Bulk transfers in JUNO production system

Xiaomei Zhang Xianghu Zhao

14th JUNO collaboration meeting

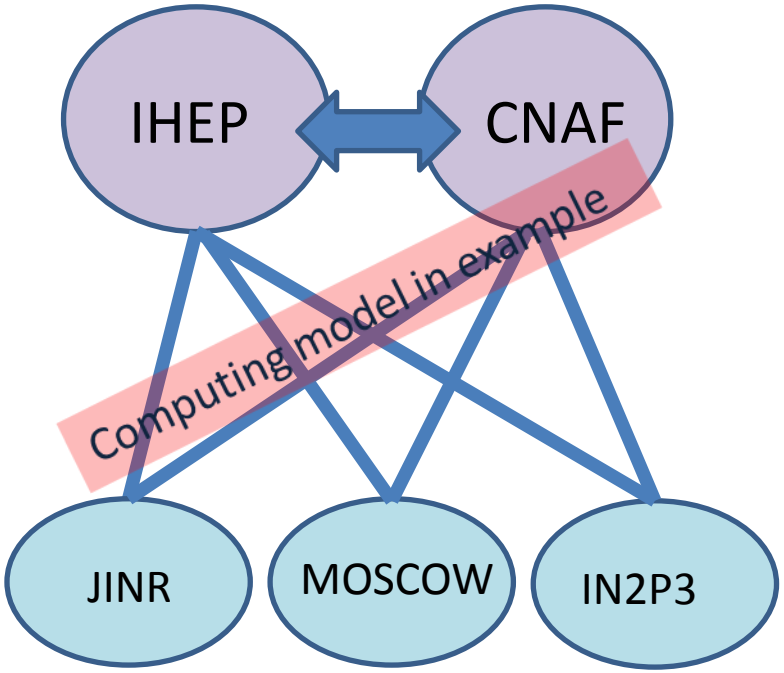
2019.7.25

Content

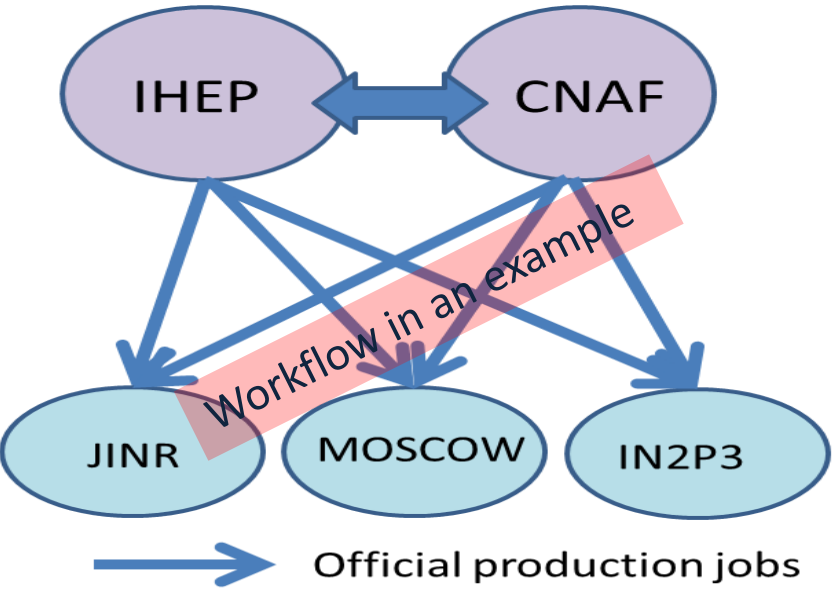
- JUNO computing model
- JUNO production system in design
- Design and architecture of transfer system
- Implementation
- Tests and performance

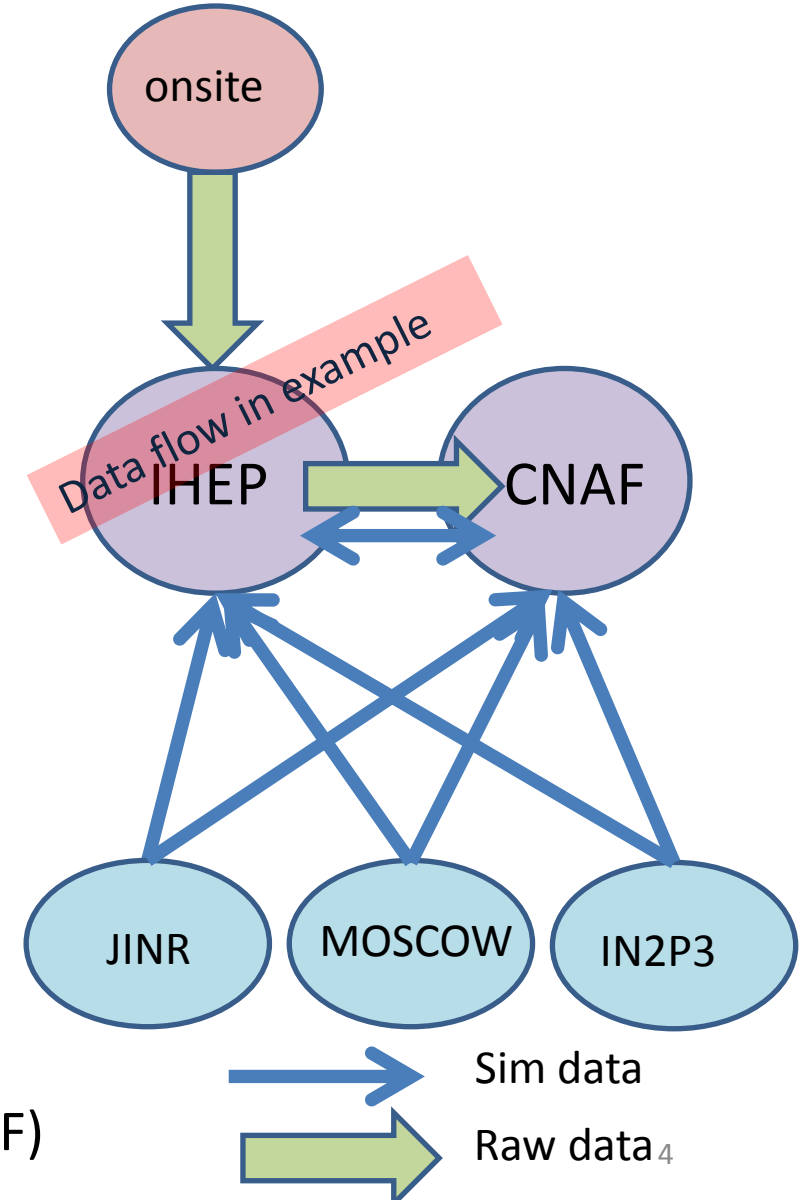
JUNO computing model in example

- The computing model hasn't been decided yet
  - Just an example to explain requirements for production system
- Focused on MC simulation process which is first considered to run in distributed computing
- All the centers join MC simulation activities
- **IHEP and CNAF**
  - Hold storage for the complete set of data, backup for each other
- **Other centers**
  - Hold storage for part of simulation and data, also for analysis data



Workflow and Dataflow for MC





- Simulation jobs are distributed by production groups to any centers
  - Workflow type: detsim, elecsim, cal, rec
- Sim data produced in other centres be copied back to IHEP or CNAF, synchronize between IHEP and CNAF
  - Dataflow type: move, replication
- Raw data would be transferred from onsite to IHEP, then from IHEP to other sites (CNAF)

JUNO production system

- It would be time-consuming if all workflow and dataflow are managed by hand
  - Complicated procedures: split, submit, reschedule, data registeration and upload, synchronize among sites……
  - Watch status of each step and take actions for next step
- JUNO production system aims to ease the management of workflow and dataflow
  - Automatically execute workflow and dataflow in a definable way
  - All activities can be monitored centrally
  - Production group can control the procedure through a steering file or web site

An example of steering files

Keep similar configuration to the existing local job submission tools – JUNOTest

[All]

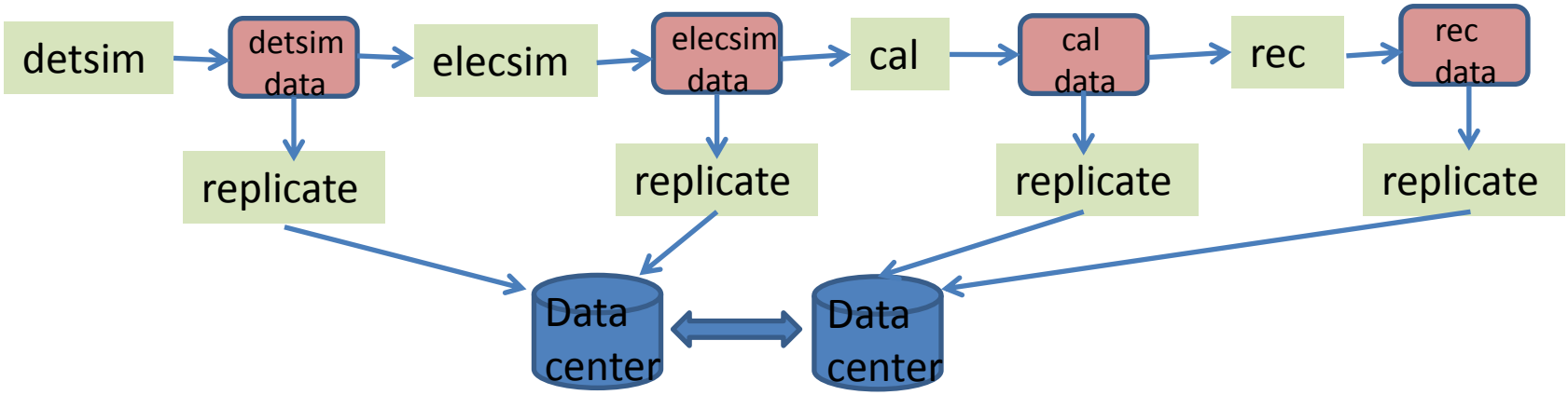
- softwareVersion = J17v1r1
- process = Positron
- prodName = JUNOProdTest
- ;site = GRID.INFN-CNAF.it CLOUD.JINRONE.ru GRID.IN2P3.fr
- outputDir = testprod/some/other/dir
- ;outputSE = ;outputMode = closest
- moveFlavor = Replication
- ;moveSourceSE = IHEP-STORM CNAF-STORM JINR-JUNO IN2P3-DCACHE
- moveTargetSE = IHEP-STORM CNAF-STORM

[Positron]

- seed = 42 ;evtmax = 1 ;njobs = 20
- tags = e+_0.0MeV e+_1.398MeV e+_4.460MeV e+_6.469MeV
- workDir = Positron; position = center
- workflow = detsim elecsim calib rec
- moveType = detsim elecsim calib rec
- detsim-mode = gun --particles {particle} --momentums {momentum} --positions 0 0 0

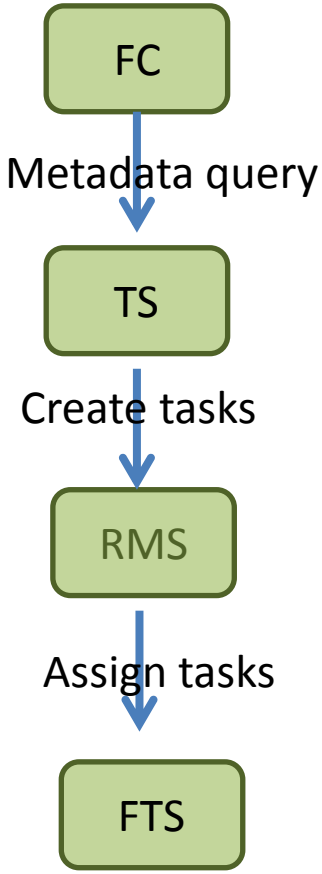
Design concept of JUNO production system

- The MC activities can be split into data processing and data management
  - Data processing: detsim, elecsim, cal, rec
  - Data management: move, replication, removal
- All the activities can be chained through datasets produced
  - Their input and output is closely cross-related
- The whole system plans to be designed in a data-driven way which allows workflow and dataflow work closely together
  - DIRAC transformation system provides basic architecture for these chains
  - Each data processing and management is designed as a transformation module with input and output data registered in metadata
    - Only detsim is an exception without input data, controlled with job number
  - Metadata in File Catalogue is a key to chain them all



Design of bulk transfers

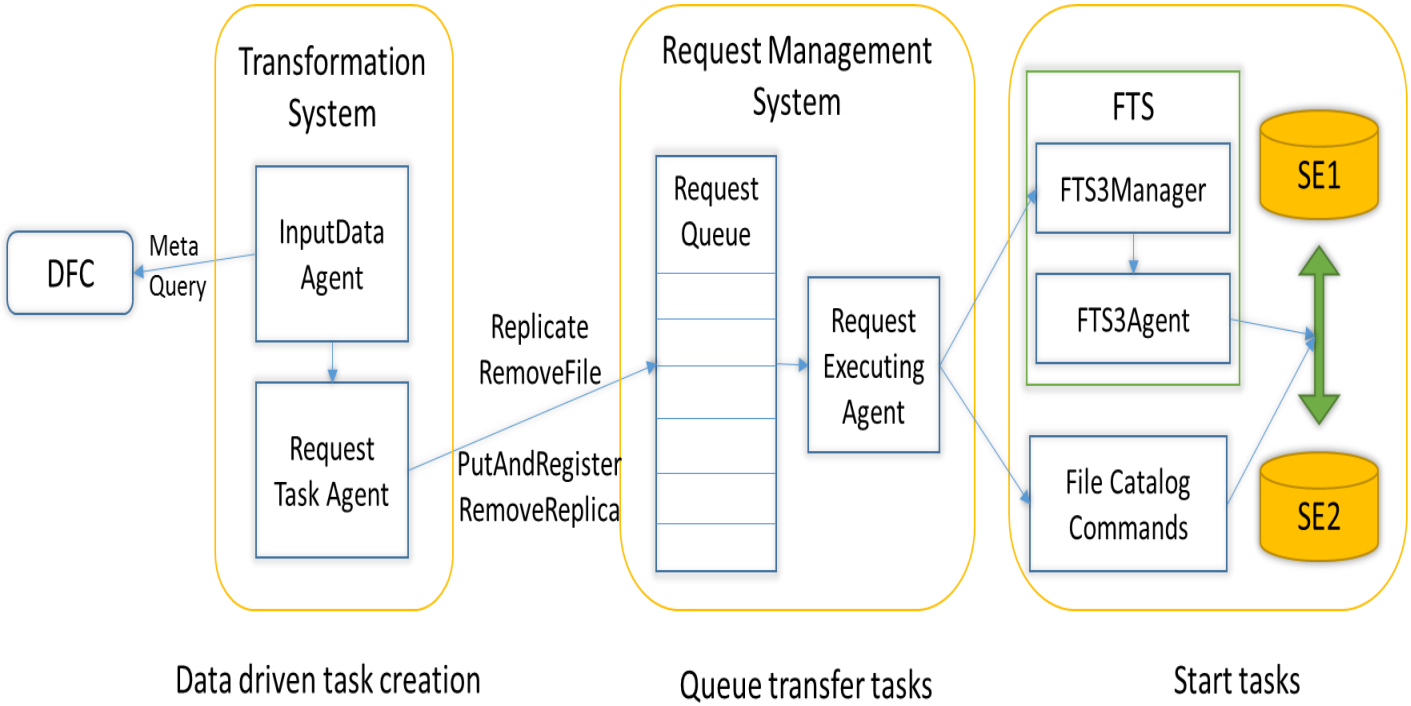
- The data management part allows data-driven data replications among sites for the produced MC data
- This part can be also used as an independent transfer system for any other official data, eg. raw data
- Design as an independent transfer system
  - Register data to be transferred in File Catalogue (FC) with metadata defined
  - Transformation system (TS) create transfer tasks based on metadata query and sent to RMS
  - Request management system(RMS) arrange transfer tasks in queue and sent to FTS (File Transfer Service)
  - FTS takes real transfer tasks and reports back status



Architecture

Four subsystems

- DFC (Dirac File Catalogue)
- Transformation system (TS)
- Request Management System (RMS)
- File Transfer Service (FTS)



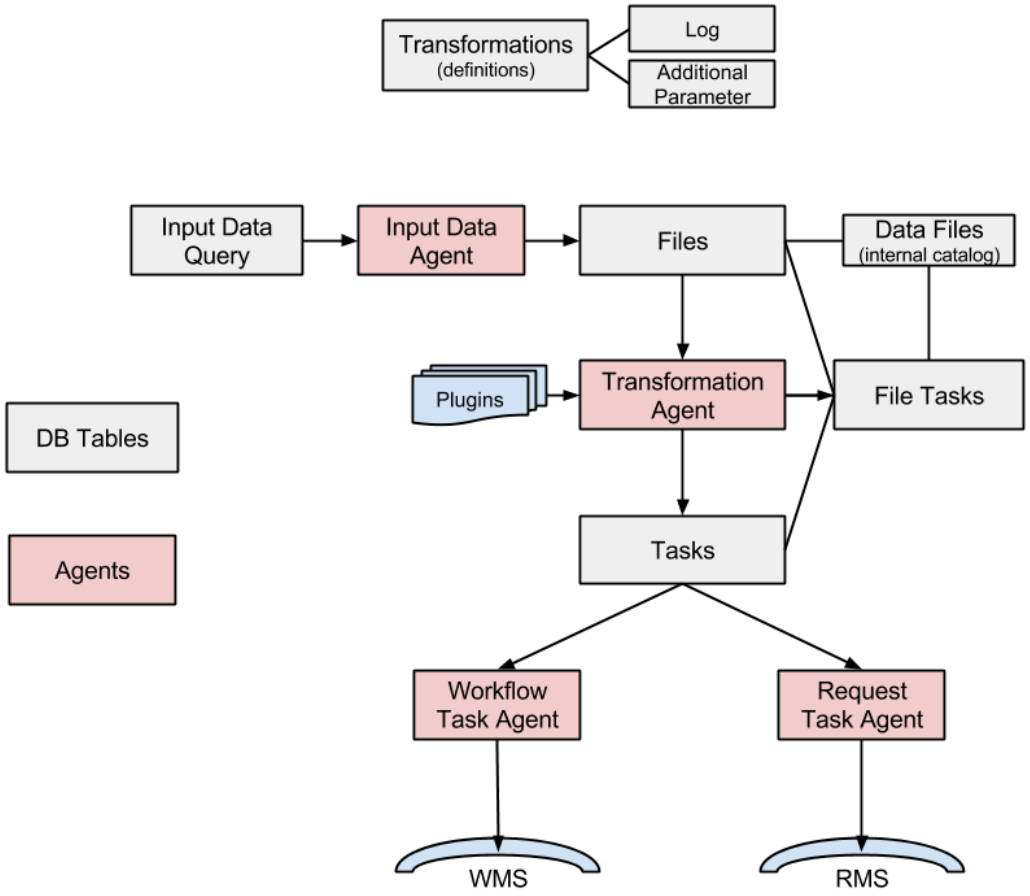
DFC

- Dirac File Catalogue
  - Meta catalogue
    - define a group of data with same properties
  - Replication catalogue
    - track location of replicas among sites
- Support both static and dynamic query from transformation system
  - Static query
    - Get a static list of files once
      - tc.addFilesToTransformation(transID['Value'], infileList)
    - This file list not changed through the whole transfer process
  - Dynamic query
    - Get a dynamic list of files collected with metadata
    - This file list allow to be changed during the process, but the transformation keep query with certain frequency
      - tc.createTransformationInputDataQuery(transID['Value'], query)
      - Simple query: juno_transfer=PmtCharacterization/container_data/Meassurements_DAQ
    - The feature allows the transfer to be triggered as soon as the first file arrived in the range of metadata query

Transformation system

- A system for handling “repetitive work” with recipes
  - i.e. create many identical tasks with a varying parameter
- 2 main cases:
  - Production jobs: the “same” job with different parameters
  - Data handling: the “same” replication or removal for a group of data
- LHCb, CTA, ILC and BelleII use it to build their own ‘Production System’ and ‘Transfer System’
- In the transfer system
  - Replicate, Move and Remove transformation modules based on TS framework need to be created to handle operations in transfers

Transformation system



- TransformationAgent
  - Processes the transformations and creates tasks given a Transformation Plugin
- InputDataAgent
  - Queries the Catalog to obtain files to be 'transformed'
- WorkflowTaskAgent
  - Transforms tasks into job workflows given a TaskManager Plugin
- RequestTaskAgent
  - Transforms tasks into data handling requests

Request Management System

- It is a very generic system in DIRAC that allows for asynchronous actions execution
  - typically for large scale data management operations like replications or removals
- Two key components:
  - Request Manager Service with ReqDB to handle queue of actions
    - Accept requests from TS, also from commands
  - Request Executing Agent to assign requests to the related services and track status
    - In our cases, it is FTS service

FTS service

- File Transfer Service
  - A very powerful independent multi-VO transfer system to handle file-by-file transfers
    - Can manage reliable and large scale transfers transparently between different storages (EOS, DPM, Object Storages, STORM, dCache, CTA, ..) and multiprotocol support (Webdav/https, GridFTP, XRootD, SRM)
    - Run in client/server mode
  - Widely used in WLCG, many experiments build its own transfer system on top of it
    - CMS PhEDEx, Atlas Rucio……
- Dirac interface services to FTS
  - FTSManager -- keep track of the submitted FTS requests
  - FTSAgent – submit FTS requests and update FC with new replicas

Implementation

- The prototype of this system has been set up
  - fts3 server has been set up
    - https://fts3.ihep.ac.cn:8449/fts3/ftsmon/#/
  - Other related services and agents have been installed and configured in DIRAC
    - TS, RMS, FTS interface services and agents
  - Transformation modules for replication have been developed
- The necessary scripts have been released to IHEPDIRAC to handle massive replication, removal, registeration operations
  - ihepdirac-transformation-transfer-metadata

    <transferName> -t <transferType> <metadata query> <sourceSE> <destinationSE>

Tests with raw data

- The raw data used for tests
  - /junofs/PmtCharacterization/container_data/Meassurements_DAQ
  - About 11TB, 810,420 files
- The command to start transfer
  - ihepdirac-transformation-transfer-metadata Meassurements_DAQ_JINR -t Transfer-JUNO juno_transfer=PmtCharacterization/container_data/Meassurements_DAQ IHEP-STORM JINR-JUNO

Monitoring

- **Status of whole process can be tracked (TS monitoring)**

ID	Status	AgentType	Type	Name	Files	Processed (%)	Created	Total Created
Request: 0								
31	Complet...	Manual	Transfer-JUNO	Trans_IHEP_JINR_Size50	300	66.6	0	6
32	Complet...	Manual	Transfer-JUNO	TestRegister8	144	0.6	0	30
33	Complet...	Manual	Transfer-JUNO	TestRegister9	224	0.0	0	23
34	Complet...	Manual	Transfer-JUNO	TestRegister10	144	0.6	0	16
35	Complet...	Manual	Transfer-JUNO	TestRegister11	144	63.1	0	11
36	Complet...	Manual	Transfer-JUNO	TestRegister12	144	56.9	0	11
37	Complet...	Manual	Transfer-JUNO	TestRegister18	224	100.0	0	13
38	Complet...	Manual	Transfer-JUNO	TestRegister16	144	100.0	0	10
39	Complet...	Manual	Transfer-JUNO	TestRegister21	1	100.0	0	1
40	Complet...	Manual	Transfer-JUNO	PMT_Measurements_DAQ	2820	80.8	0	57
41	Complet...	Manual	Transfer-JUNO	PmtCharacterization/container_data/Meassurements_DAQ	810420	100.0	0	16316

- **Individual file transfer (FTS monitoring)**

Total size	Done	Submission time	Start time	Running time	Avg. file throughput	Current job throughput
27.44 GiB	27.03 GiB	2019-04-29T08:19:17Z	2019-04-29T08:19:17Z (+0s)	468 s	1.48 MB/s	3.71 MB/s

Showing 1 to 30 out of 30

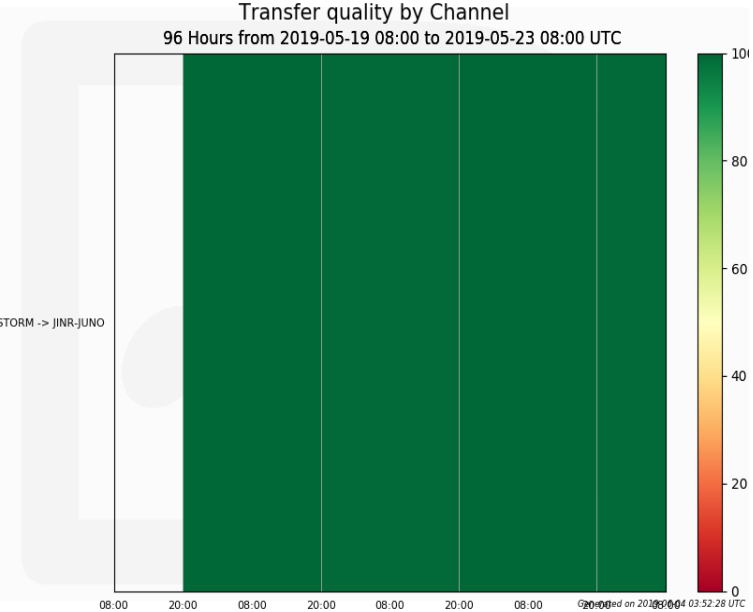
SUBMITTED | DELETE | READY | STAGING | 1 ACTIVE | STARTED | CANCELED | FAILED | 29 FINISHED | NOT_USED

First | Previous | 1 | Next | Last

File ID	File State	File Size	Throughput	Remaining	Start Time	Finish Time	Staging Start	Staging End	
+ 6658818	FINISHED	987.17 MiB	2.37 MB/s	-	2019-04-29T08:19:17Z	2019-04-29T08:26:53Z			Log
srm://storm.ihep.ac.cn:8444/srm/managerv2?SFN=/juno/TransferData/IHEP-STORM/10_mev_electron_120mev_laser_0.root									
srm://lxse-dc01.jinr.ru:8443/srm/managerv2?SFN=/pnfs/jinr.ru/data/juno/dirac/juno/TransferData/IHEP-STORM/10_mev_electron_120mev_laser_0.root									
+ 6658819	FINISHED	986.83 MiB	1.86 MB/s	-	2019-04-29T08:19:17Z	2019-04-29T08:28:34Z			Log
srm://storm.ihep.ac.cn:8444/srm/managerv2?SFN=/juno/TransferData/IHEP-STORM/10_mev_electron_120mev_laser_21.root									
srm://lxse-dc01.jinr.ru:8443/srm/managerv2?SFN=/pnfs/jinr.ru/data/juno/dirac/juno/TransferData/IHEP-STORM/10_mev_electron_120mev_laser_21.root									

Tests

- Transfers tests have been done
  - From IHEP to JINR
  - From IHEP, JINR to CNAF
  - From IHEP, JINR, CNAF to IN2P3
- The transfer quality is good
- Some problems met because of source files problems
  - Eg. Space inside file name, no permission on some files
  - CNAF StoRM SE can't accept empty files
  - IN2P3 lack of space, only have 8TB

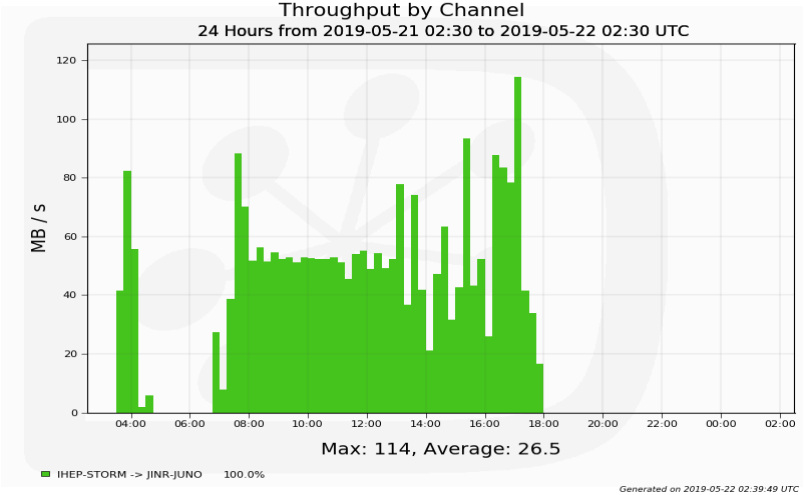



```
FC:/juno/lustre/junofs/PmtCharacterization/container_data/Meassurements_DAQ>size -l
directory: /juno/lustre/junofs/PmtCharacterization/container_data/Meassurements_DAQ
Logical Size: 11,055,135,441,481 Files: 810420 Directories: 3842

   StorageElement Size              Replicas
=============================================
 1 JINR-JUNO      11,055,135,441,481   810420
 2 IN2P3-DCACHE    4,667,421,171,629   260555
 3 CNAF-STORM     11,054,929,901,206   809269
 4 IHEP-STORM     11,055,135,441,481   810420
   Total          37,832,621,955,797  2690664
```

Performance

- The transfer speed for tests
  - from IHEP-STORM -> JINR-JUNO reach ~120MB/s
  - from IHEP-STORM, JINR-JINO->CNAF-STORM ~80MB/s
  - from IHEP-STORM, JINR-JUNO ~40MB/s
- More tests needed to understand and improve performance
- One bottleneck met
  - With dynamic query of 800,000 files, inputdata Agent for transformation system becomes slow



Conclusion

- The JUNO production system are under design and development
- The prototype of Bulk transfer part has been developed and set up
- The tests with massive raw data is successful with good quality
- Further tests needed to understand more about performance