Large-scale EXecution for Industry & Society

A TRANSNATIONAL DATA SYSTEM FOR HPC/CLOUD COMPUTING WORKFLOWS BASED ON iRODS/EUDAT

iRODS UGM

9 June, 2021

MARTIN GOLASOWSKI (IT4I, CZ)
MOHAMAD HAYEK (LRZ, DE)
RUBÉN J. GARCÍA-HERNÁNDEZ (LRZ, DE)
LEXIS Project Consortium
Large-scale EXecution for Industry & Society

- HPC & Cloud resource providers
- Scientific institutions
- Industrial companies & SMEs
- Information Technology providers

At the confluence of HPC, Cloud Computing & Big Data

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825532.

START DATE
January 1st 2019

DURATION
36 Month

Funding / Overall Funding
€12.2 M / €14 M
LEXIS project challenges

- **Dynamic, data-aware and complex workflows orchestration**
  - Execute complex workloads on Cloud and HPC
  - Easy access to state-of-the-art compute resources
  - REST-based APIs
  - Federation of supercomputing centers
  - Real-time deadline-aware workflows over both Cloud and HPC

- **Cross-site data and metadata management solution**
  - Move data between various resources using single API
  - Distributed solution based on iRODS
  - Distributed data staging between resources

- **Data sharing between Cloud and HPC resources**
  - Accelerated by dedicated Burst Buffer nodes, high bandwidth network and FPGA cards for on-line processing (I/O acceleration)
LEXIS – What it is?

Advanced, distributed platform for HPC/Cloud/Big Data workflows, with Orchestration/Data solutions.

- Distributed Big Data & Simulations
- Unified AAI
- Data Curation, MGMT, Sharing, Valorisation (EUDAT/iRODS)
- Easy uptake by SMEs, Science, Industry „HPC…aaS“
- Fine-granular accounting/billing (CYCLOPS)
- Cloud/HPC based workflows (YORC/HEAppE)
LEXIS Distributed Data Infrastructure (DDI)
High level view of the LEXIS HPC, Cloud & Big Data federation
LEXIS DDI - Storage tiers

- Collaborative & federated data (across HPC providers)
- Cloud file storage (at each HPC provider e.g. CEPH)
- Regular file servers (NFS home directories)
- HPC / Parallel File Systems (GPFS, LUSTRE…)
- Smart Burst Buffers (NVMe acceleration)
- Near-RAM
- LAN file access
- Parallel block IO

Internet and scientific networks

Federation

HPC Provider

I/O response time
LEXIS PILOT Use-Cases

**Aeronautics**
- Computation Fluid Dynamics (CFD), Rotating parts (gearboxes), 3D Visualization

**Earthquakes & Tsunamis**
- Earthquakes & Tsunami prediction models, geographic and urban databases, emergency organization, urgent computing

**Weather & Climate**
- Weather & Climate models (WRF) and various post-processors for flood, wildfire & agriculture applications
User experience

LEXIS portal & 3D remote visualization

ALL-IN-ONE WEB INTERFACE

• Manage client organization
• Manage projects
• Provision and execute application workflows
• Manage data
• Interact with large 2D and 3D results remotely in real time
LEXIS
DISTRIBUTED DATA INFRASTRUCTURE (DDI)
WORK PACKAGE 3
LEXIS DDI Integration
Distributed Data Infrastructure for the User – leveraging EUDAT components

**DDI** (Distributed Data Infrastructure with Metadata Handling / FAIR)

- **Portal**
  - Data / Workflows / Visualisation
- **Monitoring System**
- **AAI** (Authentication & Authorization Infrastructure)
- **DDI**
- **Data Discovery API**
- **Data Transfer API**
- **Monitoring/Billing API**

**Local Storage Systems**
LEXIS DDI - Data federation

LEXIS WP3 (Lead: LRZ) – leveraging iRODS & EUDAT B2SAFE (and B2HANDLE, B2STAGE)

HPC                  Cloud
Burst Buffer
orchestrated
via staging
REST API

iRODS/iCAT Servers LRZ (redundant)

FEDERATION – MIRRORING – PREFETCH

iRODS/iCAT Servers IT4I (redundant)

LRZ: „DSS“ IBM Spectrum Scale/GPFS

EUDAT/B2SAFE

IT4I: CEPH Storage

Burst Buffer
orchestrated
via staging
REST API

iRODS/iCAT Servers LRZ (redundant)

LRZLexisZone

iRODS/iCAT Servers IT4I (redundant)

IT4ILexisZone
USING OPENID IN IRODS
iRODS OpenID integration

- OpenID
  - Web based authentication protocol – JWT tokens

- Keycloak
  - Open source Identity and Access Management solution – used in LEXIS as identity provider
  - Single-Sign On, Identity Brokering and Social Login, User Federation, Client Adapters
iRODS OpenID integration

- Patches introduced by LEXIS project to **auth_plugin_openid** and **python-irodsclient**
  - Tokens larger than > 1024 B do not fit the username field in iRODS protocol
  - USER_PACKSTRUCT_INPUT_ERR: Use token hash instead of full JWT token
  - Other optimizations and extensions

[https://github.com/lexis-project/](https://github.com/lexis-project/)
SETTING UP IRODS IN HIGH AVAILABILITY
HIGH AVAILABILITY SETUP

- A version of HAIRS (High-Availability iRODS System, cf. contributions of Kawai et al. to this meeting series) was deployed
  - Two instances of the ICAT server
  - A frontend instance containing HA Proxy
  - All three instances refer to themselves with the FQDN of the iRODS server

- Small problems (4.2.8):
  - Lots of error messages in the rodsServerLog
    - readWorkerTask - readStartupPack failed. -4000
  - Noisy logs causing the failure when executing some iRODS rules
    - Github issue #5471
    - readWorkerTask - readStartupPack failed. -4000
HIGH AVAILABILITY SETUP (continued)

- A redundant PostgreSQL database setup with repmgr and pgpool was deployed
  - Two instances of PostgreSQL containing the ICAT database
  - Replication between the two instances is enabled through repmgr
  - At a certain point in time, only one instance is set to primary and read/write access is allowed to the database

- Failover mechanism
  - Pgpool with an instance of PostgreSQL is deployed on a third machine.
  - Pgpool checks the status of the primary and the secondary databases.
  - When the primary database is down, pgpool triggers a failover mechanism
INTEGRATION WITH EUDAT
LEXIS DISTRIBUTED DATA INFRASTRUCTURE (DDI)

Immersion in European Data Management (EUDAT): EUDAT-B2* modules used

- **B2SAFE** – iRODS + add-on for policy-based data mirroring
- **B2HANDLE** – Persistent Identifier Provider → **FAIR** (Findable – Accessible – Interoperable – Reusable) Data
- **B2STAGE** – High Performance Data Movement
- **B2FIND** – Searchable Metadata Aggregator
- **B2ACCESS** – Authentication and Authorisation
- **B2DROP** – Data Workspace
- **B2SHARE** – Searchable Data Repository

Source: de Witt, S., “The Data Lifecycle” – presentation in EUDAT context
https://eudat.eu/sites/default/files/Session1-EUDAT%20Services%20in%20the%20DLC-compressed.pdf
The FAIR side of LEXIS: Metadata, PIDs

Findable, Accessible, Interoperable, Reuseable Research Data

• Most basic FAIR data requirements:
  – metadata
  – (world-)unique dataset identifier

• Metadata in LEXIS:
  – stored in iRODS Attribute-Value(-Unit) store for each data set
  – schema oriented at the basics from DataCite (schema.datacite.org)

• PIDs in LEXIS: B2HANDLE

• Aiming for findability of LEXIS public data sets via EUDAT-B2FIND
B2* SERVICES in LEXIS

• **B2HANDLE**
  - Based on the Handle System which offers a very reliable resolution service.
  - Adds metadata to an iRODS object/collection containing a unique PID and the PIDs of children objects/collections.

• **B2SAFE**
  - Adds a plugin on top of iRODS
  - Uses B2HANDLE and iRODS native rules to replicate data and keep track of children datasets

• **B2STAGE**
  - Adds a GridFTP server connection to iRODS.
  - Allows users to ingest data into iRODS through the reliable, high-performance GridFTP protocol
CUSTOM APIS AND THE USE OF IRODS PYTHON CLIENT
LEXIS IRODS API

• The LEXIS iRODS API is used to:
  ◦ Create and delete users across the federated iRODS zones
  ◦ Create projects collections across the federated iRODS zones
  ◦ Sets user’s ACLs based on project rights
  ◦ Provides a token service that is used to connect to iRODS

• iRODS python client fork
  ◦ The python client had to be forked to support openid authentication
  ◦ https://github.com/lexis-project/python-irodsclient/tree/openid_20201105
LEXIS Staging API
Overview of the Staging API

- Django based RESTful API
- Scope: LEXIS orchestrator can move data by simple HTTP request
  - between iRODS,
  - Cloud, and
  - HPC resources at all LEXIS centers.
- Uses LEXIS AAI and the HEAppE middleware to authenticate the requests and the access to the resources
- Deploys a queuing system using Celery and RabbitMQ to allow asynchronous requests.
LEXIS Staging API (continued)

Architecture

```
+----------------+               +----------------+               +----------------+
| API call       |               | RabbitMQ Server |               | RabbitMQ Server |
| API response   | <-- WSGI  --> | Staging API server | <-- Django | Celery worker    |
| Apache HTTP Server |            |            | REST Framework |            | Celery worker    |
|                |               | RabbitMQ       | Task queue    |               |                |
| PostgreSQL Server |          |                |               | RabbitMQ      |
|                |               | Staging API client |            |                |
```
Encryption and Compression API

• Django based RESTful API
• Deploys a queuing system using Celery and RabbitMQ to allow asynchronous requests.
• Allows user to encrypt and/or compress data before staging it to iRODS
• Encryption:
  ◦ Uses aes-256-ctr
  ◦ 1 encryption per project
  ◦ Uses a dedicated machine with 64 VCPUs and NVME disk to perform the encryption
  ◦ Available at each center
• Compression:
  ◦ Staging large number of small files into iRODS results in a slow data transfer rate
  ◦ Compressing the data before moving it to iRODS improves the transfer rate by up to x12
  ◦ Uses a dedicated machine with 64 VCPUs and NVME disk to perform the compression
  ◦ Available at each center
**CONCLUSIONS AND OUTLOOK**

- LEXIS European Cloud-HPC Workflow Platform (H2020) works with a Distributed Data Infrastructure based on iRODS/EUDAT-B2SAFE
- iRODS was chosen due to its ability to federate geographically distributed data sources
- Different setups of iRODS were tested. The HAIRS deployment with redundant PostgreSQL setup, provided highly available access to the federated data infrastructure.
- EUDAT services provided us with the means to achieve the DATA FAIR principles
- The iRODS Python client has been crucial for developing interfaces to other LEXIS components.
- The iRODS OpenID connection provided an obstacle when trying to connect the LEXIS AAI to iRODS. Although we found a workaround, it would be interesting to see a native iRODS implementation in iRODS 4.3.X
CONTACT

Martin Golasowski
LEXIS Task 3.2 & 3.4
martin.golasowski@vsb.cz

Mohamad Hayek
LEXIS Task 3.3 lead
mohamad.hayek@lrz.de

Website & further contacts:
www.lexis-project.eu

CONSORTIUM

Large-scale EXecution for Industry & Society