Final Project Report

National Science App Store

Connecting Data and Processing Power

1. The project and the infrastructure

Nature, purpose and scope

The project's vision was to create a simple and intuitive environment, which allows researchers to control their data throughout its full life cycle: discovery and re-use, analysis and post-processing, data management, sharing and archiving, handling of sensitive data. As the needs for research are varied, the solution should be flexible and able to integrate/interface to different, possibly external, tools. We created a private cloud solution designed for research, that integrates a secure data storage, data management capabilities, analyses workflows, (national) computing infrastructures, data archiving and publishing tools. Such research cloud solution is able to interface to both national (e.g. HPC centers) and European services like e.g. Zenodo, EUDAT, Datacite, ORCID, or the future EOSC portal.

From the researcher perspective, having a single environment with a "complete" set of digital tools that allows to control data throughout its life cycle, can increase the efficiency and quality of the research process. By taking advantage of the automation integrated in the platform, researchers can focus on the "hard" research questions instead of spending valuable time on tasks such as data handling (e.g. moving data across different systems, control access/sharing of datasets), re-implement analysis workflows, ensure reproducibility, annotate data sets with relevant metadata information, search for relevant datasets, etc. The concept of this research cloud is similar to the one of successful commercial cloud solutions, which allow businesses to increase productivity by focusing on their core products instead of infrastructure operations.

Project organization and execution

The project has been anchored at the SDU eScience center and it was organized as an open consortium among partners. A kick-off meeting was organized at the start of the project, after which the project leader (SDU) has overseen and coordinated the project activities. The SDU eScience center had the responsibility of the development of the core components of the project, while the proposed proof-of-concept research apps have been developed in collaboration with all the members of the consortium. All the code for the software infrastructure produced in this project and its documentation is publicly available on github: https://github.com/SDU-eScience/NSAS-project. The infrastructure can be accessed via a web interface and can uses WAYF as identity service to provide access to Danish researchers. The resources consumed via the infrastructure, such as storage space and computing resources, can be billed to the users or their institutions. The infrastructure will be put in a production environment at SDU in 2019.

Technical setup

The infrastructure is accessed through a modern web-application written using the React library. This front-end application communicates via REST APIs to the backend system.

The backend powering the infrastructure has been designed with a microservice architecture. Such microservice architecture allows to deliver a scalable and reliable solution with no service interruptions. The backend is built by using open-source technologies and components and it is designed to have no single point-of-failure.

We use Ceph as our storage back-end. Ceph is an object store that provides distributed, scalable and reliable storage which is software-defined. Among other, it in in production at CERN to handle double digits petabytes (~70PB) of data. For data discovery the infrastructure integrates Elasticsearch. Elasticsearch is a distributed search and analytics engine, which is designed to be highly reliable. Kafka is used in the system for stream processing and it allows loosely-coupled communication between microservices. Having microservices be loosely-coupled is crucial to achieve better scalability. Kafka is also used as the infrastructure backbone for logging of all operations. All system activities are automatically sent through Kafka, and then shipped through Logstash into Elasticsearch. We use the analytics functionality of Elasticsearch to monitor activity in real-time. This detailed logging system is required to handle sensitive data. Each feature of infrastructure is implemented as its own microservice (e.g. authentication, storage, publishing, sharing, etc.)

For the production system, all infrastructure components have been containerized so that the system can be deployed and run on a Kubernetes cluster.

List of relevant technologies used:
web-app/frontend: javascript/typescript, react, redux, JWT, webpack, semantic-ui, REST;
backend: Ceph, Elasticstack (elasticsearch/logstash/kibana), Kafka, Kotlin/Java, Docker containers, WAYF/eduGAIN, PostgreSQL, microservices.

2. Achievement of the pilot project

The only relevant change respect to the original plan has been the use of a different platform than "data.deic.dk", as this was being redesigned at the time of the project. Therefore, we build a completely new platform based on a front-end web application and a microservice backend system (see overview above). Although more time consuming, this allows for more flexibility and better scalability of the solution. New functionality can be built into this system by adding new microservices, which scale horizontally. This has also given us the opportunity to use the latest proven technologies, resulting in a modern design for the platform.

Deliverables

"Basic app" container & "App Store"
All success criteria have been met. "Apps" are accessed from the App store which is part the front-end application and integrate seamlessly with the system: apps can access data stored on the cloud platform and they be launched from the web-interface. The user receives a live feedback of the status of launched applications, including a live standard output streams while the app is running. Data transfer to/from computing infrastructures, such as the Abacus2.0 system, is automatic without the need for manual user intervention. Applications as defined in this project use either the Docker or Singularity container format (both are supported) with a metadata description in JSON format. The user interface is automatically generated from the app description by the front-end application. An "App store" catalog is integrated in the web-application, which allows users to discover and access the applications. The "apps" functionality can be accessed programmatically via REST APIs exposed by the backend system.

"Password-less" transfer of data to/from Abacus2.0

All success criteria have been met. "Apps" are able to move data to/from the Abacus2.0 HPC system without the need for user intervention (e.g. typing a password). For this we use the same technology employed in the frontend, which is based on access tokens (JSON web tokens/JWT), and SSH in the case of Abacus2.0. In addition, a Command Line Interface (CLI) has been written, so that more experienced HPC users can transfer data to/from Abacus2.0 from the Linux shell.

Apps interface to Abacus 2.0

All success criteria have been met. "Apps" can be launched on Abacus2.0 from the web user interface in a completely automatic way. The user receives live feedback on the status of all running applications (each submitted application has its own status page). A REST API has been created to allow programmatic access to this functionality. User documentation was written to describe the use of the system (provided in the linked GitHub repository above).

Zenodo Integration

All success criteria have been met. The system integrates the functionality provided by Zenodo and allows users to publish datasets by linking to their Zenodo account. Data transfer is automatic for the user and the system can create a publication on the user's Zenodo account. The list of published datasets is also visible in the web application. There was no need to extend Zenodo APIs to meet our success criteria.

Proof-of-concept Apps

All success criteria have been met. The research teams have led the development of the specific "apps" to test the design and functionality of the new cloud system. The scientific apps created for this pilot provide a real-world test of the concept and prove that the prototype is already flexible enough to accommodate a variety of different applications. The "apps" developed are: "BWA-MEM", for genome sequence alignment to a reference genome (app leader: OUH); "tqDist: triplet distance", for measuring triplet distance between trees (app leader BiRC/AU); "RapidNJ", for reconstructing phylogenies by using the neighbor-joining method (app leader BiRC/AU); "SearchCLI: MS-GF+", to performs peptide identification against a database (app leader BMB/SDU). Each app has its own page on the app store, containing a description of the application and several other metadata (e.g. version, authors, tags, default run parameters, etc).

The current prototype has a Technology Readiness Level TRL7. It is planned to have a production system operating at SDU in 2019 for general access by researchers. The system can be accessed via WAYF, i.e. Danish researcher will be able to access the system, although the availability of resources will depend on agreements with their host institutions. As future goals, the infrastructure could be deployed at a second Danish institution and the two private clouds could be federated for resource sharing (data, computing).

3. Sustainability and vision for permanent infrastructure

SDU is further developing the infrastructure and it is planned to deploy a production system in 2019. The system can be access by all Danish researchers via WAYF. Resources (storage, compute) on the system can be billed to individual users, groups/projects or institutions. The system can facilitate the use of the national HPC infrastructures such as Abacus2.0, as demonstrated in this project. Such access can easily be extended to other (national) computing infrastructures, as it requires minimal to no changes in a typical HPC facility. SDU is in dialog with AAU for collaborations on the digital infrastructure and related technologies. Other partners, interested in the infrastructure, are welcome to join the collaboration.

Appendix 1: Financial Account

Regnskabsskema - Forskningsinfrastruktur

Grundoplysninger

1.	Bevillingens akronym og titel:	NSAS - National Science App Store					√
2.	Bevillingshavers navn, institution og email-adresse:	Claudio Pica, Syddansk Universitet, pica@cp3.sdu.dk					√
3.	Faglig kontakts navn institution og email-adresse:	Claudio Pica, Syddansk Universitet, pica@cp3.sdu.dk					√
4.	Økonomimedarbejders navn og email-adresse:	Svend Kiilerich, kiil@imada.sdu.dk					√
5.	Sagsnr. (f.eks. 5000-01234B):						
6.	Regnskab for perioden (dd-mm-åååå):	Fra	01-08-2016	Til	30-06-2018		√
7.	Type af regnskab - års eller slut (sæt kryds):	Års		Slut	x		√

Regnskab

8. Udgifter sammenlignet med seneste godkendte budget:

	Bevilling		Egenfinansiering		Anden finansiering		Afvigelser ift bevilling				
	Budget	Udgifter	Budget	Udgifter	Budget	Udgifter	Afvigelser	Afvigelser i %			
VIP-løn			914.000,00	894.000,00						√	
TAP-løn	1.350.000,00	1.390.355,29	1.335.000,00	1.324.000,00			-40.355,29	-2,99%	√	√	
Instrumenter og udstyr			450.000,00	351.863,72	500.000,00	50.000,00				√	√
Internationale medlemsbidrag											
Andet	50.000,00	9.644,71					40.355,29	80,71%	√		
I alt	1.400.000,00	1.400.000,00	2.699.000,00	2.569.863,72	500.000,00	50.000,00	0,00				

9.	Den samlede bevilling (alle år):	1.400.000,00	√
10.	Bevilget beløb for regnskabsperioden:	1.400.000,00	√
11.	Bogført udbetalt beløb fra FI i regnskabsperioden:	1.190.000,00	√
12.	Evt. overført uforbrugt/merforbrug fra foregående år:	0,00	
13.	Anden indtægt (renter mv.)	0,00	
14.	Indtægter i alt jf. ovenstående:	1.190.000,00	
15.	Udgifter i alt:	1.400.000,00	
16.	Total uforbrugt/merforbrug (punkt 13 fratrukket punkt 14):	-210.000,00	
17.	Medsend ny udbetalingsprofil, hvis boksen er afkrydset:		
18.	Erklæringer vedr. medfinansiering (sæt kryds):	x Medfinansiering x Andre kilder	√ √
19.	Medfinansiering i alt	2.619.863,72	
20.	Graden af medfinansiering ift bevilling	68,77%	

21. Evt. kommentarer til regnskabet.

Underskrift og påtegning

22.	Underskrift bekræfter, at bevillingen er anvendt indenfor bevillingsformålet og i overensstemmelse med bevillingsgrundlaget (sæt kryds):	x	√
23.	Dato og bevillingshavers underskrift:		

24. Påtegning af regnskabschef eller bemyndiget medarbejder:

	Navn:	Annette Schøn Hansen	√
	Virksomhed/institution:	Syddansk Universitet	√
	Stilling:	Chef Forskerservice-Økonomi	√
25.	EAN-nummer	5798000423084	√
	CVR/CPR-nummer for adm.	29283958	√

SDU

Annette Schøn Hansen

Digitalt signeret af Annette Schøn Hansen
Dato: 2018.09.05 11:58:49 +02'00'