FERMILAB-CONF-09-743-CD

A New CDF Model for Data Movement Based on SRM

Manoj Kumar Jha
INFN Bologna, via Carlo Berti Pichat, Bologna, 40127 Italy
E-mail: jha@bo.infn.it

Gabriele Compostella, Donatella Lucchesi, Simone P. Griso
INFN Padova, via Marzolo 8, 35131 Italy
E-mail: compostella@pd.infn.it, donatella.lucchesi@pd.infn.it, simone.pagan@pd.infn.it

Doug Benjamin
Duke University, Durham, USA
E-mail: dbenjamin@fnal.gov

Abstract. Being a large international collaboration established well before the full development of the Grid as the main computing tool for High Energy Physics, CDF has recently changed and improved its computing model, decentralizing some parts of it in order to be able to exploit the rising number of distributed resources available nowadays. Despite those efforts, while the large majority of CDF Monte Carlo production has moved to the Grid, data processing is still mainly performed in dedicated farms hosted at FNAL, requiring a centralized management of data and Monte Carlo samples needed for physics analysis. This rises the question on how to manage the transfer of produced Monte Carlo samples from remote Grid sites to FNAL in an efficient way; up to now CDF has relied on a non scalable centralized solution based on dedicated data servers accessed through rcp protocol, which has proven to be unsatisfactory. A new data transfer model has been designed that uses SRMs as local caches for remote Monte Carlo production sites, interfaces them with SAM, the experiment data catalog, and finally realizes the file movement exploiting the features provided by the data catalog transfer layer. We describe here the model and its integration within the current CDF computing architecture.

1. Introduction

The CDF experiment has generated more than 5 fb^{-1} of raw data. Monte Carlo(MC) data are needed for detector understanding and physics analysis. It is not feasible to produce these MC data on-site due to limitation on computing resources. The CDF remote sites or Grid Tier1 and Tier2 can be utilized for producing MC data. From our past experience, we learnt that one of the most important limitation in the heavy usage of off site resources is that the Worker Nodes(WN) were sitting idle for several hours just because another WN was transferring data to Storage Element(SE) at Fermilab. This situation leads to inefficient uses of computing resources

and sometimes to the loss of the output from the worker nodes. In the present CDF Analysis Framework(CAF), there is no mechanism to deal with the sudden arrival of MC job's output from different remote grid sites during same interval of time in the SE at Fermilab. We found that this causes overloading of available file-servers and leads to failure of data handling part of MC production. Moreover, there don't exists catalog of produced MC files on individual user's basis. The user maintains themselves the records of produced MC files. Hence, a robust framework is needed for produced MC file management and its transportation from remote compute sites to SE at Fermilab.

The long wait time on WN after job completion arises due to limitation on available bandwidth between WN and SE at destination sites. It can be overcome by first storing the MC output (from WN) temporarily in Storage Element (SE) closer to WN and, then transfer sequentially to storage element at destination site. Figure 1 illustrates the movement of data from WN to SE at destination site for different CDF Analysis Farms (CAF). Each headnode represents different CAF and there exists a SE closer to WN. The term *closer* means that the bandwidth between WN and SE is comparatively good or they are within the same Local Area Network (LAN). When the user's job end on WN, the output data is first temporarily stored in SE closer to WN and then moved it to SE at destination site. In this case, the waiting time of output data on WN has been considerably reduced due to the large bandwidth available between WN and its SE. After transferring output data from WN to SE, this WN can be assigned to other user's job. In this way the CPU resources available per unit time have been increased with respect to earlier case when the output data was being directly transfered from WN to SE at remote destination site.

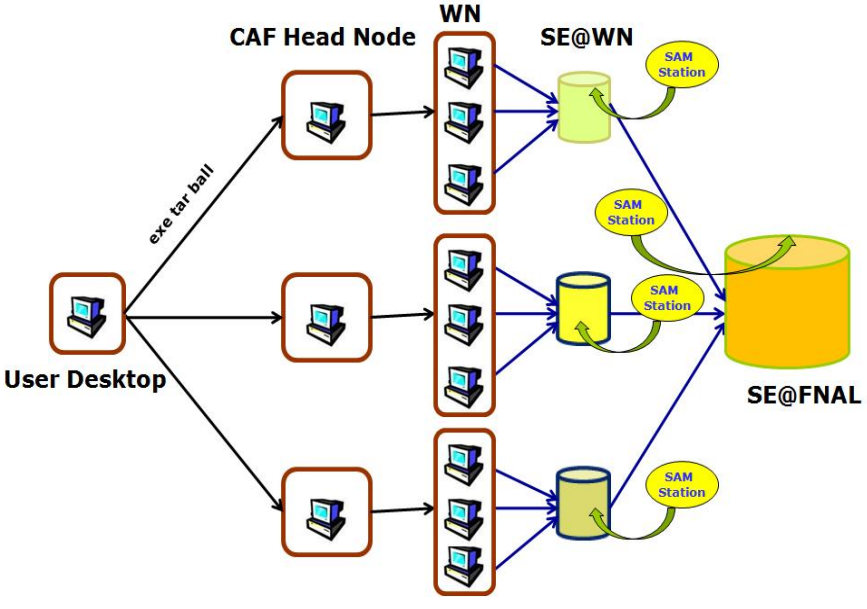

Figure 1. A prototype model for transfer of MC data from WN to SE at remote destination sites.

Figure 2 represent the proposed model for transfer of MC data for a single CAF. In the rest of document, we will be talking about the model for a single CAF. The entities in the rectangular blocks constitutes different part of the model. Each SE acts as cache of a SAM station. We are using the features of SAM-SRM interface [2] for data movement, replication and caching. A short introduction on SAM-SRM interface is being given in Section 2. Section 3 gives a brief

summary of the present CDF Analysis Framework. Section 4 gives the detailed description of the proposed model. The evaluation of the data transfer model is presented in section 5 while its integration in the present CDF analysis framework is layout in the subsection 5.1 and 5.3.

2. SAM-SRM Interface

The SAM-SRM interface apply SAM data handling policies over the generic storage system managed by SRM protocol. The premise of the concept is similarity between SAM managed disk storage and SRM managed storage in terms of many supported operations. In particular such operations as transferring file, removing file, building directory listing, and retrieving file meta-data are common for POSIX file systems and are also supported by SRMs.

SAM disks are used as root locations to place data files. Disks also have size that limits disk location space usage by SAM. The Logical disks and SAM file replacement policy constitute SAM cache that brokers resources between storage elements and storage element clients. Naturally, the same concept is reused to point SAM to an SRM location where files should placed or removed from. More information on SAM to SRM mapping can be found in specification document [2]. SAM-SRM interface is expected to support following properties.

- User should be able to pick SRM location where he/she expect SAM to place and access data.
- User expects SAM to place data to a chosen location on demand and according to polices set by the SAM station configuration (pre-fetching, fair share, replication algorithms and etc.) and SRM. Data placement on demand assumes that user does not have explicit knowledge about data source, data status at the storage or overall topology of underlying storage deployment. SAM, therefore, is responsible to ensure that data is moved and made accessible to a user in a transparent manner.
- User expects SAM to translate logical location into one of the following strings that should comply with access protocols: dcap , gsiftp. These strings should be made available to application such that latter can access actual data bits. The exact choice of the protocol is either static or dynamic.
- User expects SAM to register data in the replica catalog in order to reflect successful placement attempts. Registered locations should provide enough information to build a successful query to SRM by registering or external stations.
- User expects SAM to synchronize on demand replica locations with actual status of data in SRM storage.
- A minimal policy SAM should support is the policy of replacement based on the storage size and storage type (permanent/durable).

3. CDF Analysis Framework

The CDF Analysis Framework(CAF) [5] is a large network of computing farms dispersed over a large network of resources designed to allow users to run multiple instances of the same CPU-intensive jobs. The heart of the CAF resides at Fermilab. However, there are several other CAFs around the world(DCAF), including, but not limited to, SDSCCAF in San Diego, MITCAF at MIT, CNAF at Bologna and many other facilities. Submission, monitoring,and output are authenticated by user's FNAL.GOV kerberos ticket, allowing users to access these facilities from any kerberized machine in the world. The CAF is based on the Condor system. In order to manage security more effectively, the CAF has a single node that centralizes all of the user's command and control functions.

4. Proposed Model

The proposed model relies on integration of SAM with SRM. We used SAM because it is the default data handling framework of the CDF. SAM offers tools to describe storage deployments as well as implements policies of data movement, replication and caching. The reason for using SRM is to avoid unnecessary complications which may arise from different flavors of SRM managed SE at different grid access sites. SAM-managed storage elements are organized as configured station consumption sites with limited by size on storage system. SE and SAM file replacement policy constitute SAM cache that brokers resources between storage elements and storage element clients. To efficiently manage data flow, SAM cache is operated with pre-set assumptions on costs of data access for its storage elements. In particular, cost of access is assumed to be uniform across the SAM cache. Moreover, the data can neither appear nor disappear without explicit request from SAM. Data management and its replications are inherent features of the proposed model.

Figure 2 shows a prototype model for movement of MC data from WNs to the SE at the remote destination site. The destination SE in our case is at Fermilab. The destination site can be any one of the remote DCAFs so that job's output at other DCAF can be transfer there. Following are the components of the proposed model.

- SAM managed SE closer to WN. The space on this SE is of volatile nature.
- SAM managed SE at the destination site. SE are managed by SRMs, the physical location of SAM stations don't matter in this model.
- Users would have to indicate the name of nearest SAM station for getting the MC job's output.
- Model proposes separate datasets for output and log files. Output files are of the order of several Mega bytes, while log files are in the range of few Kilo bytes. Our concern is to minimize the number of small file size request which arises from a user's job accessing the MC data. Number of small file size request can be avoided by having log and output datasets separately.
- Each dataset corresponding to a JID has unique name.

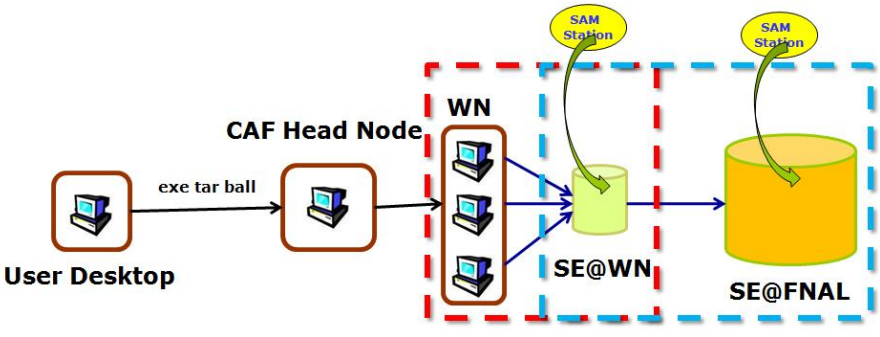

Figure 2. Proposed model for transfer of MC data for a single CAF.

5. Test Framework

A test framework as shown in Figure 3 has been setup for evaluating the performance of the proposed model. CNAF Tier-I facility (CNAFCAF) has been used as CE . SE closer to WN is dCache managed SRM at Gridka. The Gridka SRM can handle 10 channels per request and it

File Type	File Size
A	10 MB < size < 100 MB
B	100 MB < size < 1 GB
C	1 GB < size < 3 GB
D	> 3GB

Table 1. Categorization of dummy files on the basis of its size.

Dataset Type	Dataset Size
A	10 MB < size < 1 GB
B	1 GB < size < 3 GB
C	3 GB < size < 10 GB
D	> 10 GB

Table 2. Categorization of datasets on the basis of its size.

acts as cache of a SAM station "cdf-cnafTest". The destination SE is dCache managed SRM at University of California, San Diego(UCSD). The UCSD SRM can handle 50 channels per request and attached to SAM station "canto-test".

A cornjob (replaced by user) submits a job of variable number of segments every 10 minutes at CNAFCAF. The maximum segments per job is 5. Each segments creates a dummy files of random sizes which vary between 10 Mega bytes to 5 Giga bytes. Figure 4 shows the distribution of submitted job sections. Numbers inside the circle represent total number of section being submitted while number outside the circle represents number of sections being submitted in a single job. For example, a job with 1 section has been submitted 56 times, with 2 sections 64 times and so on. When the job finishes, the proposed model temporarily collects the job's output in the SE closer to WN (SRM at Gridka). A cron process is running at station "cdf-cnafTest" for creation of datasets from the files in SRM at Gridka. Another cron process is running on station "canto-test" for transfer of datasets between two stations.

For presenting the results, the dummy files and datasets created by the cron processes have been divided in different categories as shown in Table 1 and Table 2. Subsection 5.1 gives brief description of processes which happen after job completion on WN. Subsection 5.2 list the various steps needed to create datasets from job's output. The procedure for getting datastes in the destination SE has been described in subsection 5.3.

5.1. Description: On the WN

User submits the MC job using the present CDF Analysis Framework(CAF). All the output and log files produced by the user job's are unique. One option is to add CAF name, JID and section number in the user's provided file name.

If SAM station environment is available on the WN, output and log files are declared to SAM database from the WN. Otherwise, files declaration happens at the dataset creation time. Minimum information for creation of metadata is created on demand. Option also exists for a user to provide specific dimensions in the metadata on the basis of which they can query the datasets. For ex., a user would like to have all the list of datasets which corresponds to a certain userid or date of creation or physics group or physics channel of their interest. In this way, the model keeps track of all the produced MC files on the CDF CAFs. Book keeping of the produced MC files is the inherent feature of the MC data transfer model. Figure **??** depicts the file declaration time in SAM database from WN. The time needed to declare datasets in SAM

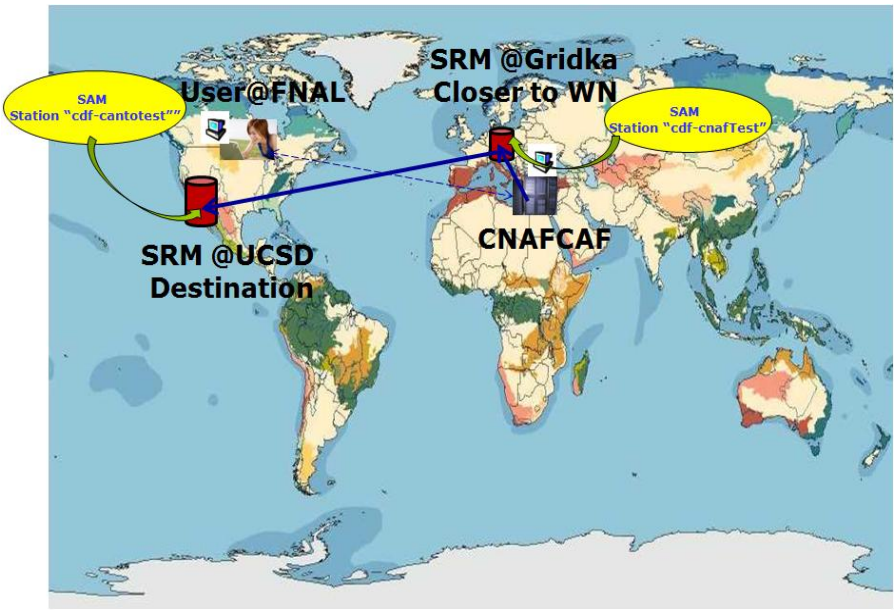

Figure 3. Test framework for evaluating the performance on MC data transfer model.

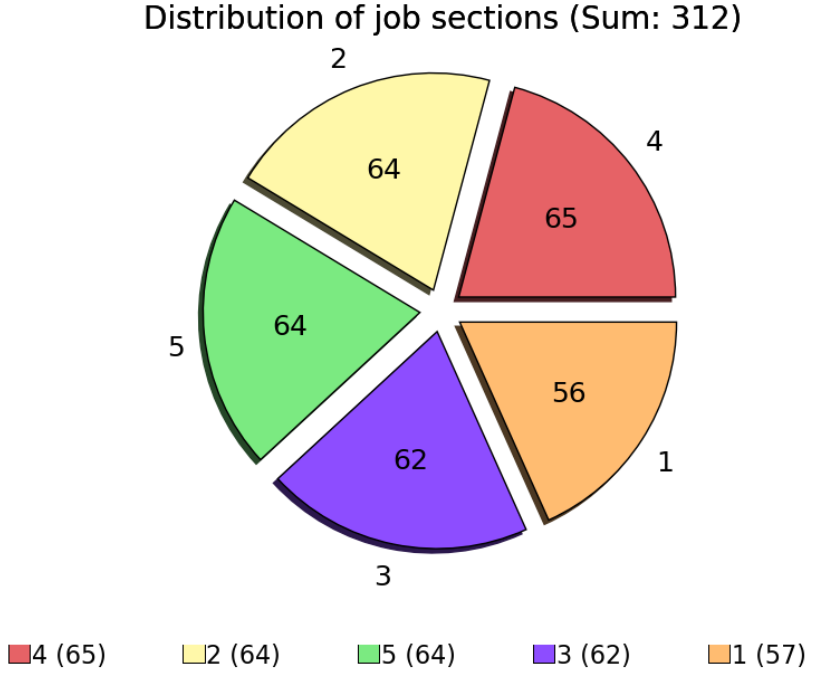

Figure 4. Distribution of submitted number of job sections.

is constant for all file types.

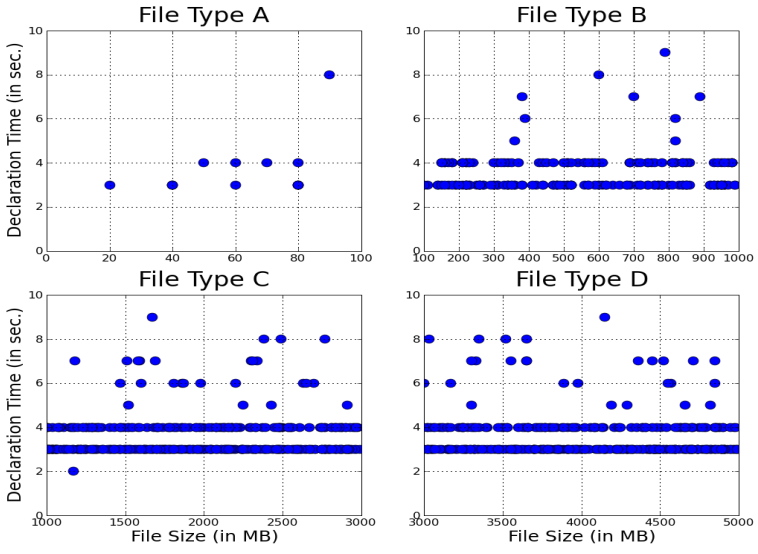

Figure 5. Time needed to declare dataset from output files in SAM.

A pre define storage element SURL "CAF_SE_SURL" managed by SRM will be known to each CAF. This predefined SE can also be opportunistic one. Proposed model has the ability to manage the space on SE which has been allotted through fixed lifetime space tokens. User's proxy available from the CAF head node has been used for grid authentication purpose.

At the completion of user's job, the model creates a folder "CAF_SE_SURL/JID". All the output and log files get transfer to the "CAF_SE_SURL/JID" folder using the available SRM tool on the WN. One can use either the Fermi SRM or LCG tools for transferring of files from WN to SE managed by SRM. Figure 6 shows the transfer rate of MC job output from CNAFCAF to SE at Gridka. Transfer rate increases with file size and then saturates for a file size greater than 3 GB. Wait time for job's output on WN has been considerably reduced from the earlier case when it was directly transfered to the remote SE.

5.2. Description: Near the SE closer to WN

The predefined "CAF_SE_SURL" acts as cache of a SAM station "cdf-cnafTest". Service certificate of station is being used for grid authentication. A cronjob runs on the station's pc which check:

- Existence of new JID folder (say "123456") in the "CAF_SE_SURL/123456" .
- Check status of JID "123456" from the CNAFCAF head node.
- If finished, it will go to the next step. Otherwise go to the next "CAF_SE_SURL/JID" folder.
- Create a list of files for output and log files. Now the following steps will be executed individually for each output and log files list.
  - Create a file which stores the list of file names. This is needed in case of SRM operation failure.
  - Check whether the file has been declared in SAM or not. If not, declare the file in SAM using the minimum metadata information.

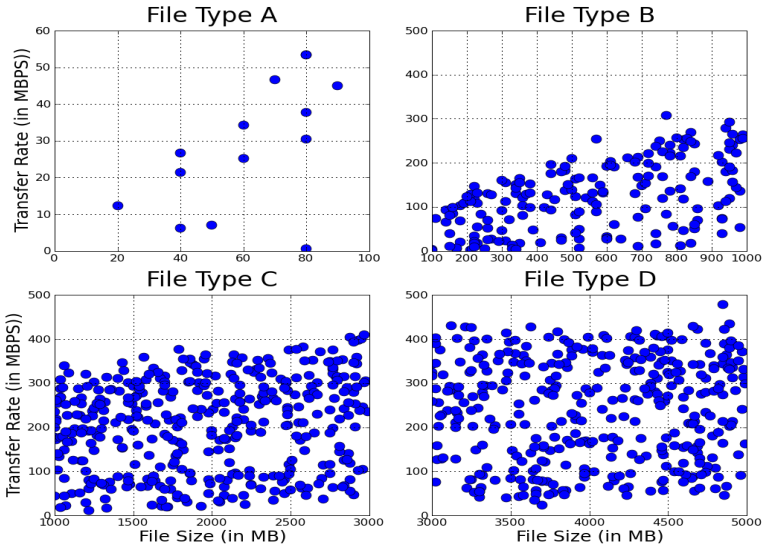

Figure 6. Transfer rate of MC jobs output from WN to its closer SE.

  - Move the files to SAM upload folder "CAF_SE_SURL/upload".
  - Create a dataset from list of files stored on the local disk of PC. Dataset name should be unique. One option is to use the JID and CAF name in the dataset name.
  - In the present case, dataset for output and log files are "FILE_CNAFCAF_123456" and "LOG_CNAFCAF_123456".
  - Make aware the created dataset to SAM. SAM will create a space for the created datasets. Figure 7 shows the dataset declaration time for each type of datasets. If sufficient space don't exists, older files will be deleted from the station cache. It is necessary to have sufficient cache space of station so that new dataset gets transfer to destination station cache before its deletion from the CAF station cache.
  - Update log files.
- Remove the JID folder 123456.
- Inform user for the availability of the job's JID "123456" output at their desired station.
- Repeat the above steps for next JID folder.

5.3. Description: SE at destination site

Due to non availability of SRM at Fermilab for CDF, the destination site is UCSD in the present case. UCSD SE acts as cache of destination SAM station "canto-test". A process runs on "canto-test" station's pc and do the following steps:

- Prepare a list of dataset entry in SAM database which has been created during a specific time period. Presently, we have taken one day before and after of the present time.
- Pick a dataset from the above list. Create a list of files which constitute the dataset.
- Check the location of file. If the location of file contains the destination node(UCSD SE SURL), it means file is available at the destination SE. Otherwise, download the dataset to the cache of the destination station.
- Update log file.

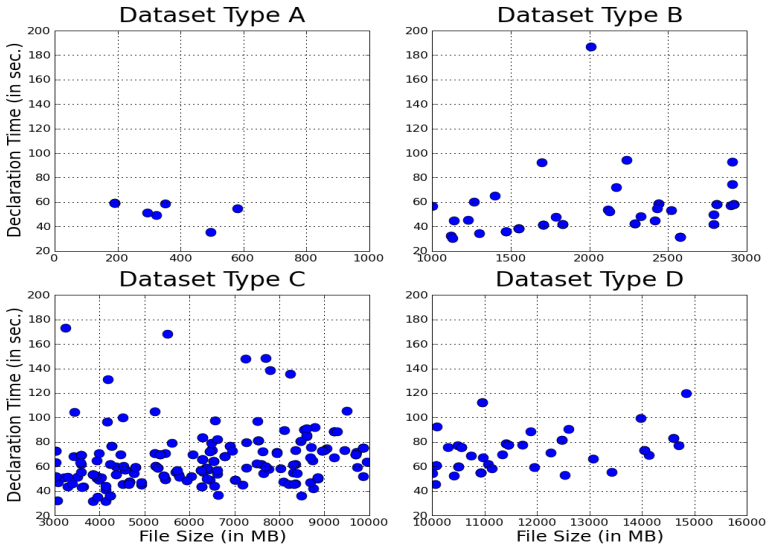

Figure 7. Time needed to declare for different type of datasets.

- Go to the next dataset and repeat the above steps.

The trans Atlantic transfer rate is shown in Figure 8. Around 150 MBPS is easily achievable for dataset type C which is much larger than the presently available rcp/fcp transfer rate (2 MBPS).

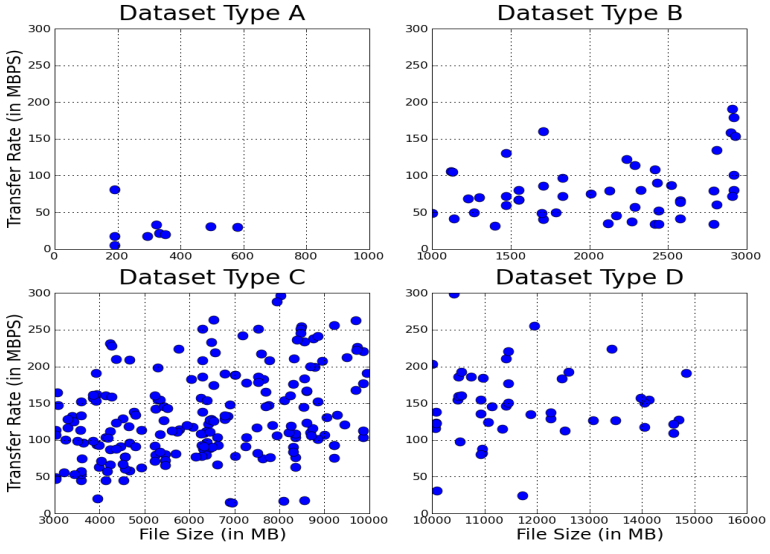

Figure 8. Trans Atlantic transfer rate for different datasets.

6. Getting files on user's desktop

The output files reside in the SAM managed SE. Following environments are needed on user's desktop.

- SAM station environment.
- SRM client on user's desktop.

The proposed model follows a fix syntax for naming the dataset corresponding to log and output files. User's can query the list of files which matches dataset using the sam command "sam list files –dim = <datasetName>". Location of files can be obtained using the command "sam locate <fileName> ". After knowing the location of the file, user can use the SRM client for transferring file from SE to desktop. A wrapper script can be written such that above changes will be transparent to CDF users.

7. Conclusions

A new data transfer model has been designed that uses SRMs as local caches for remote Monte Carlo production sites, interfaces them with SAM, the experiment data catalog, and finally realizes the file movement exploiting the features provided by the data catalog transfer layer. We also evaluated the performance of the model and its integration within the current CDF computing architecture. We found that the proposed model have better data management and transfer rate with respect to the present one.

8. Future Plan

There are also some other spin off from this study. One can use this model for transporting real data from productions site to the WN at remote site for further processing of real data.

Acknowledgment

We would like to thank Frank Wuerthwein and Abhishek Singh Rana (from UCSD), Thomas Kuhr (from Karlsruhe), Brian Bockelman (from Nebraska), Jon Bakken (from FNAL), for providing resources on SRMs.

References

[1] CDF Homepage, *http://www-cdf.fnal.gov*
[2] SAM-SRM design document, *https://plone3.fnal.gov/SAMGrid/Wiki/SAM-SRM-Design.doc/download*
[3] SRM Working Group Homepage, *http://sdm.lbl.gov/srm-wg/*
[4] SAM Homepage, *http://d0ora1.fnal.gov/sam/*
[5] D. Lucchesi "CDF way to grid" *presented at Computing in High Energy and Nuclear Physics, Prague, Czech Republic, 21-27 March 2009* **414 2009.**
[6] CAF Homepage, *http://cdfcaf.fnal.gov*
[7] dCache Homepage, *http://www.dcache.org/*
[8] srmcp Homepage, *https://srm.fnal.gov/twiki/bin/view/SrmProject/SrmcpClient*
[9] Condor Homepage, *http://www.cs.wisc.edu/condor/*