H2020 FETHPC-1-2014

An Exascale Programming, Multi-objective Optimisation and Resilience Management Environment Based on Nested Recursive Parallelism
Project Number 671603

D6.8 – Installation, integration and deployment of the AllScale environment and pilot applications (a)

WP6: Integration testing and pilot applications

Version:	1.0
Author(s):	Fearghal O'Donncha, Emanuele Ragnoli, Philipp Gschwandtner, Herbert Jordan
Date:	17/01/2017

Due date:	PM16
Submission date:	
Project start date:	01/10/2015
Project duration:	36 months
Deliverable lead organization	IBM
Version:	1.0
Status	Final
Author(s):	Fearghal O'Donncha, Emanuele Ragnoli, Philipp Gschwandtner, Herbert Jordan
Reviewer(s)	Benoit Leonard, Peter Thoman

Dissemination level	
PU	*Public*

Disclaimer

This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement Nr 671603. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements.

Acknowledgements

The work presented in this document has been conducted in the context of the EU Horizon 2020. AllScale is a 36-month project that started on October 1st, 2015 and is funded by the European Commission.

The partners in the project are UNIVERSITÄT INNSBRUCK (UBIK), FRIEDRICH-ALEXANDER-UNIVERSITÄT ERLANGEN NÜRNBERG (FAU), THE QUEEN'S UNIVERSITY OF BELFAST (QUB), KUNGLIGA TEKNISKA HÖGSKOLAN (KTH), NUMERICAL MECHANICS APPLICATIONS INTERNATIONAL SA (NUMECA), IBM IRELAND LIMITED (IBM).

The content of this document is the result of extensive discussions within the AllScale Consortium as a whole.

More information

Public AllScale reports and other information pertaining to the project are available through the AllScale public Web site under http://www.allscale.eu.

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	05-01-2017	First draft	Fearghal O'Donncha, Emanuele Ragnoli
0.2	06-01-2017	Comments from Kiril Dichev	
0.3	09-01-2017	Comments from Erwin Laure	
0.4	17-01-2017	Comments from Benoit Leonard and Peter Thoman	

Table of Contents

Executive Summary

This document describes the status and structure of the installation, integration, and deployment of the AllScale environment and pilot applications (D6.8). We discuss the details of a continuous integration (CI) environment hosted on one of the consortium partners (QUB) infrastructure. The CI environments consists of multiple compute nodes for functional testing, a GitLab server for source control and a server running Jenkins for continuous integration tests and source code fidelity. The document presents the different components of the AllScale codebase and how we implemented the CI environment for the different modules.

1 Introduction

In order to allow collaboration and feedback among the different consortium partners for the development, testing and validation of the AllScale software stack with the three pilot applications, requirements and needs were collected and agreed among the partners between M3 and M6 of the project. In general terms those consisted of:

1) A single shared repository that allows build automation, self-testing, automate deployment with defined security based on industry standards
2) A software versioning and revision control system to maintain current and historical versions of files such as source code, and documentation
3) A functional testing environment with computational resources to ensure that minimal performance and computational benchmarking requirements are met
4) All the above to be based on open source products

Based on those above QUB offered to host the infrastructure on one of their servers at no cost to the project. Details of the infrastructure are in the following sections.

2 Infrastructure

The continuous integration and functional testing infrastructure testing is hosted by QUB in the Institute for Electronics Communications & Information Technology. The infrastructure was built on open-source software hosted on a single server freely provided by QUB. VPN access is provided to all consortium members in line with system administration security requirements. The software management system consists of two distinct systems

- A source code repository server to store and exchange the code of AllScale components
- A continuous integration system for the ongoing testing of modifications to the AllScale components

We identified requirements of the repository server and CI system as in Table 1.

Table 1: Requirements for source control and CI system

Source code repository server	Continuous integration (CI) system
• very low performance and memory requirements (e.g. one node, single core, 2 GB RAM) • allows source code exchange • needs to be accessible to all AllScale consortium members for pulling/pushing changes	• small distributed memory cluster (e.g. 4 nodes, each multi-core and multi-processor, 32 GB RAM, ideally equipped with GPUs) • used for functional testing only, not for performance or scalability testing • needs to be able to pull changes from the source code repository server and publish the results e.g. via HTTP

	• runs all AllScale test cases whenever a new version of the infrastructure or the project package is pushed to the source repository

The infrastructure consists of one ESX R520 running 2.40 GHz Intel(R) Xeon(R) CPU E5-2407 v2 with 4 cores. The server provides 192GB RAM and 2 x 1TB SATA (Serial Advanced Technology Attachment) disk storage. The server hosts eight virtual machines (VMs) each consisting of one virtual node hosting Ubuntu version 14.04 operating system (OS). The virtual nodes are 4 test or compute nodes (for functional testing), one hosting GitLab server, one hosting Jenkins server, and a node each dedicated to the lightweight tasks of login management and VPN server. Each test node has 16GB RAM with 50GB of OS disk space and 100GB scratch space. The GitLab server has 2GB RAM with 50GB of OS space and 100GB scratch space. Similarly the Jenkins server has 2GB RAM, 100GB OS space and 150GB scratch space. The system is supported by ECIT system administration with a Request Tracker ticketing system.

2.1 Continuous Integration

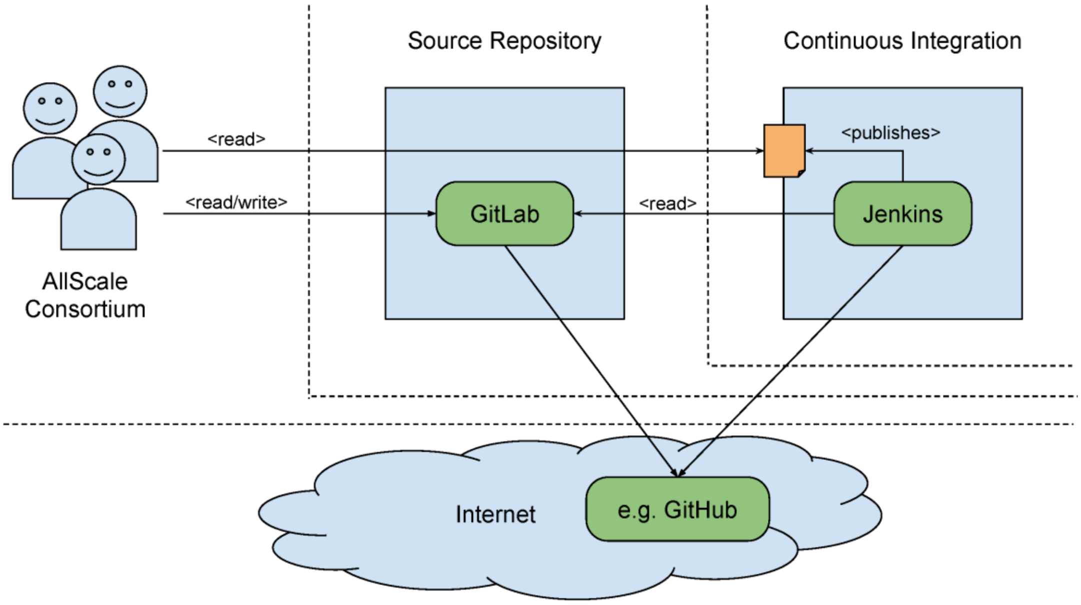

Figure 1: Continuous Integration System

The continuous integration system structure is composed as follows (Figure 1):

- Four nodes for running functional tests using the AllScale environment
- An additional node for compilation and management running Jenkins server
- The Manager/Compilation node must be able to access all source code repositories and hence have access to all AllScale components (both that contained locally in GitLab and on the internet on GitHub)
- No direct access to this system is required except for system setup and administration. Read access is provided to all developers with administration rights restricted to system administrators.

Figure 2 presents an example of a typical CI loop performed by the system

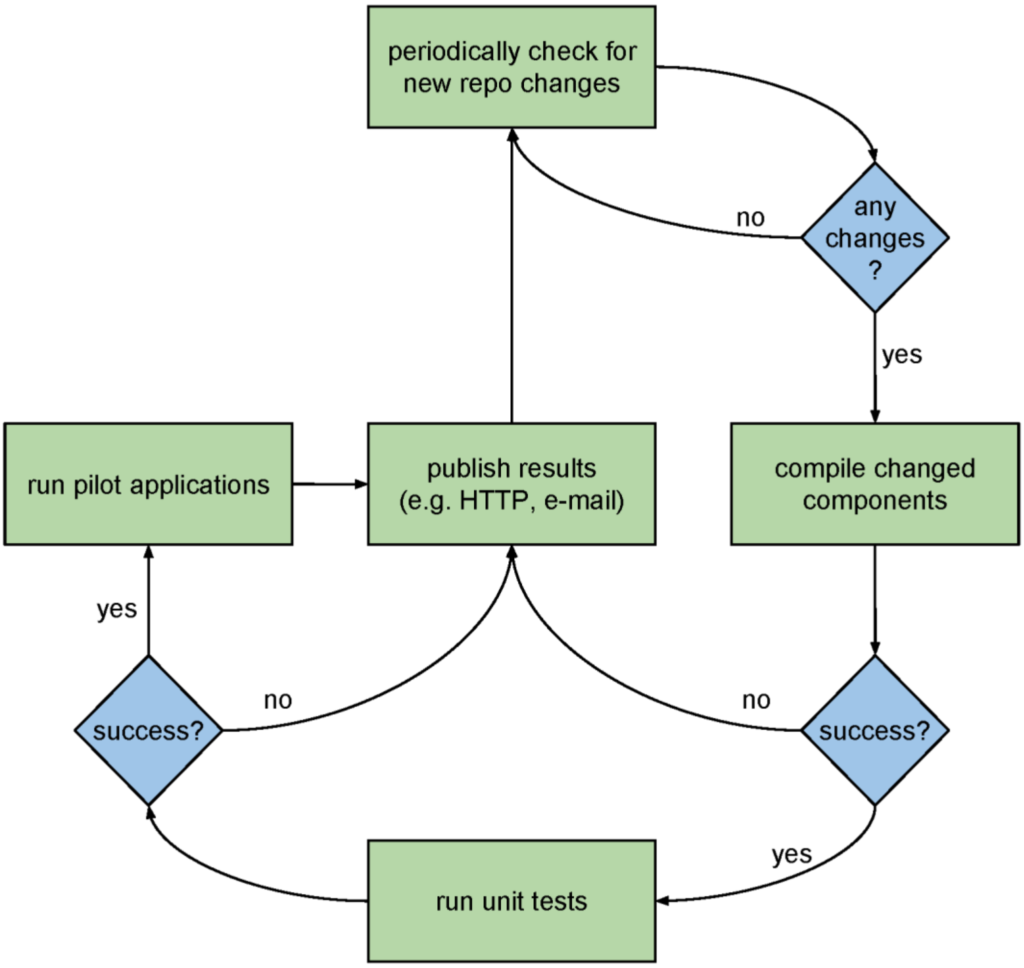
periodically check for new repo changes
any changes ?
no
yes
run pilot applications
publish results (e.g. HTTP, e-mail)
compile changed components
yes
no
no
success?
success?
yes
run unit tests

Figure 2: Example CI loop to monitor fidelity of code changes

2.2 Source Control

To maintain the software repositories for the AllScale project a single instance of a GitLab installation on the source repository server is used. This server is available at http://10.10.5.114/ (VPN access is required) and is primarily used to host codes with proprietary components. Some system level components (primarily those developed by academic partners) with open access policy are available on https://github.com/allscale. Within this installation, several projects are maintained, each covering a specific component of the project and managed by the corresponding work package / task leader. Secondary to this, some components were developed outside the AllScale project and reside on GitHub repositories accessible to the CI system. Figure 3 presents an overview of the source control structure.

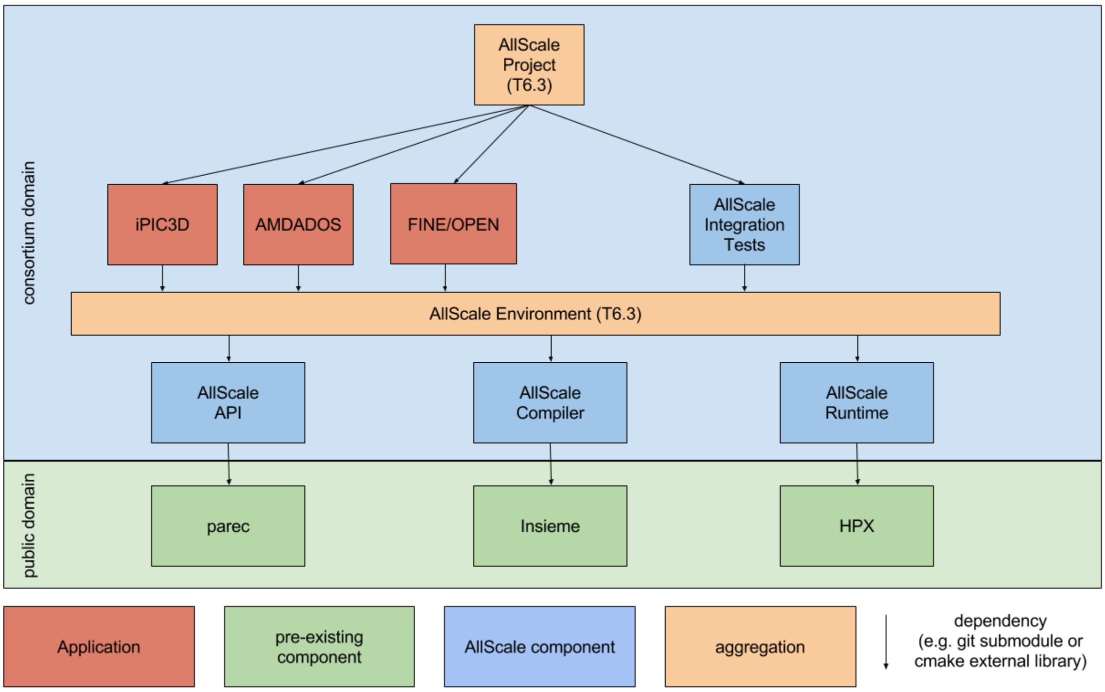

Figure 3: Source code repository structure

The source code repository structure consists of a number of different components:

1) Software components (rectangles in the figure) are git repositories, dependencies between them are implemented as git submodules.
2) Pre-existing components are available in the public domain (e.g. github) and will remain so; AllScale components, aggregation repositories and application repositories are available to all AllScale consortium members either via GitLab repositories for codes with proprietary aspects or via GitLab for codes with open access policy.
3) Aggregation packages will combine other components and add little-to-nothing new content on their own. Their main purpose is to allow a single, consistent version of the AllScale environment and the entire project.
4) The pilot applications only reference the "AllScale Environment" package, facilitating referencing specific versions and ensuring a consistent and stable system.
5) All components, including the pilot applications, are referenced by a common top-level package ("AllScale Project" in the figure), and must be available to all members participating in the development. If any component or application is not available in the proper version publically or on the protected repository server, its compatibility with other AllScale components cannot be ensured by individual developers.

The components included in Figure 3 consist of:

- The green packages (parec, Insieme, HPX and potential Monitoring tools) will be obtained from external sources and are expected to be maintained and tested externally.
- The blue packages are contributions developed for the AllScale infrastructure and its evaluation. Those will be managed, developed and tested by the corresponding work-package/task leaders.

- The orange packages are managed by task T6.3 for the integration of the AllScale components and to provide a consistent view on the infrastructure to the pilots, CI system and externals.
- The red packages are managed, developed and tested by the pilot groups and combined and tested with the other AllScale components on the CI system (by the AllScale project package).

3 Summary

The Installation, Integration and Deployment AllScale Environment is capable of fostering testing, development and validation among the consortium partners. All the AllScale pilots are, as per M16, deployed on that and are being benchmarked and tested within it with the AllScale software stack. This works towards the M18 deliverable of prototypes of the AllScale pilots integrated with the AllScale API (D6.2, 6.4 and 6.6 representing the iPIC3D, Fine/Open and AMDADOS applications respectively).