ARC Data Management and Future Developments by NDGF

David Cameron

EPF Seminar 2/11/07

NDGF

NORDIC DATAGRID FACILITY

Data in Jobs

- I will talk about data management associated to jobs
  - ARC does not manage large-scale data transfer outside the job framework
- When submitting a job to NorduGrid, input and/or output data can be specified
- The job has to access input data and write output data and there are two options
  - Submit the data with the job/copy the output back to the user
  - Have the data ‘on the Grid’
- In the second case the Grid middleware takes care of the data for you
- I will concentrate on this case here

Specifying Data

- Input data is specified in the description of the job in URLs in one of two ways
  - Giving the path to the file stored on a Grid storage element
    - eg "srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/file1“
    - The file is downloaded for your job from here
  - Giving the name of the file in an indexing service
    - eg “rls://atlasrls.nordugrid.org/file1”
    - An indexing service maps Logical File Names (LFNs) (file1 in this case) to physical replicas
    - The file is downloaded from one of the possible physical locations in the indexing service

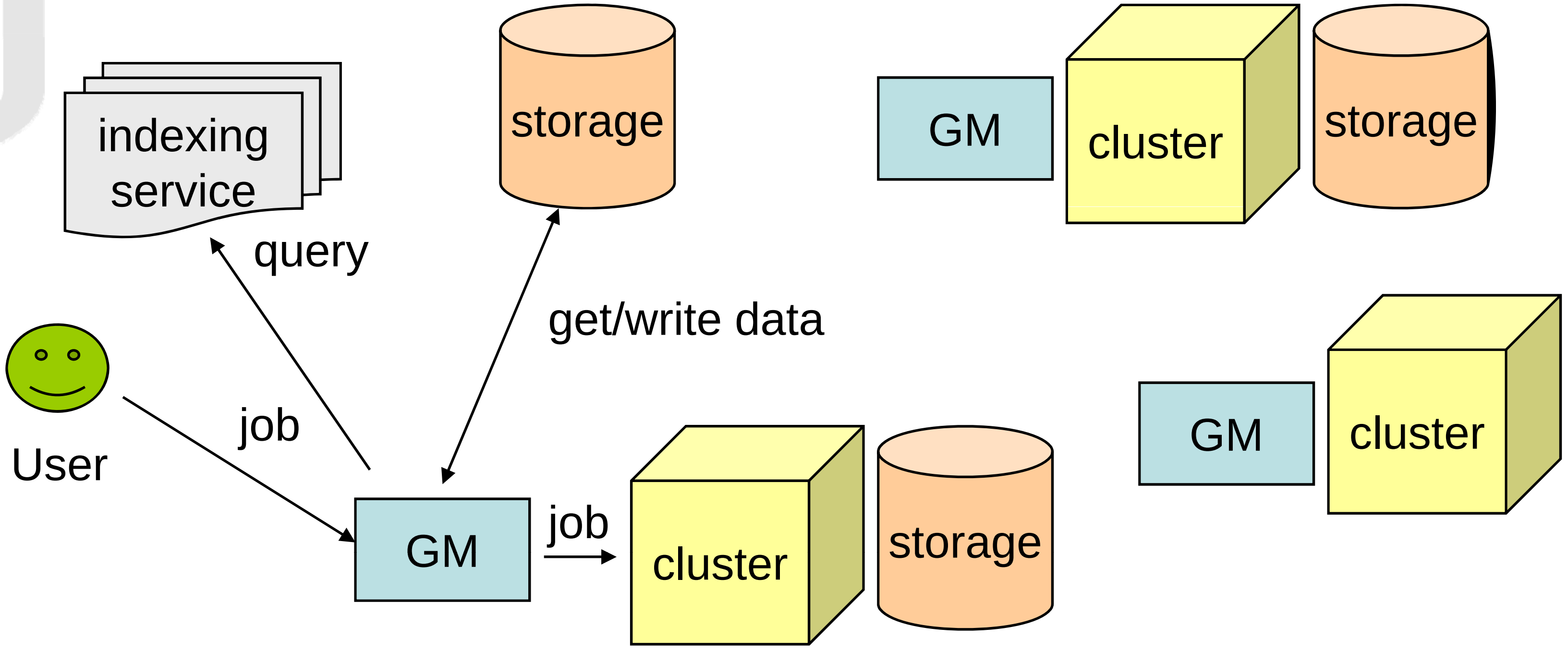
Specifying Output Data

- The same options are available for output data
  - Giving a physical location for the data
    - eg “srm://srm.ndgf.org/pnfs/ndgf.org/data/atlas/file2”
  - Giving an indexing service with a LFN
    - eg “rls://atlasrls.nordugrid.org/file2”
    - The physical destination for the file is chosen from a list within the indexing service

NDGF
Nordic DataGrid Facility

Data Movement

- So how do the input files get to where the job is running?
- This is the job of the ARC Grid Manager (GM)
  - A GM runs per cluster of Grid resources and takes care of data management for all Grid jobs submitted to that resource



NDGF
NORDIC DATAGRID FACILITY

Data Caching

- Each GM maintains a local cache, in order that input files do not have to be downloaded from Grid storage every time they are requested
- When the cache is full, the oldest unused files are deleted

- **Rule of ARC: No middleware on the worker node!**
  - Read this as everything Grid-related is done before and after the job enters the local batch system
  - Once the job is running it knows nothing about Grid

Planned Improvements

- It has been stated that there are no plans in KnowARC for development on the GM
  - So NDGF, which is supporting ‘classic ARC’, will implement some improved features
- The order I present these is not the order or priority for implementation!
  - But any thoughts you have on this are welcome

NDGF
Nordic DataGrid Facility

Planned Improvements

- Use of gLite File Transfer Service for data movement
  - This solves a lot of the problems further down this list
  - But requires that GM cache is accessible from outside
  - gsiftp endpoints work but are not supported in FTS
    - It is unlikely we could ask all sites to use SRM
  - Use this for large sites and have current solution as fallback and for small sites?
- Multiple hosts per GM for doing the data transfer
  - It seems we have reached a limit on data transfer which is the cable into the box hosting the GM
  - GM can spawn the data transfer processes on different nodes
  - Or have multiple GMs per site
  - Or split large sites into sub-sites
- Having the GM cache spread across multiple directories
  - This could solve the above problem, if the directories were on physically different hosts, mounted on the GM host

Planned Improvements

- Coordinate downloading large sets of large input files
  - So as not to have jobs blocked by other jobs with large input data sets
  - Possibly having separate queues/priorities for large/small input data sets
- Management of the GM cache - use for data aware scheduling
  - Cached files are registered in the indexing service but are not used for brokering during job submission
- Storing the DN associated with cached files
  - Each time a cached file is used we have to check access permissions of a copy on Grid storage
  - If we store the DN we don’t need to check this if same user requests the same file

NDGF
NORDIC DATAGRID FACILITY

Planned Improvements

- Dealing with error conditions eg Grid services down, retry policies
  - At the moment the retry policies are very simple
- Dealing with data on tape
  - If a requested file is stored on tape, the GM will simply block until the file is staged to disk
- For output files, be able to specify output storage preferences as an ordered list
  - This is from the ATLAS production system use case, where they want to have a primary storage and fallback storages in case of failure of the primary
- Direct I/O of files on Grid storage
  - Many use cases involve accessing a small piece of a large file
  - Here it makes more sense not to copy the whole file for the job
    - But remember – no middleware on the worker node!

NDGF
NORDIC DATAGRID FACILITY

Conclusions

- There is an endless list of possible improvements to be made
    - We now have to prioritise these
- The ‘problem’ is that the current system works very well
    - So we have to try not to introduce anything that interferes too much with this
    - So we plan to start with minimalistic changes ☺
    - We also have to keep an eye on KnowARC development