File and Metadata Management for BESIII Distributed Computing

C Nicholson, L Lin, Z Y Deng, W D Li, X M Zhang, Y H Zheng

The BESIII grid

The BESIII experiment at the Institute for High Energy Physics (IHEP), Beijing, studies physics in the τ-charm region around 3.7 GeV.

- 350 collaboration members
- 49 institutions (28 in China)
- World's largest sample of J/ψ events
- Current computing model highly centralized, with ~3500 CPU cores and 6PB of storage at IHEP used for reconstruction, simulation and analysis
- IHEP resources insufficient for current and future processing needs → move to grid / cloud / volunteer computing
- Small collaboration, limited manpower and expertise, low network connectivity between sites → BESIII grid needs to be easy to set up / maintain, intuitive for users, reliable and robust, minimize data transfer
- Ganga and DIRAC adopted as job management system
- **File and metadata management system required which is: scalable up to ~10 million files, ~100 concurrent users, with searchable file-level metadata, support for datasets, authentication and authorization, well integrated with job management.**

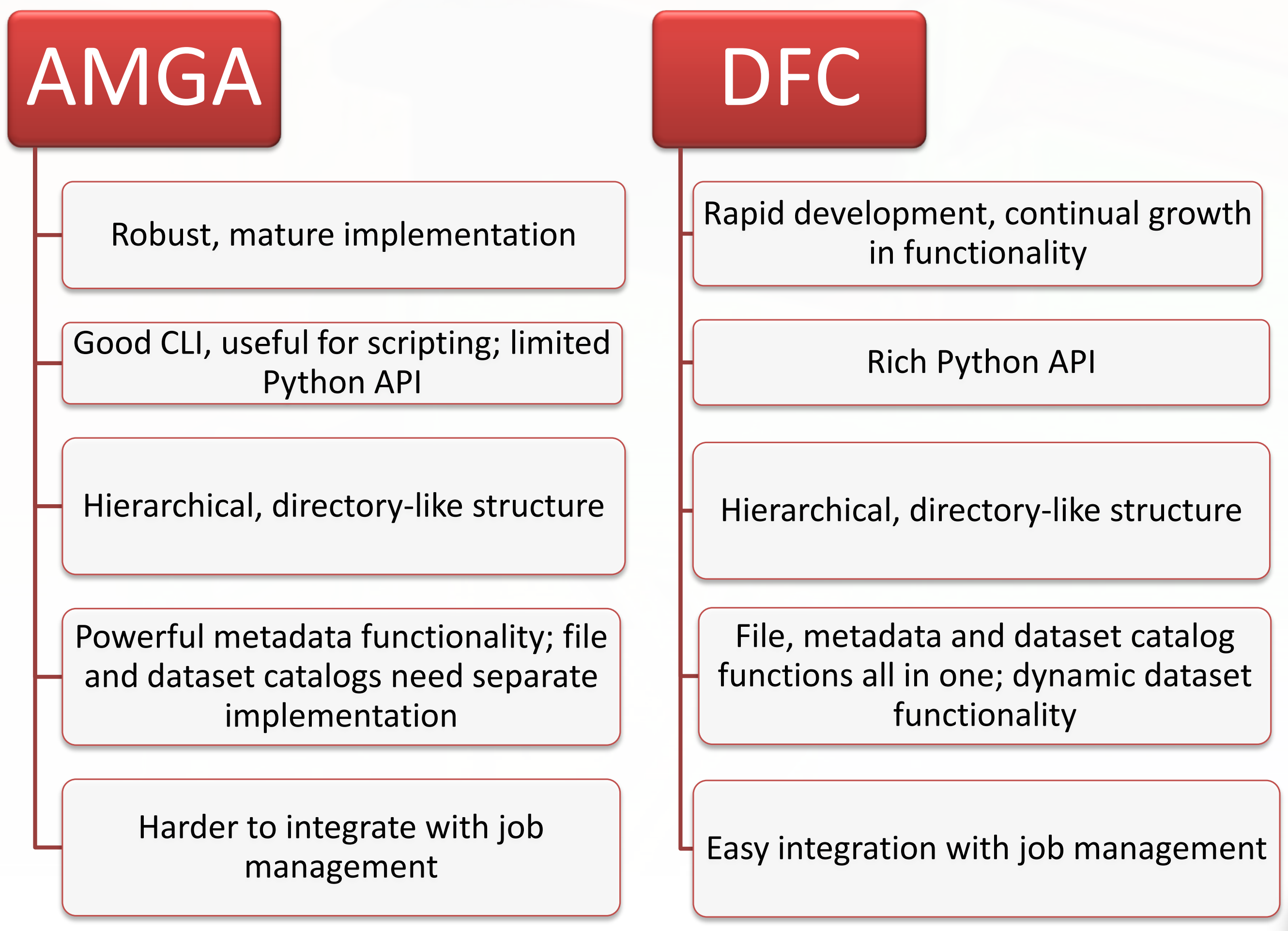
Metadata schema

Attribute	Data type	Description	File Metadata?	Dataset Metadata?
GUID	varchar(32)	Globally unique file ID	✓	
LFN	varchar(100)	Logical file name	✓	
Dataset ID	varchar(32)	Globally unique dataset ID		✓
Dataset name	varchar(100)	User-friendly dataset name		✓
Group ID	Int	Unique ID of physics group		✓
Data type	varchar(10)	BESIII data format (DST / RAW / TAG)	✓	✓
Event type	varchar(10)	BESIII event type	✓	✓
Resonance	varchar(10)	Data-taking resonance (J/ψ , ψ', etc)	✓	✓
Experiment no.	varchar(10)	Internal BESIII bookkeeping attribute	✓	✓
Software version	varchar(10)	Version of software used in reconstruction	✓	✓
runL	Int	Lowest run number in file / dataset	✓	✓
runH	Int	Highest run number in file / dataset	✓	✓
Stream ID	varchar(10)	Internal BESIII bookkeeping for MC data	✓	✓
File size	Int	Size of data file	✓	
Dataset size	int	Total size of dataset		✓
Number of events	int	Number of events in file / dataset	✓	✓
Status	int	Data is good / bad / other	✓	
Creation time	timestamp	Time of registration in catalog	✓	✓
Modification time	timestamp	Time of last modification	✓	✓
Description	varchar(100)	Extra notes / user-defined metadata	✓	✓

Lightweight metadata: max. ~10M files in 2 years
→ 3 GB file metadata

Functionality evaluation

Both catalogs have strengths and weaknesses, but DFC meets more of BESIII requirements: only one catalog needed to fulfil file, metadata and dataset catalog functions; already part of DIRAC and easily integrated with GANGA.



AMGA	DFC
Robust, mature implementation	Rapid development, continual growth in functionality
Good CLI, useful for scripting; limited Python API	Rich Python API
Hierarchical, directory-like structure	Hierarchical, directory-like structure
Powerful metadata functionality; file and dataset catalogs need separate implementation	File, metadata and dataset catalog functions all in one; dynamic dataset functionality
Harder to integrate with job management	Easy integration with job management

BADGER: BESIII Advanced Data ManaGER

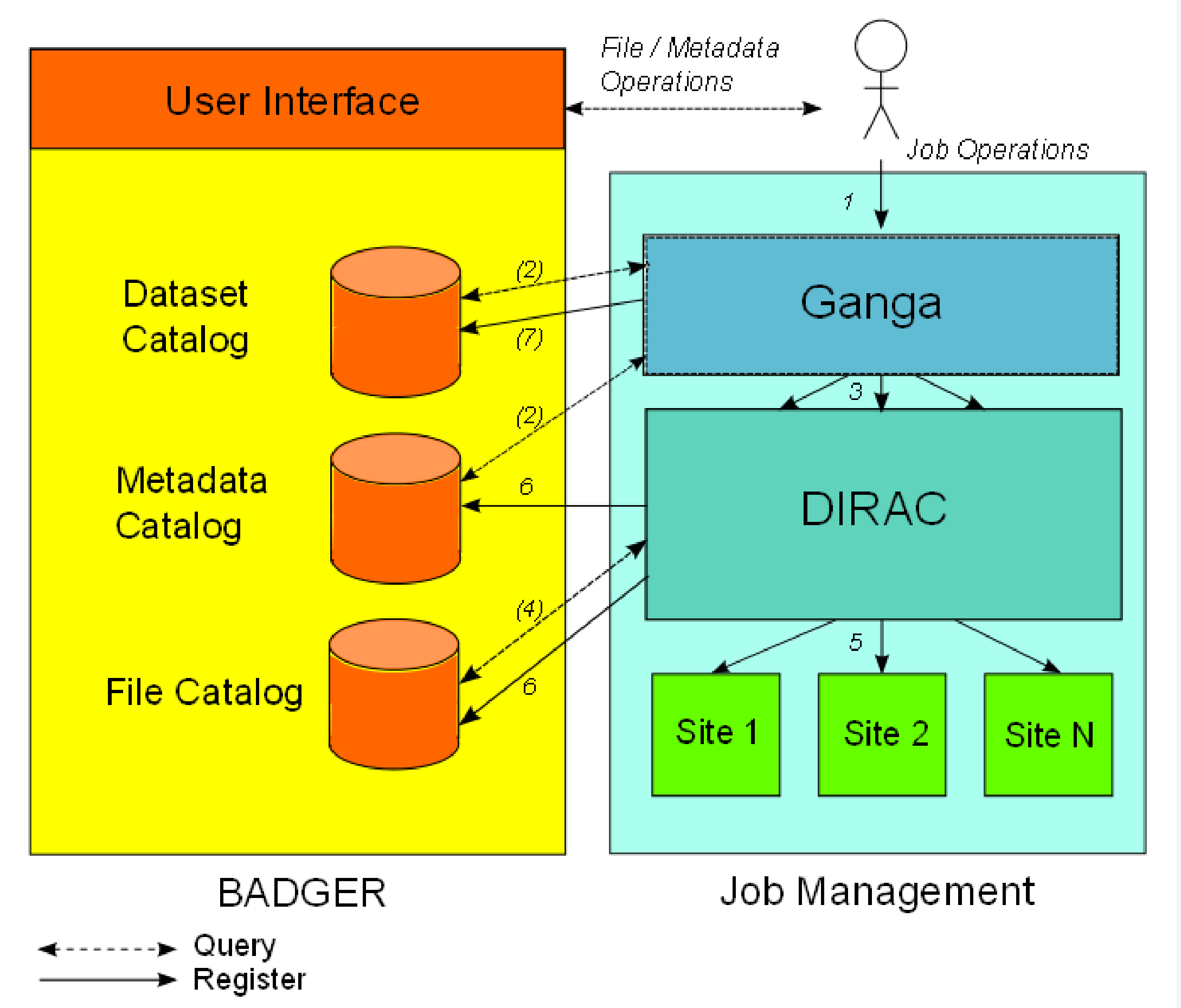
File and metadata management system must provide:

- File Catalog – map logical file names to physical file replicas
- Metadata Catalog – map files to associated metadata
- Dataset Catalog – map dataset names to file list
- User Interface
- Python API

Users should be able to interact with BADGER UI for data queries

Job management system should be able to query and register data from running jobs

Consider AMGA and DFC (DIRAC File Catalog) as possible catalog choices



Conceptual diagram of BADGER components and interaction with job management

Performance tests

Metadata schema implemented in AMGA and DFC

- 2 identical servers (HP Proliant DL180 2.40GHz 2x4-core CPU, 16GB RAM)
- MySQL backend

Optimised configuration:

- 8 DFC instances, max. 50 threads / instance
- AMGA max. 140 processes

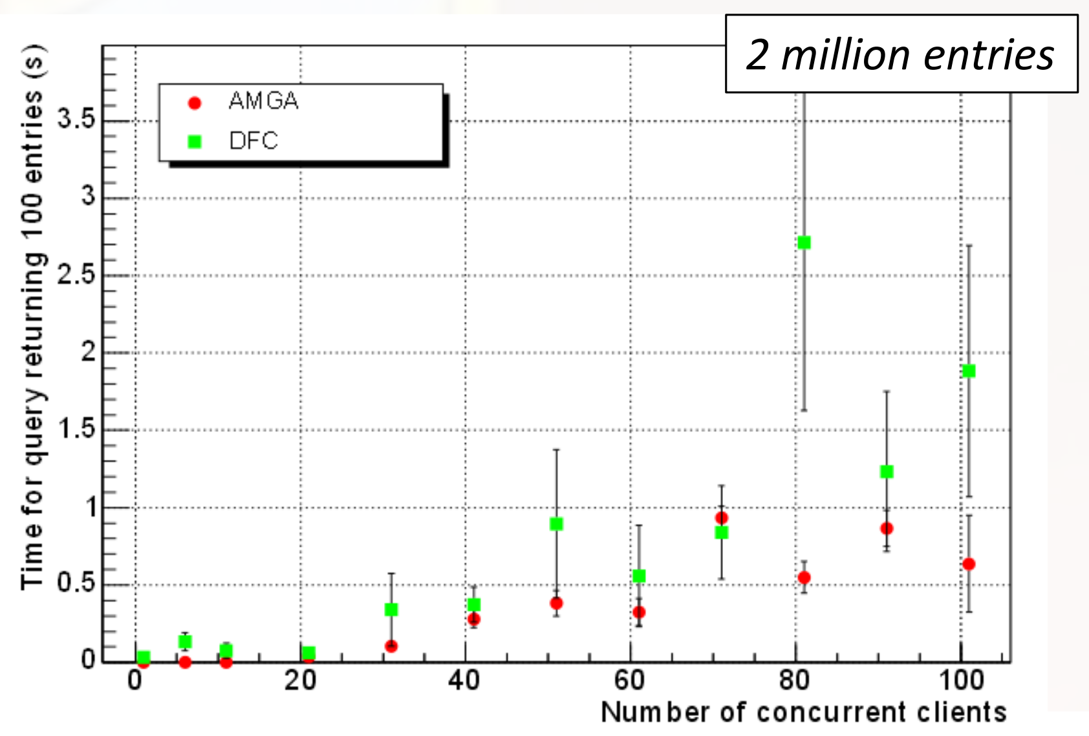
Current BESIII data set loaded (~200,000 files)

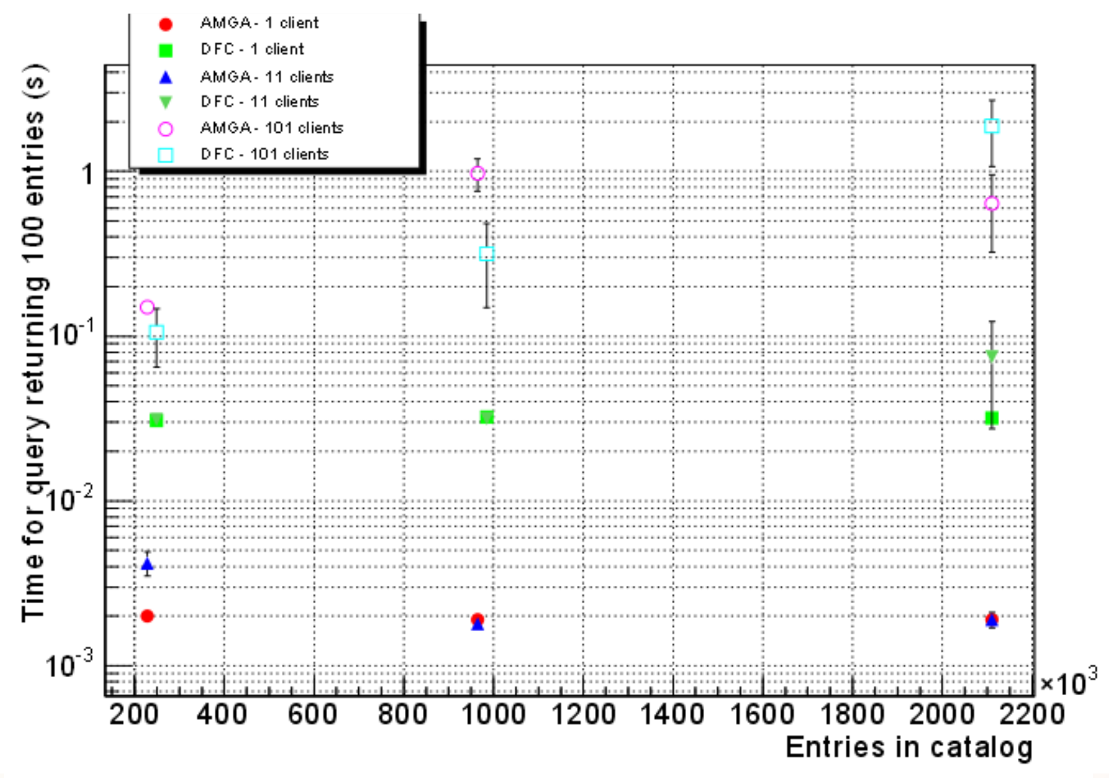
- Extended to 1M and 2M files for testing

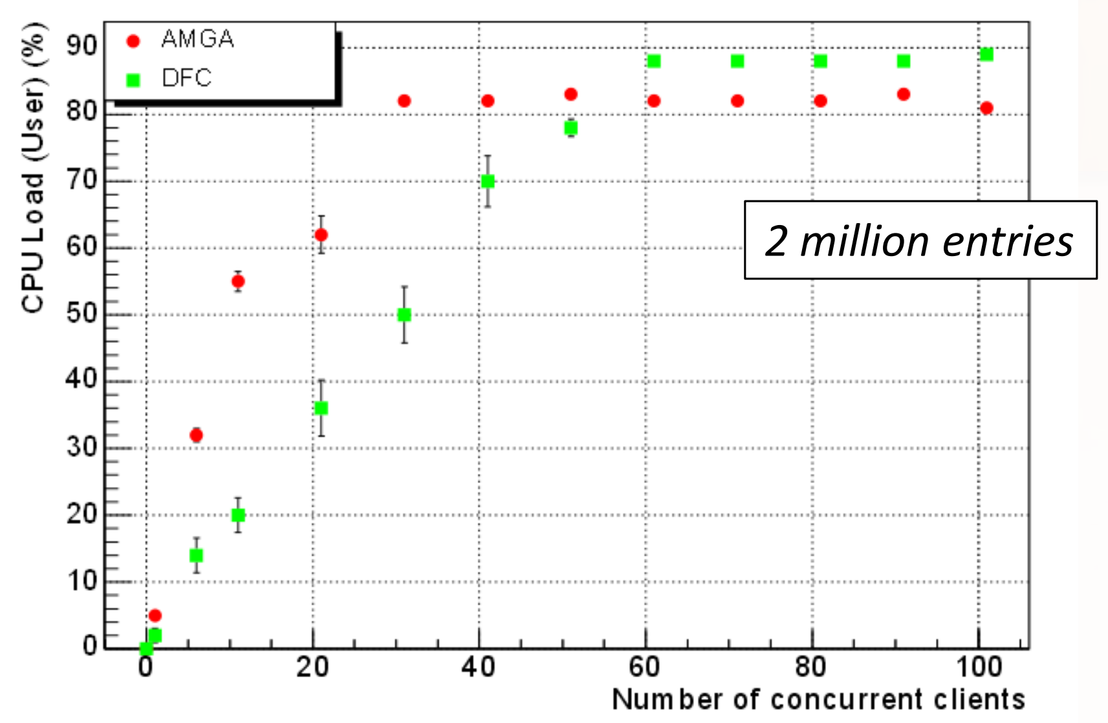
Test query times and CPU usage for increasing number of clients:

- With low number of clients, AMGA queries ~10x faster
- With high number of clients, query times approx. equal
- Catalog size important only for high number of clients → deploy load-balancing for production use
- DFC CPU usage rises more slowly

Both give acceptable performance







Current status and future plans

A prototype BADGER API has been written, based on DFC, and integrated with GANGA; simulation jobs can successfully register files and metadata in DFC on completion. Analysis jobs can query for files by metadata or by dataset.

Next steps include registration of new datasets, integration with file transfer tools, and development of a graphical User Interface. Deployment of a production service should follow, further optimising the server, database and DFC parameters as required to get the best performance.

中国科学院研究生院
GRADUATE UNIVERSITY OF CHINESE ACADEMY OF SCIENCES

中国科学院高能物理研究所
Institute of High Energy Physics Chinese Academy of Sciences