

DDS: The Dynamic Deployment System

A. Manafov¹ and A. Lebedev¹

¹GSI, Darmstadt, Germany

The Dynamic Deployment System (DDS) [1] is a tool-set that automates and dramatically simplifies the process of deployment of user defined processes with their dependencies on any resource management system using a given topology.

DDS is a successor of PoD [2]. Unlike PoD, which automates PROOF [3] deployment, DDS will handle any kind of user processes with complex dependencies between processes. The system is designed and being implemented within the new ALFA framework [4].

Concept

A key point of this design is the so called “topology language”. DDS is a user oriented system, i.e. definition of topologies is simple and powerful at the same time. The basic building block of the system is a task. Namely, a task is a user defined executable or a shell script, which will be deployed and executed by DDS on a Resource Management System. To describe dependencies between tasks in a topology we use properties. A property is represented as a key-value pair, where a value is any string. DDS implements an efficient engine for properties synchronization. We call it a key-value propagation feature.

Tasks can be grouped into DDS collections and DDS groups. The difference between collections and groups is that collections are a signal to DDS topology parser that tasks of given collections will be executed on the same physical machine. This is useful if tasks have lots of communication or they want to access the same shared memory. A set of tasks and task collections can be also grouped into task groups.

DDS utilizes a plug-in system in order to use different job submission front-ends. The first and the main plug-in of the system is an SSH plug-in, which can be used to dynamically turn a bunch of machines to user worker nodes. The SSH plug-in is a perfect tool for a Cloud based solutions.

Status

During 2014 the core and main modules of the DDS have been developed and the first stable prototype has been released.

The prototype has been tested on Alice HLT cluster using 40 computing nodes with 32 processes per node. DDS’s SSH plugin has been used to successfully distribute and manage 1281 AliceO2 user tasks (1 sampler, 640 FLP and 640 EPN [5]). Throughout the test, one DDS commander

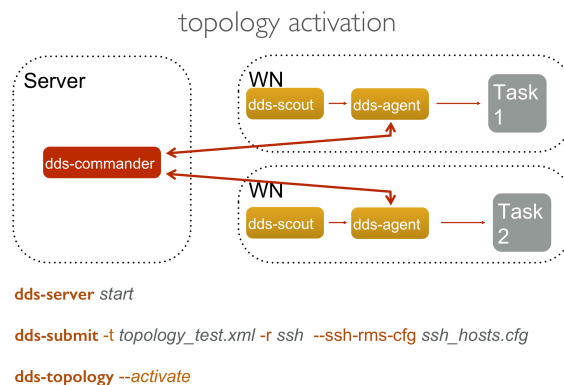


Figure 1: Illustration of the DDS topology activation.

server has propagated more than 1.5M key-value properties in less than 30 s. Which is a great performance considering early stage of the product development.

Development

DDS is being actively developed using modern development tools, C++11, and Boost libraries [6]. As the continuous integration framework we use BuildBot [7]. The web site and DDS’s users manual are based on DocBook [8]. We developed and maintain a unique Git workflow to simplify and secure the development [9].

References

- [1] Dynamic Deployment System (DDS), <http://dds.gsi.de>.
- [2] A. Manafov et al, “PROOF on Demand”, IT-07, GSI Scientific Report 2012.
- [3] TheParallelROOTFacility(PROOF), <http://root.cern.ch/drupal/content/proof>
- [4] M. Al-Turany et al., Status of the FairRoot framework, this report.
- [5] A. Rybalchenko et al., Evolution and use cases of FairMQ, this report.
- [6] BOOST C++ Libraries, <http://www.boost.org>.
- [7] Buildbot. The Continuous Integration Framework, <http://www.buildbot.net>.
- [8] DocBook, <http://www.docbook.org>.
- [9] Git Workflow, <https://github.com/AnarManafov/GitWorkflow>.