Application of iRODS metadata management for cancer genome analysis

iRODS User Group Meeting 2016

e:Med SYSTEMS MEDICINE

e:Med initiative
sponsored by the

Federal Ministry of Education and Research

grant no. 01ZX1303A

Lech Nieroda

University of Cologne
Regional Computing Center

Agenda

- Who are we?
- SMOOSE Project: data management requirements
- Why use iRODS?
- Our iRODS integration
- Various Concerns

Regional Computing Center (RRZK)

University of Cologne

- High Performance Computing for scientists of North Rhine-Westphalia
- IT infrastructure & support for University of Cologne

CHEOPS HPC Cluster at the RRZK

Photo: D. Both, Bull GmbH

Peak- and Linpack Performance	100 TFLOP/s and 85,9 TFLOP/s
Top500 Rank	90 (11/2010)
Number of Nodes / Cores	841 / 9712
Total RAM	35,5 TB
Total Storage (Lustre + GPFS)	500 TB + 900 TB
Interconnect (Infiniband and Ethernet)	QDR 40Gb/s and 10Gb/s

Why a data management system?

SMOOSE Project

Systems-level analysis of MOdulators of Oncogenic Signaling

- Identification of new genetic markers for diagnosis and treatment of cancer
- Devise therapy and move findings into clinical application
- Interdisciplinary consortium with 10 sub-projects

rrzk

Department of Translational Genomics

UNIKLINIK KÖLN

Universitätsklinikum Essen

CECAD COLOGNE Excellent in Aging Research – Exzellent in Alternsforschung

thp Institute for Theoretical Physics University of Cologne

technische universität dortmund

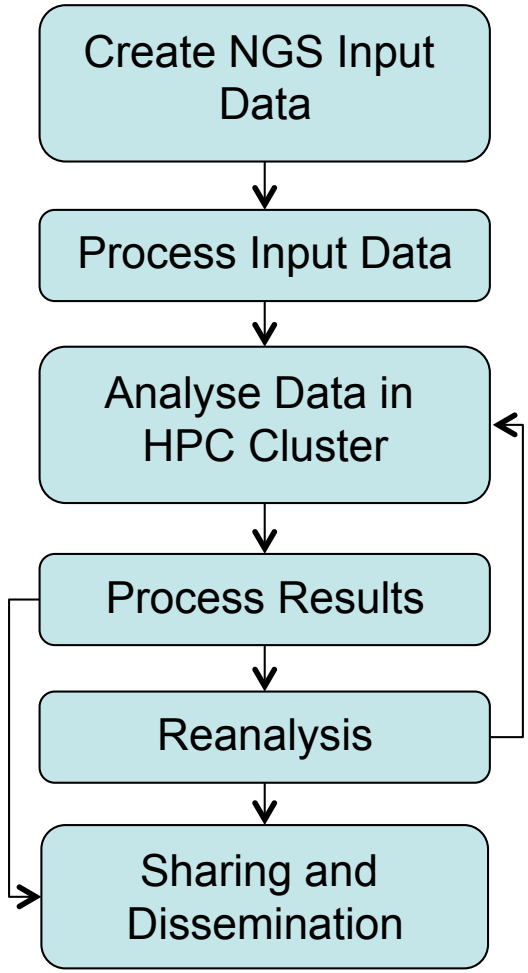
LungCancerGroup Cologne

e:Med initiative
sponsored by the
Federal Ministry of Education and Research
grant no. 01ZX1303A

Our part in SMOOSE Project:

- Optimize cancer genome analysis workflows
  - Computation:
    - Optimize speed/throughput
    - Miniaturisation, use of accelerators
    - Lower operational cost
  - Data handling
    - Efficient & redundancy free storage
    - Secure access to patient data
    - Quick reanalysis anad result dissemination

NGS Data Life Cycle:



Cross-section requirements:

Authenticate & Track Access

Verify Metadata

Verify data integrity

Data discoverability

iRODS

- Allows freely customizable and searchable metadata
- Rule-engine allows predefined actions in regular intervals, triggered by events or manually
- Any data operation can be augmented with matching actions, allowing audit trails
- Fine grained access control
- Virtual Filespace that can be adapted to organizational structures

How did we implement it?

Metadata groups:

INPUT	ANALYSIS RUN	RESULTS
– Data Provider	– Analyst	– N_reads_align
– Sample_ID	– Sample_ID	– N_mean_coverage
– Sample_Type	– Sample_Type	– N_targets_hit
– Species	– Reference_ID	– N_n_reads
– Input format	– Applications	– N_n_overlap
– Project_Name	– Parameters	– N_reads_in_target
– …	– …	– ...

- *Metadata can be entered manually or created dynamically, for example by parsing log files*
- *Metadata is intended for various levels (Project, Sample, Run, ...)*
- *Metadata needs to be verified to ensure consistency!*

Metadata verification

Example CSV Sheet with INPUT-Metadata:

	A	B	C	D
1		**MetaData INPUT**		
2	**Attribute**	**Value**	**Value Domain**	**Further tests**
3	LocalPath	/test/path/sample	[a-zA-Z0-9_+-/.]	Path readability
4	Filename1	testfile_T.bam	[a-zA-Z0-9_+-.].{bam|fastq}	File readability
5	Filename2	testfile_N.bam	[a-zA-Z0-9_+-.].{bam|fastq}	File readability
6	Sample_ID	P1234-PB03	[a-zA-Z0-9_-+.]	None
7	Sample_Type	exome	{exome|genome}	None
8	Project_Name	SMOOSE	[a-zA-Z0-9_-+.]	None
9	Data_Provider	Max Mustermann	[a-zA-Z-.]	Name DB
10	Species	human	{human|mouse}	None

Verification within a perl-import Script:

```
foreach my $key (keys %metaHash) {
  my $value = $metaHash{$key};
  die "VERIFICATION FAILED: key $key not known, stopped" \
          if not exists $verificationPatterns{$key};
  my $pattern = $verificationPatterns{$key};
  die "VERIFICATION FAILED: value $value of key $key doesn't verify with $pattern, stopped" \
          if not $value =~ m/^$pattern+$/;
  die "VERIFICATION FAILED: path $value cannot be accessed, stopped" \
          if $key eq 'LocalPath' and not -r $value;
  ...
}
```

Virtual Paths

- Perl scripts create unique, predefined Virtual Paths according to metadata values
- Access control to those predefined V. Paths has been set up for appropriate user/group
- An upload/download only succeeds if the user/group may access the Zone, Project, Sample_Type, Sample_ID and thus the V. Path

```
/<Zone>/archive/<Projectname>/<Sample_Type>/<Sample_ID>/input
/<Zone>/archive/<Projectname>/<Sample_Type>/<Sample_ID>/run_1
/<Zone>/archive/<Projectname>/<Sample_Type>/<Sample_ID>/run_2
...
```

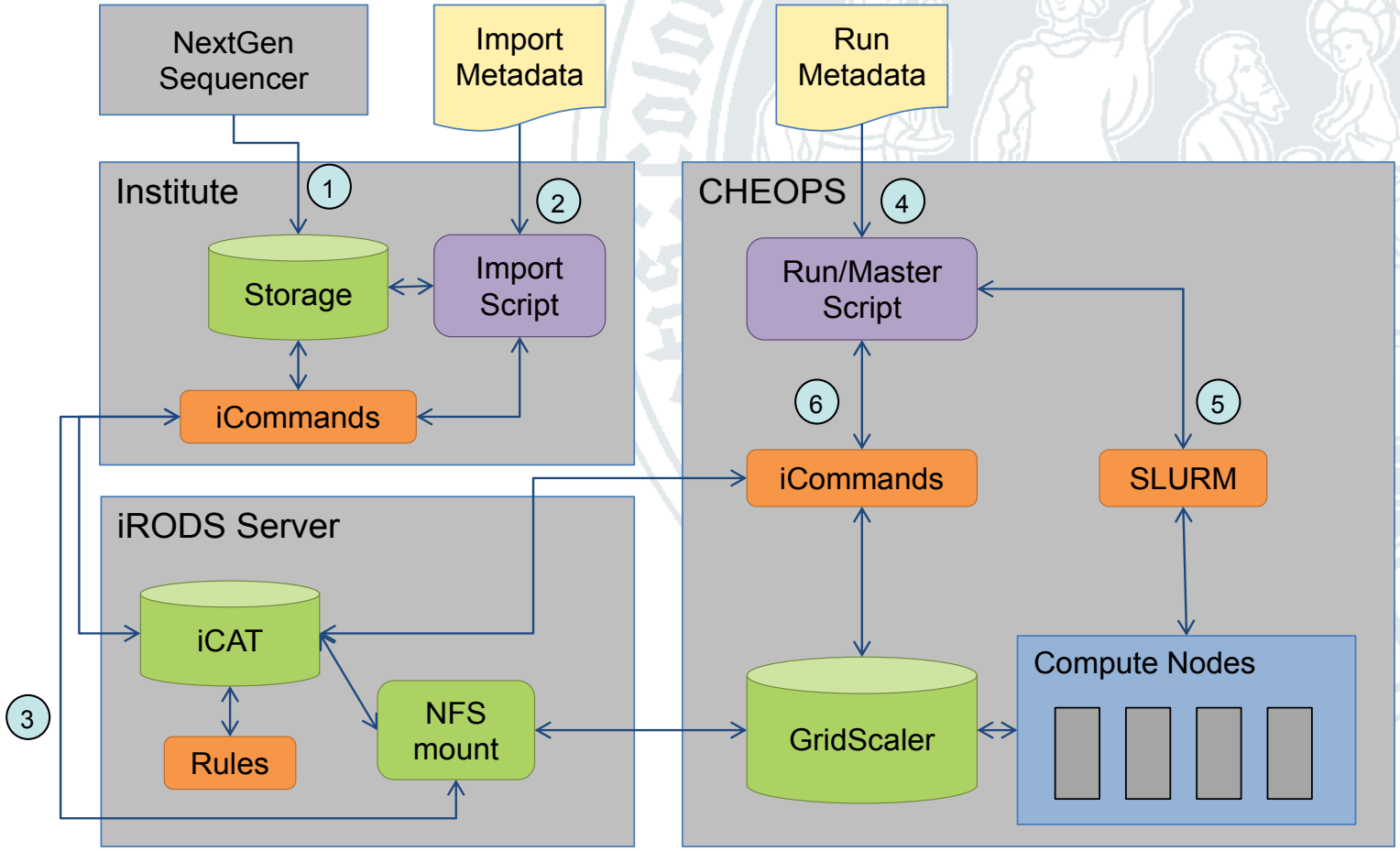
➢ *Leveraging fine grained iRODS access control!*

iRODS rule example

- A simple rule to fetch input files for a certain Zone, Project, Sample type, Sample ID:

```
getSampleFiles {
   #Input parameters:
   # Sample_ID, Sample_Type, Project, Zone
   #Output:
   # List of Matching input files
   #Example launch:
   # irule -F getSampleFilesExt.r "*Sid='testid'" "*Project='testproject'"
   *Coll="/*Zone/archiv/*Project/*Sid/*Stype/input“
   msiExecStrCondQuery("SELECT DATA_NAME WHERE COLL_NAME = '*Coll‘ and \
       META_DATA_ATTR_NAME = 'Sample_ID' and META_DATA_ATTR_VALUE = '*Sid'",*QOut);
   foreach(*QOut) {
       msiGetValByKey(*QOut,"DATA_NAME", *File);
       writeLine("stdout","*Coll/*File");
   }
}
INPUT *Sid="test", *Stype="exome", *Project="test", *Zone="SMOOSEzone"
OUTPUT ruleExecOut
```

Data Flow


NextGen Sequencer
Import Metadata
Run Metadata
Institute
1
2
Storage
Import Script
iCommands
CHEOPS
4
Run/Master Script
6
5
iCommands
SLURM
iRODS Server
iCAT
3
NFS mount
Rules
GridScaler
Compute Nodes

Security and privacy Concerns

- Anonymization/Pseudonymization of genetic samples
- SSL between iRODS Server <-> iCommand Client
- Host-based FQDN Certificates restrict allowed hosts
- PAM authentication (Kerberos failed with 4.1.6!)
- One dedicated server provides access to iRODS ressources
- File access in vault restricted to single irods user
- NFS access restricted to single irods user

➢ *iRODS has full control over stored Data/Metadata!*

Data control

Should iRODS manage only metadata and cede file control to the file system?

Pro:

- Data security relies on well known file systems instead of internal iRODS authentication
- Faster data access as long as it‘s on the same server since upload/download is no longer necessary

Con:

- Metadata can point to nonexistant or changed files!
- Filesystem must support POSIX ACL
- File registration requires admin or file owner
  - Loosening of NFS export restrictions
  - Loosening of iRODS permission checking rules
- Direct Access plugin requires root access
  - Loosening of NFS export restrictions

Summary

- Verifiable data schemas enforce metadata consistency and enable a hierarchical file space with predefined locations, making access control easier
- Efficient access control is paramount in the clinical context
- We have decided to rely on iRODS authentication to use it for both data and metadata management, while tightening security and restricting services
- This resulted in a comprehensive system including results as well as the underlying sources with matching, discoverable descriptions and reasonable security

Questions?