STATUS OF THE LISA PROTO-DPC

LISA DPC team:

Mylène Batmanabane,
Jean-Baptiste Bayle,
Cécile Cavet,
Hubert Halloin,
Maude Le Jeune,
Etienne Marin-Matholaz,
Joseph Martino,
Antoine Petiteau,
Eric Plagnol

Outline

1. Overview
2. Status of the proto-DPC
3. From a proto-DPC to a consortium DPC

LISA

1. Overview

2. Status of the proto-DPC

3. From a proto-DPC to a consortium DPC

Overview

Context

The DPC is a set of tools provided to **ease** the challenging data analysis tasks of LISA:

- Hardware (CPU and disk) usage not a major concern
- DA itself is challenging: lot of unknowns, complex noises and pre-processing

→ Keep a simple and easy to use DPC infrastructure.

- How IT will look like in 10 years ? Will virtualization be the next standard ? (hypervisors, containers)

Our guideline

The DPC has to be easy-to-use, simple, flexible and easily upgradeable until the end of the mission.

DPC basics

1. Development environment
2. Data base / data model
3. Execution environment

Development environement

Objectives: from the basics to the more ambitious ones

1. Ease the collaborative work (from preparation to exploitation)
2. During the operation: guarantee reproducibility of a rapidly evolving and composite DA pipeline
3. In fine: keep control of performance, precision, readibility, etc

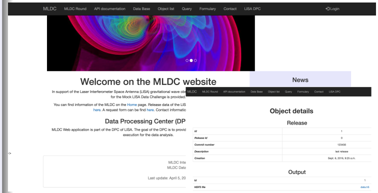
Using existing standard tools

- Control version system: widely used in the scientific community
  - allows to keep track of code revision history
  - also team project management and workflows
- Continous integration: used in some projects like Euclid, LSST
  - a suite of non-regression tests automatically run after each commit
  - working version available at any time = sucessfull tests (parsed from a web interface)
  - One can elaborate specific tests to address point 3
- Docker image: the trending tool, really easy-to-use
  - a way to encapsulate source code + its execution environment
  - software environment summarized in a single readable text file
  - impact on block 3 execution environment: smooth prototyping to operation transition

Database / data model

Motivations

- Data sharing among people and computing centers (from preparation to exploitation)
- Mainly processed, temporary or intermediate data: need meta data to use them
- Automatic tracking wrt code/pipeline revision number, parameters, input data etc etc
- Possibly a lot of information: a web 2.0 (intuitive) interface is mandatory (search engine, DB request, tree view to show data dependancies, etc)



Welcome on the MLDC website

In support of the Laser Interferometer Space Antenna (LISA) gravitational wave obs
for the Mock LISA Data Challenge is provided.

You can find information of the MLDC on the Home page. Release data of the LIS
here. A request form can be find here. Contact informatio

Data Processing Center (DP

MLDC Web application is part of the DPC of LISA. The goal of the DPC is to provid
execution for the data analysis.

MLDC Inte
MLDC Data

Last update: April 5, 20

Object details

Release

Id	1
Release id	0
Commit number	123456
Description	last release
Creation	Sept. 6, 2016, 9:25 a.m.

Output

Id	1
HDF5 file	data.h5

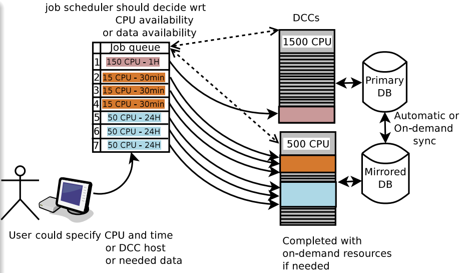
Context

- Not a big deal given the LISA data volume
- But still implies some specific developments even if using standard data format (like hdf5). One has to define LISA data model first
- Could start now to support simulation MLDC activities
  - providing the common input simulation data sets
  - then improve from there

Execution environment

Objectives: a composite computer center

- Pooling of CPU resources with a single scheduler for all DCCs
  - the user-friendly way to go
  - a dynamic CPU pool to adapt the resources to the actual needs (the economic way)
  - transfering data if needed
- Assumptions
  - it's easy to plug new hardware
  - it's easy to transfer data

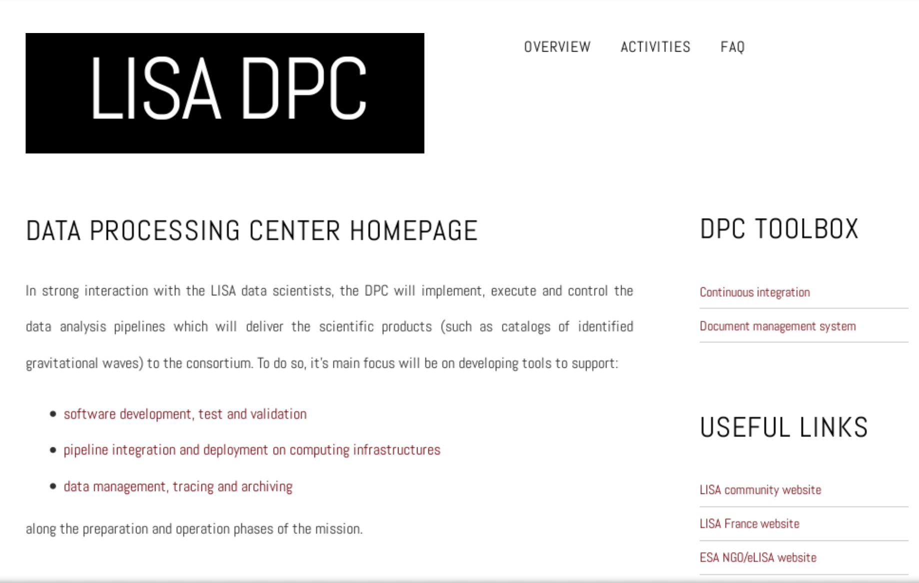


same principles than grid computing with a shorter learning phase.

A moving IT landscape

- Virtualization (the full one - cloud computing, or the light one - containers) should help with the 'easy to plug'
- Academic resources providers already considering this as the near future.
- Too early to start building it, assumptions have to be verified first.

DPC website: https://elisadpc.in2p3.fr/home



LISA DPC

OVERVIEW ACTIVITIES FAQ

DATA PROCESSING CENTER HOMEPAGE

In strong interaction with the LISA data scientists, the DPC will implement, execute and control the data analysis pipelines which will deliver the scientific products (such as catalogs of identified gravitational waves) to the consortium. To do so, it's main focus will be on developing tools to support:

- software development, test and validation
- pipeline integration and deployment on computing infrastructures
- data management, tracing and archiving

along the preparation and operation phases of the mission.

DPC TOOLBOX

Continuous integration

Document management system

USEFUL LINKS

LISA community website

LISA France website

ESA NGO/eLISA website

1. Overview

2. Status of the proto-DPC

3. From a proto-DPC to a consortium DPC

French environment

The proto-DPC emerged from APC using

- CNES financial support
- The FACe center, where LISA PF and IT people stands together
- Interaction between scientist and computer engineers driven by simulation activities. DPC supports sims and vice-versa
- CI, cloud, DA pipelining expertises acquired through other experiments (Planck, Euclid, LSST), IT people mainly working in both LISA and Euclid.

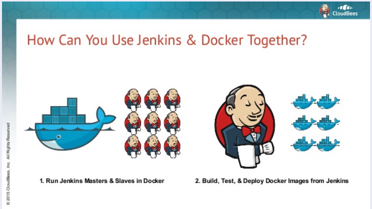
DPC support will/could be extended by

- CNES expertise on space based mission: for LISA mission, a duo CNES DPC ground segment manager + APC DPC scientific manager.
- CC IN2P3: national computing center
  - 27 000 CPU, 340 PB (CERN experiments, LSST)
  - + web services: VCS, Forge, CI, Document management system, mailing list etc
  - Openstack cloud instance: LISA first customer
- IN2P3 labs and CEA/IRFU customary connections. A common expertise network on computing (RI3, Journées Informatiques every 2 years)

What we've done

The proto-DPC started in 2015

For now, it answers point 1: development environment → gather software in a common place.



CloudBees

How Can You Use Jenkins & Docker Together?

1. Run Jenkins Masters & Slaves in Docker

2. Build, Test, & Deploy Docker Images from Jenkins

© 2015 CloudBees, Inc. All Rights Reserved

Minimal effort using out-of-the-box standard tools for:

- continuous integration (Jenkins),
- version control system (git),
- code analysis tools (SonarQube)
- virtual environment (Docker)

with interesting but moving interconnections between them → room for improvement

Put to the test by the simulation software development

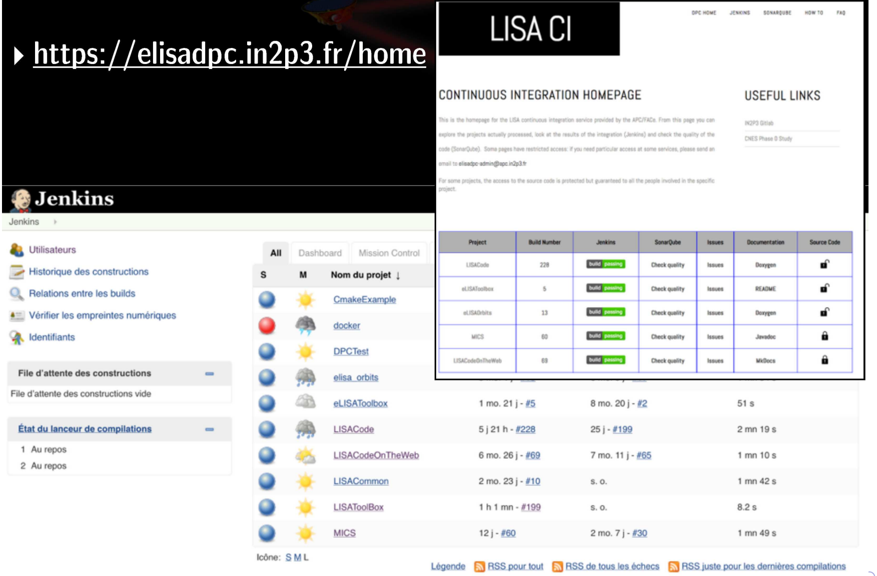
The output of this test are:

- our non regression test case: issues rapidly detected.
- on the developer side: discussions on workflows, test strategy → gather some idea on future rules and advices
- DPC quick start user guide and documentation

We definitively need more projects to really test the platform.

LISA proto-DPC

- https://elisadpc.in2p3.fr/home



Jenkins
Jenkins ›
Utilisateurs
Historique des constructions
Relations entre les builds
Vérifier les empreintes numériques
Identifiants
All
Dashboard
Mission Control
Icône: S M L
Légende
RSS pour tout
RSS de tous les échecs
RSS juste pour les dernières compilations

File d'attente des constructions

File d'attente des constructions vide

État du lanceur de compilations

1 Au repos
2 Au repos

S	M	Nom du projet ↓			
●	☀	CmakeExample			
●	⛈	docker			
●	☀	DPCTest			
●	⛈	elisa_orbits			
●	☁	eLISAToolbox	1 mo. 21 j - #5	8 mo. 20 j - #2	51 s
●	⛈	LISACode	5 j 21 h - #228	25 j - #199	2 mn 19 s
●	⛅	LISACodeOnTheWeb	6 mo. 26 j - #69	7 mo. 11 j - #65	1 mn 10 s
●	☀	LISACommon	2 mo. 23 j - #10	s. o.	1 mn 42 s
●	☀	LISAToolBox	1 h 1 mn - #199	s. o.	8.2 s
●	☀	MICS	12 j - #60	2 mo. 7 j - #30	1 mn 49 s

CONTINUOUS INTEGRATION HOMEPAGE

This is the homepage for the LISA continuous integration service provided by the APC/FACe. From this page you can explore the projects actually processed, look at the results of the integration (Jenkins) and check the quality of the code (SonarQube). Soma pages have restricted access: If you need particular access at some services, please send an email to elisadpc-admin@apc.in2p3.fr

For some projects, the access to the source code is protected but guaranteed to all the people involved in the specific project.

USEFUL LINKS

IN2P3 Gitlab

CNES Phase 0 Study

Project	Build Number	Jenkins	SonarQube	Issues	Documentation	Source Code
LISACode	228	build passing	Check quality	Issues	Doxygen	🔓
eLISAToolbox	5	build passing	Check quality	Issues	README	🔓
eLISAOrbits	13	build passing	Check quality	Issues	Doxygen	🔓
MICS	60	build passing	Check quality	Issues	Javadoc	🔒
LISACodeOnTheWeb	69	build passing	Check quality	Issues	MkDocs	🔒

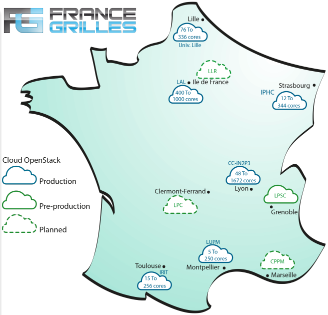
R&D on virtualization and on-demand infrastructure

A CNES R&T study performed in 2014-2017

Orchestration of docker jobs between the CNES computing center and a cloud computing provider company.
Conclusion: rapidly evolving IT landscape, doable but automation was not pushed very far.

Technology watch at APC

- Involved in the French cloud institute expert network
  - Take benefit on grid experience
  - 6 academic cloud instances (openstack)
- Actual testing of public cloud platform
  - Euclid CI server
  - 3/4 individual use cases: SDSS, Integral
  - Gather feedback from APC users.
- and container job orchestrators
  - SVOM pipeline using docker
  - Singularity installed on our small cluster



Going further: short term plan

DPC basics

0. Define and consolidate the DPC organisation (roles, basic functions, workpackages, etc. . .)
1. Development environment: could be expanded in 2 directions
   - From the user point of view: hosting more projects, improve wrt consortium needs
   - From the (lazy) administrator point of view: improving automation
2. Data base / data model
   - to be started in 2017 along MLDC needs.
   - a proto DB to distribute simulation outputs
   - together with meta data ie what's needed to reproduce the simulation. (software revision number, parameters, etc)
   - through a website providing on-line request engine (django framework).
3. Execution environment: R&D to be continued.

Contribution to the simulation software

- one way to improve on our cost forecast.
- support code development with best practices: modularity, arbitrary level of details, CPU time performance, industry proof, doc, test.
- objective: a simulation software framework used from phase A to phase E.

1. Overview

2. Status of the proto-DPC

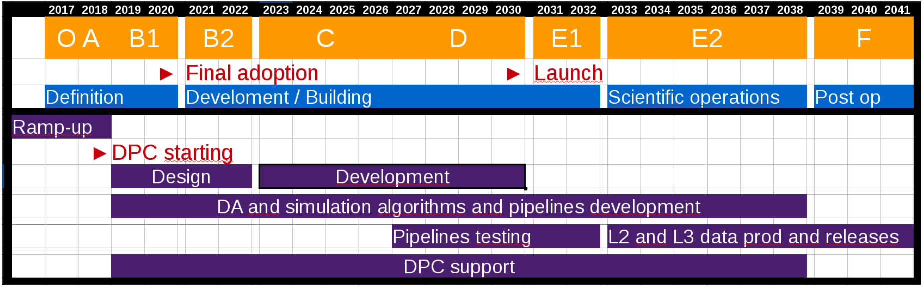
3. From a proto-DPC to a consortium DPC

A rough development plan and schedule

Driven by the following contrains

- Address the consortium needs in time: simulation effort starting now, data analysis peak in 15 years.
- Provide tools which can be easily replaced or upgraded as technologies evolve
- Consortium needs will also evolve, improve with respect to its feedback

By starting early, we'll have time to test and adjust.

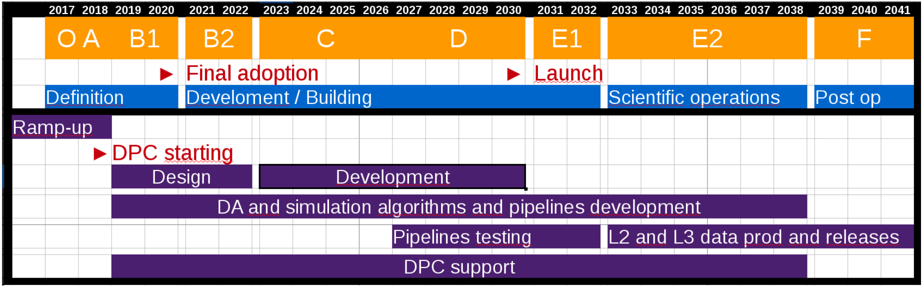


A rough development plan and schedule

Proposed plan

- DPC starting in 2018, design phase up to 2022.
- DPC development starting in phase C
- Actual testing of the regular pipeline in phase D
- Delivery of processed data to the consortium starting in phase E2

→ then loop over pipeline: process data, analyse results, refine the processing etc



Philosophy and framework

Handling rapidly evolving IT by abstracting service supplying

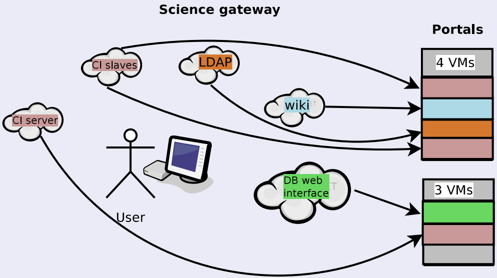
- Modularity of the DPC set of tools: wiki, CI, DB portal, LDAP, etc
- We should follow the same rules/best practises than code development (well defined interface, configuration compacted in a single readable file, automatic test, team working, etc)

This will ease the replacement of a tool, or its maintenance.
And will pay off the overhead of building (un)pluggable services and tools.

DPC as a tools provider

Added-value:

- dynamic DPC
- any service on any hardware at any time
- redondancy, smooth upgrade, confidence



Summary

In a short time scale

Cooperation could start with 2 kinds of contribution:

- on the system side:
  - check assumptions regarding DCC hardware abstraction
  - can we deploy anything (CI with Jenkins for example) on other DCC ?
  - knowing that some tools are missing: calendar, agenda (authentication / authorization)
- on the dev. side:
  - IT partners provide local support to sim/da development in their lab
  - good practice spreading
  - feedback and improvement

In other words, start to work as a team.

In a longer term

Define quantified contribution like number of hardware CPUs, or well defined workpackages. This could be drafted after this meeting.

	CPU &/or tools provision and admin	Support to code dev and optim
Short term	...	...
Long term	...	...