
Grid Components in STAR: Experience, Needs and Plans

A. Vaniachine

Lawrence Berkeley National Laboratory

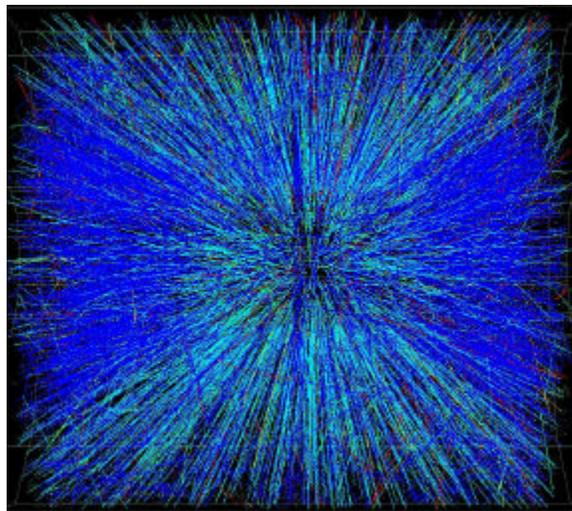
July 14, 2000

PPDG Workshop, Argonne



Outline

- STAR is taking data:
- Grid in analysis
- Grid for production?
- Plans



Decisions Taken

- Event split into components:
 - daq (raw data) or fzd (Geant3)
 - dst: legacy C-structures
 - event: same info in OO
 - tags: like an n-tuple
 - hist: for QA
 - ...
- ROOT-based Architecture
 - Persistency:
 - STL containers - structural and reference
 - schema evolution
 - Analysis:
 - “good-bye” PAW



Grid Components Deployed for Analysis

- tag database: to speed up user response
 - data in ROOT TTree files (n-tuple):
 - data description stored together with the data (solves the schema evolution problem)
 - comments preserved
 - 2-level hierarchy: 500 tags are in seven groups
 - split mode for structures (not objects: no need to load classes to analyze tags)
- indexed tags (Grand Challenge deployed)
 - efficient cache management
 - transparent user access to HPSS
 - in principle, to the raw data as well:
 - “the Dream” is coming true



Component-based Architectures

- **Grand Challenge:**
 - large size components
one component per developer
- **NOVA:**
 - smaller-size components
one components per task

Needed: small components, easily adaptable to experiments

e.g., "Is" for the HPSS



Grid Components for Production?

have to load all distributed nodes:
• **sustained data transfer methods**

have to use HPSS data storage:
• **efficient cache management**
• **transparent user access**

have to use low-cost commodity components:
• **latency tolerant algorithms**

\$\$\$
constraints



Plans for PPDG

- Deploy file transfer methods for the BNL-LBNL link
 - current rate: rcf ® pdsf = 100 GB/day
- Managed file caching for high-throughput computing
 - pdsf as a development test-bed
 - rcf as a target for deployment

