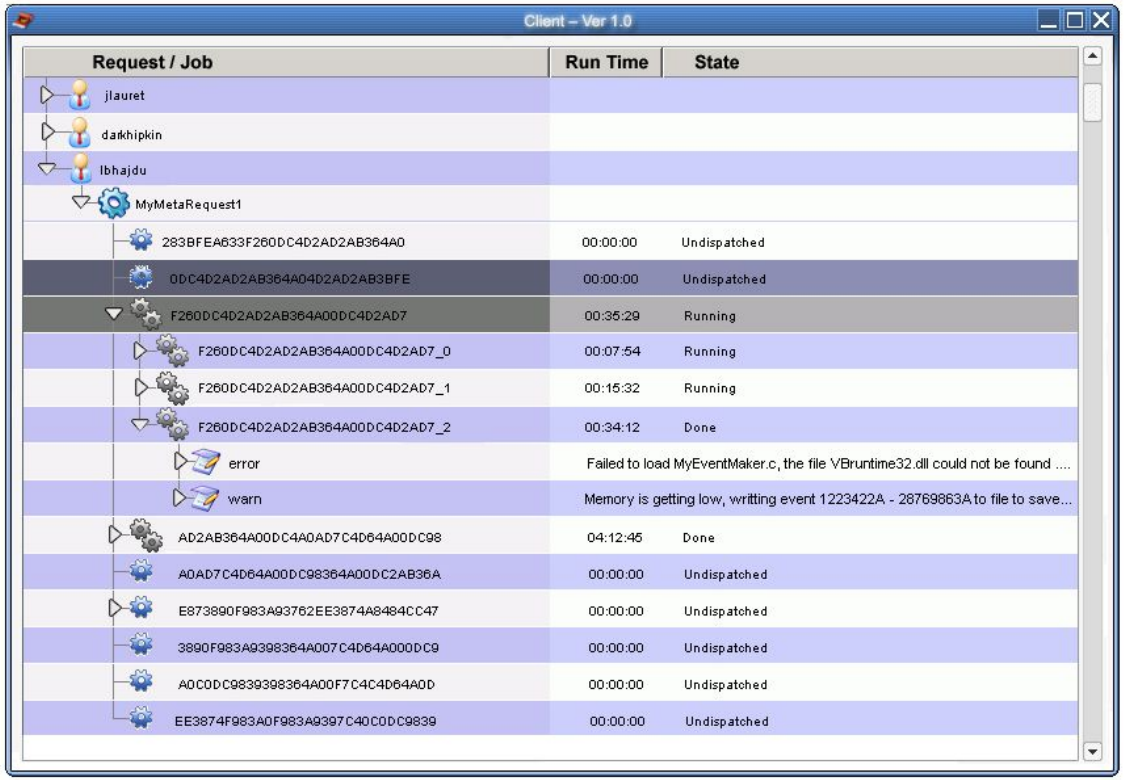
Extending proof of principle U-JDL to a feature reach Request Description Language (RDL)

- SBIR Phase I submitted to Phase II
- Supports workflow, multi-job, …
- Allows multiple datasets
- …

SUMS future



- **Multiple scheduler**
  - Will replace with submission WS
  - Could replace with other Meta-Scheduler (MOAB, …)



Job control and GUI

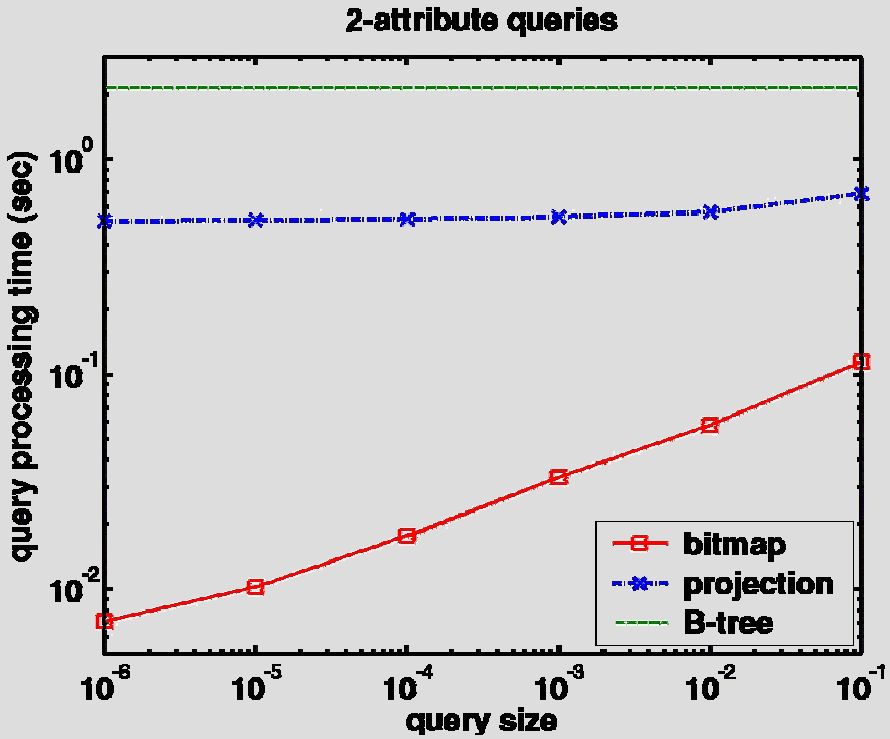
Mature enough (3 years) for spend time on GUI interface “appealing” application for any environment, easy(ier) to use

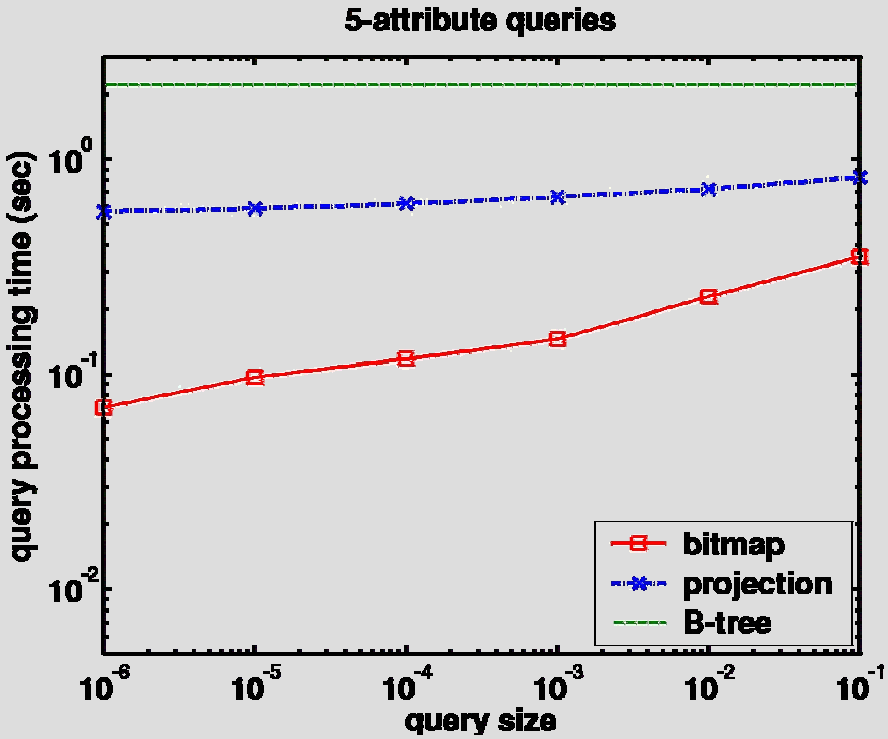
GridCollector

"Using an Event Catalog to Speed up User Analysis in Distributed Environment"

"tags" (bitmap index) based

- need to be define a-priori [production]
- Current version mix production tags AND FileCatalog information (derived from event tags)

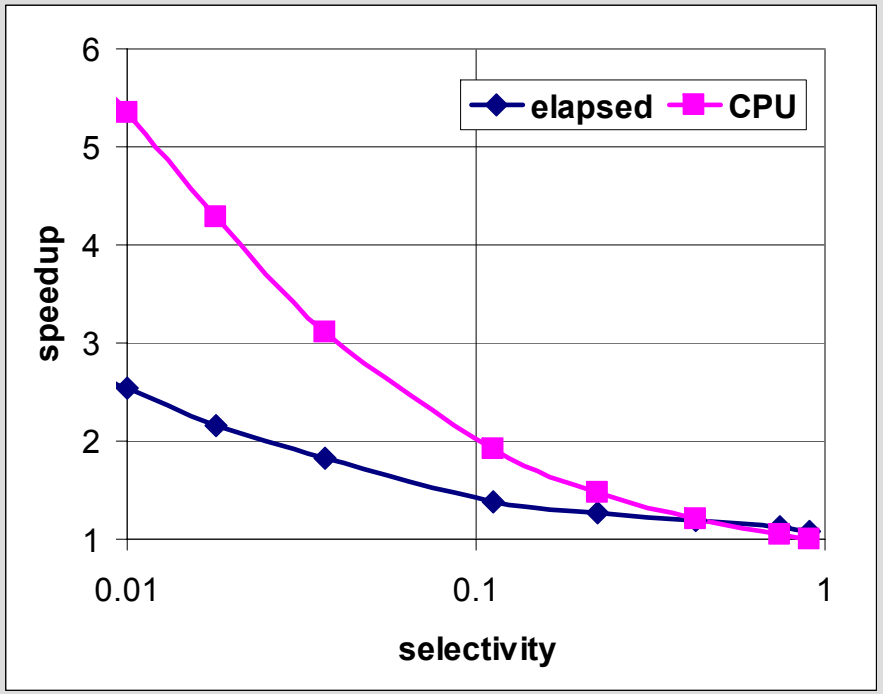




The compressed bitmap index is at least 10X faster than B-tree and 3X faster than the projection index

GridCollector

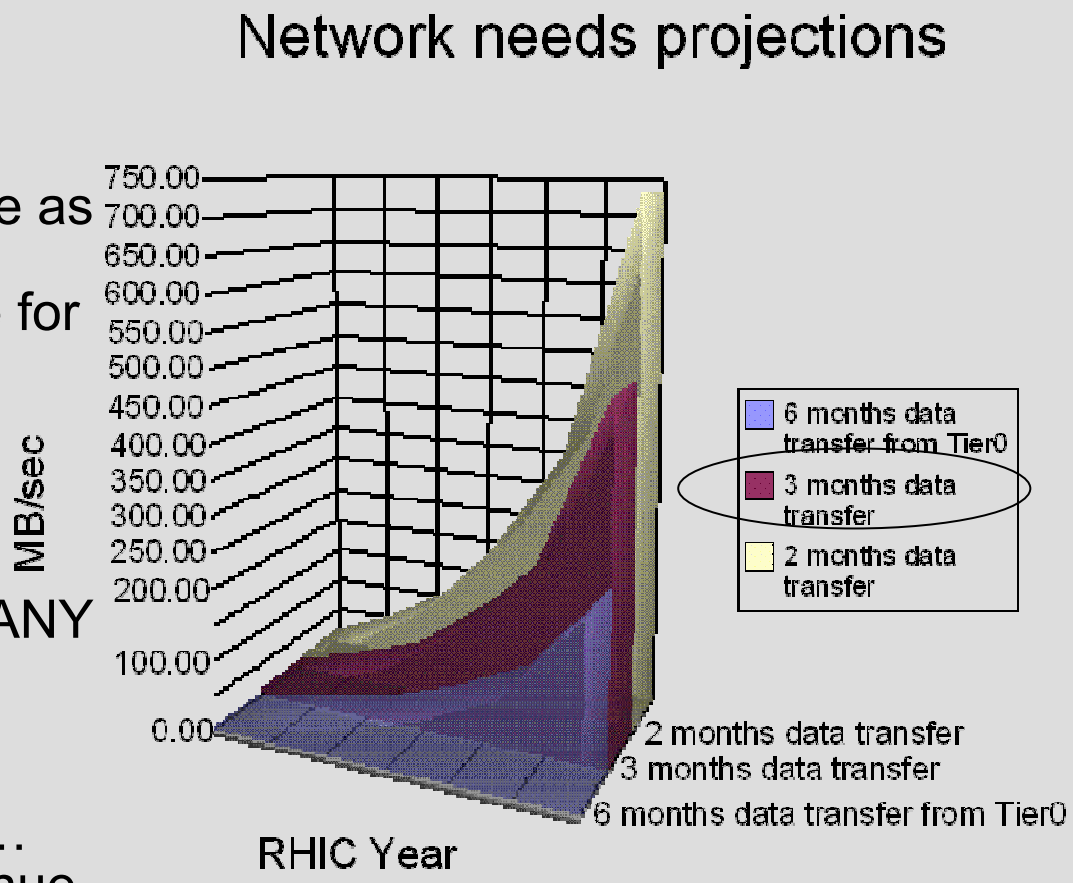
- **Usage in STAR**
  - Rest on now well tested and robust SRM (DRM+HRM) deployed in STAR anyhow
    - Immediate Access and managed SE
    - Files moved transparaentely by delegation to SRM service
  - Easier to maintain, prospects are enormous
    - "Smart" IO-related improvements and home-made formats no faster than using GridCollector (a priori)
      - **Physicists could get back to physics**
      - **And STAR technical personnel better off supporting GC**



- **It is a WORKING prototype of Grid interactive analysis framework**

Network needs in future

- **Grid is a production reality**
  - To support it, the projections are as follow
  - How does this picture looks like for user jobs support ??
- **Philosophy versus practical**
  - If network allows, send jobs to ANY CE and move data …
    - Minor issue of finding the "closest" available data, advanced reservation, etc …
  - If bandwidth do not allow, continue with placement ASAP …as we do now … and move jobs where files are (long lifetime data placement, re-use)



Moving from "dedicated" resources to "On Demand" → OpenScienceGrid

- **Have been using grid tools in production at sites with STAR software pre-installed.**
  - Success rate was 100% when Grid infrastructure was "up"
    - Only recommend to be careful with coordination local/global SE
  - Moving forward …
- **The two features to be achieved in the transition to OSG are**
  - Install necessary environment with jobs
    - Enables Computing On Demand
  - Integrate SRM/RRS into compute job workflow
    - Makes cataloging generated data seamless with compute work (not yet achieved for all STAR compute modes)