



Contribution ID: 241

Type: **Poster presentation**

ATLAS software configuration and build tool optimisation

Monday, 14 October 2013 15:00 (45 minutes)

The ATLAS software code base is over 7 million lines organised in about 2000 packages. It makes use of some 100 external software packages, is developed by more than 400 developers and used by more than 2500 physicists from over 200 universities and laboratories in 6 continents. To meet the challenge of configuration and building of this software, the Configuration Management Tool (CMT) is used. CMT expects each package to describe its build targets, build and environment setup parameters, dependencies on other packages in a text file called requirements, and each project (group of packages) to describe its policies and dependencies on other projects in a text project file. Based on the effective set of configuration parameters read from the requirements files of dependent packages and project files, CMT commands build the packages, generate the environment for their use, or query the packages.

The main focus was on build time performance that was optimised within several approaches:

- reduction of the number of reads of requirements files that are now read once per package by a CMT build command that generates cached requirements files for subsequent CMT build commands;
- introduction of more package level build parallelism, i.e., dependent applications and libraries are compiled in parallel;
- code optimisation of CMT commands used for build;
- introduction of project level build parallelism, i.e., parallelise the build of independent packages.

By default, CMT launches NUMBER-OF-PROCESSORS build commands in parallel. The other focus was on CMT commands optimisation in general that made them about 2 times faster.

CMT can generate a cached requirements file for the environment setup command, which is especially useful for deployment on distributed file systems like AFS or CERN VMFS.

The use of parallelism, caching and code optimisation significantly - by several times - reduced software build time, environment setup time, increased the efficiency of multi-core computing resources utilisation, and considerably improved software developer and user experience.

Primary author: RYBKIN, Grigori (Universite de Paris-Sud 11 (FR))

Presenter: RYBKIN, Grigori (Universite de Paris-Sud 11 (FR))

Session Classification: Poster presentations

Track Classification: Software Engineering, Parallelism & Multi-Core