Benchmarking the ATLAS software through the Kit Validation engine

Alessandro De Salvo[1], Franco Brasolin[2]

[1] Istituto Nazionale di Fisica Nucleare, Sezione di Roma,
[2] Istituto Nazionale di Fisica Nucleare, Sezione di Bologna

[1]Alessandro.DeSalvo@roma1.infn.it, [2]Franco.Brasolin@bo.infn.it

Abstract. The measurement of the experiment software performance is a very important metric in order to choose the most effective resources to be used and to discover the bottlenecks of the code implementation. In this work we present the benchmark techniques used to measure the ATLAS software performance through the ATLAS offline testing engine Kit Validation and the online portal Global Kit Validation. The performance measurements, the data collection, the online analysis and display of the results will be presented. The results of the measurement on different platforms and architectures will be shown, giving a full report on the CPU power and memory consumption of the Monte Carlo generation, simulation, digitization and reconstruction of the most CPU-intensive channels. The impact of the multi-core computing on the ATLAS software performance will also be presented, comparing the behavior of different architectures when increasing the number of concurrent processes. The benchmark techniques described in this paper have been used in the HEPiX group since the beginning of 2008 to help defining the performance metrics for the High Energy Physics applications, based on the real experiment software.

1. Introduction

In preparation to the LHC data taking all the computing centres must be ready to cope with the massive CPU and storage requests. An important aspect will be the acquisition of the most appropriate resources for every LHC experiment in which they participate. On the other hand the experiment code must be very efficient to avoid wasting time and memory on the computing centres, while keeping the same behavior in all the sites. The Benchmarking system of ATLAS, based on the KitValidation engine, is addressing all these issues, offering a comprehensive platform for performance and code efficiency measurements of the ATLAS software on different hardware, thus publishing very important information to all the Collaboration and the associated computing centres.

2. The architecture of the KitValidation Benchmark System

The ATLAS software is based on the Athena software framework and provides a modular structure for several tasks, including:

- event generation;
- event simulation and digitization;
- event trigger;
- event reconstruction;
- physics analysis tools.

Athena is a control framework and is a concrete implementation of an underlying architecture called Gaudi. The Gaudi project was originally developed by LHCb. Nowadays the Gaudi project is a kernel of software common to both experiments and co-developed, while Athena is the sum of this kernel plus ATLAS-specific enhancements.

The ATLAS software releases [1] are created at CERN and made available to the users both in AFS and as pacman [2] installation packages. Currently a full installation of an ATLAS software release uses about 8 GB of local disk space.

Software called KitValidation (*KV*) is used to validate the installed releases in a site, checking the functionalities of the installed packages, before using them. In order to fully validate the software, a comprehensive test of the major components of each release is performed, as described in this paper. The timings and performance data obtained from these representative tests is then used to extract the benchmarks of the same software on different machines and architectures. The benchmark data are then collected in a web portal, called Global Kit Validation (GKV), from where any authorized user can retrieve the measured values and compare different type of machines.

2.1. Kit Validation (*KV*)

KitValidation is a driver for tests. The test definitions are written in XML and are either contained in the software releases or provided via an external source, like http URLs or plain files.

The base KV code is developed in shell (bash) scripting language and is using python to parse the XML definitions of the tests and to send data to the web portal GKV. The test definitions to perform a full validation of a software release are generally contained in the packages of the release itself, and consist of basic tasks, created to perform a full chain test of the installed ATLAS software. The involved logical steps are:

- event generation;
- event simulation;
- data digitization;
- data reconstruction.

A KitValidation session is generally self-consistent, i.e. the input data needed by a test is produced by the previous test in the sequence, if needed. For example the simulated hits produced in the simulation test are used as input to the reconstruction test.

Custom tests can also be used by providing their XML definitions in a file, downloadable via web or stored locally in the node.

The typical amount of time needed for a full benchmark session, producing 100 tt events on recent CPUs, is about 100 minutes, and the amount of disk space needed for the temporary files is < 1 GB.

The logfiles and the exit codes of each test are sent to the GKV portal. The data is transferred using curl or an embedded python posting engine via http or https. Both http and https proxies are supported.

KV is used by the whole ATLAS collaboration for both standard and Grid software installations, in order to validate the local setup, providing an easy way to compare machine performance, in terms of

CPU and memory. KV is in use since 2003, and about 203,000 validations have been performed so far, testing ~158 different CPU types (32/64 bit) and ~160 ATLAS software releases.

2.2. Global Kit Validation (GKV)

The Global Kit Validation portal[1] is a web interface with search capabilities, based on *MySQL5* and *PHP5*.

The portal receives data from KitValidation, when the data sending is enabled by the user, collecting different kind of information, and in particular:

- CPU architecture and family;
- CPU speed;
- memory size.

Additionally, the logfiles and the exit codes of each test are collected in GKV.

The machine details and the full logfiles are made available to certified users only, authenticating them through their digital certificates. Normal users can still browse the results of the tests, but are not entitled to access the private data of the nodes.

The portal offers multiple search options to find the appropriate results:

- machine name;
- CPU type;
- custom string (tag), provided by the user when running the tests;
- software versions;
- username used when performing the tests;
- OS type and version.

The benchmarks results are not displayed by default, but can be shown by selecting the appropriate option in the initial search form of the portal.

2.3. Running KitValidation in Benchmark mode

KitValidation is run in Benchmark mode via an external tool, sw-mgr. This tool is also used to deploy and validate the ATLAS software releases at the Grid sites, and is part of the LCG/OSG Installation System [3][4].

By using sw-mgr over KitValidation some important features are added:

- automatic installation of the software release to be used in the benchmarks;
- possibility to run multiple parallel threads of tests at the same time;
- single-line command to validate a release.

In particular, the multiple thread feature is very important to run full-load tests on a machine, as required by the HEPiX benchmarks.

By default the KV benchmarks are provided in 3 different flavors, with increasing complexity of the involved software components and number of events, thus affecting the total amount of time needed to complete the sessions. The user can choose which benchmark suite to run, depending on the target of the test. All the default test definitions are kept in a XML file[2], downloadable from the KV web site.

[1] https://kv.roma1.infn.it/KV

[2] https://kv.roma1.infn.it/KV/KV_benchmark.xml

A driver script to perform a full benchmark test, including the installation of the required software release, is also provided from the main KV site[3]. This script provides a default tag to associate to the current benchmark session, built from the characteristics of the local machine. By default, a reference release (14.1.0) is installed and used for the benchmarks, but the user may override it and use the same script for any ATLAS release later than version 8.0.0.

2.4. Data Processing

The timings and performance measurements are calculated by parsing the log files, using the PerfMon ATLAS facility [5].

The log files are scanned by GKV the first time a user requests a benchmark result for a given setup. The calculated values are then stored into the GKV database for further retrieval. The following requests of the same Benchmark results are obtained from the GKV database.

Using PerfMon it is possible to calculate important metrics like:

- total test running time;
- number of events per second on single threads;
- average number of events per box at full load.

The results are calculated for each test. When the number of parallel threads available is not equal to the number of cores available in the box, the results are extrapolated to simulate a machine at full load. In case the number of threads matches the number of cores available in the machine, no extrapolation is performed and the benchmark results are directly used for the full load measurements.

2.5. Online Presenter

The Global Kit Validation portal provides an online presenter for the Benchmark results. By using the search form and selecting the benchmark option, the portal can dynamically generate comparison tables and charts for CPU and memory occupancy.

The results of the same test on the same machine are aggregated together in the tables and used to compute the average values. The CPU and memory charts are generated using the aggregated values.

The computed results are used to compare different combinations of the entities, for example:

- performance of a given release on different hardware (CPU power benchmark);
- different releases on the same hardware (code efficiency);
- memory occupancy for the same task in different releases (memory allocation efficiency).

All the users can browse the benchmark results, but only authorized users can view the full machine and test details. The authorization of the users is performed thorough digital certificates. All users having a valid certificate, issued by an LCG-trusted CA, are allowed to access the generic benchmark results.

3. Results

3.1. HEPiX results

The authors represent the ATLAS collaboration in the HEPiX benchmarking group since 2008.
The ATLAS results have been used to define a new CPU benchmark unit for the WLCG collaboration in early 2009.

[3] https://kv.roma1.infn.it/KV/atlas-benchmark

In Figure 1, 2 and 3 the KV benchmark results vs the HEP SPEC 06 [6], as computed by the HEPiX group, are shown. The benchmarks are performed generating 10,000 events and simulating, digitizing and reconstructing 100 events. The results[4] show a good linear scaling with the HEP SPEC 06 values.

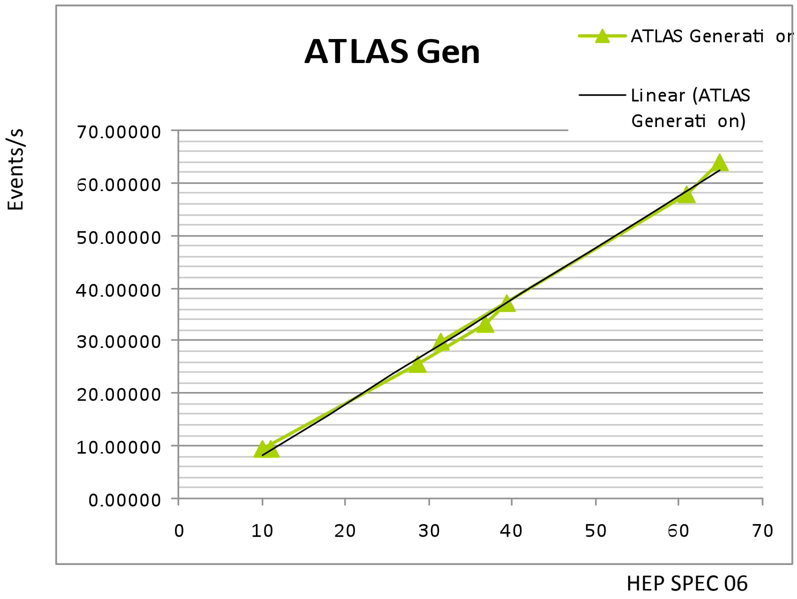

Figure 1 – ATLAS benchmark results measured by KV, used by HEPiX. The plot show the results for the MC event generation.

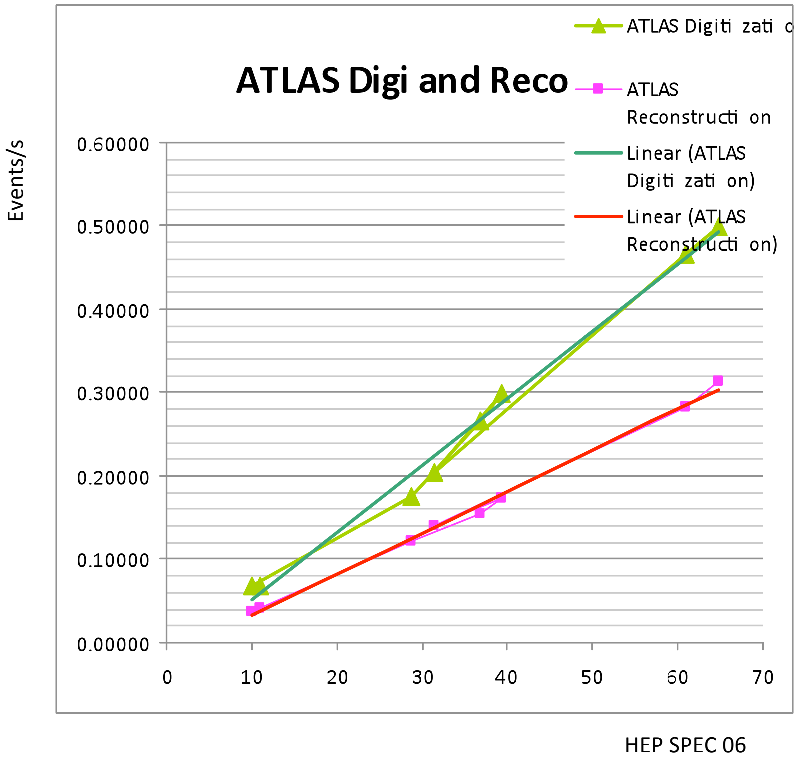

Figure 2 – ATLAS benchmark results for the MC simulation, measured by KV.

[4] Charts courtesy by M. Michelotto - INFN Sezione di Padova - HEPiX Benchmark Group

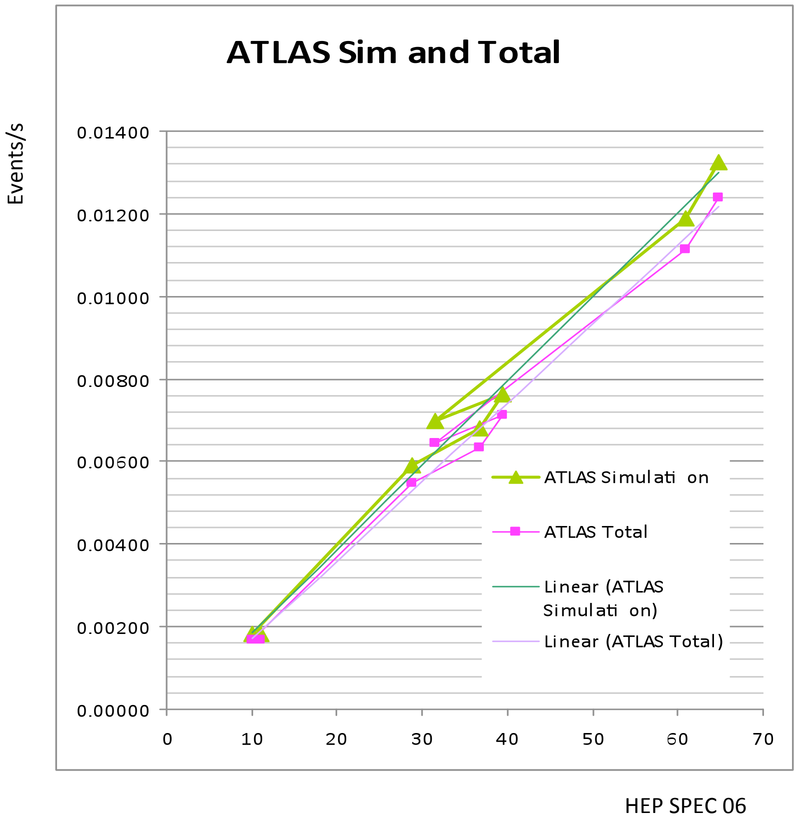

Figure 3 – ATLAS benchmark results measured by KV, used by HEPiX. The plot show the results for the full-chain event processing. The results show a good linearity in the number of events processed vs the HEP SPEC 06 benchmark values.

3.2. CPU and Memory Consumption comparison

GKV can dynamically produce charts and tables of results, to compare different processors and machines. The users can select the needed architecture and/or benchmark sessions to compare the values. The results of the comparison are available via the GKV Web Interface. Comparisons of different ATLAS SW releases is also possible, in order to show the difference in performance and to identify code inefficiencies.

The Memory consumption of each task, as obtained from PerfMon, is also stored in the GKV database and can be used for comparisons of different tasks in the same release or the same task in different releases. Both VM (Virtual Memory) and RSS (Resident Set Size) are taken into account making the comparison extremely useful to identify anomalous behavior of the code in terms of memory usage.

The VM and RSS values stored in the GKV database are the maximum values measured during each task.

In Figure 4 the memory and CPU consumption of two different ATLAS software releases, tested on the same box, are shown.

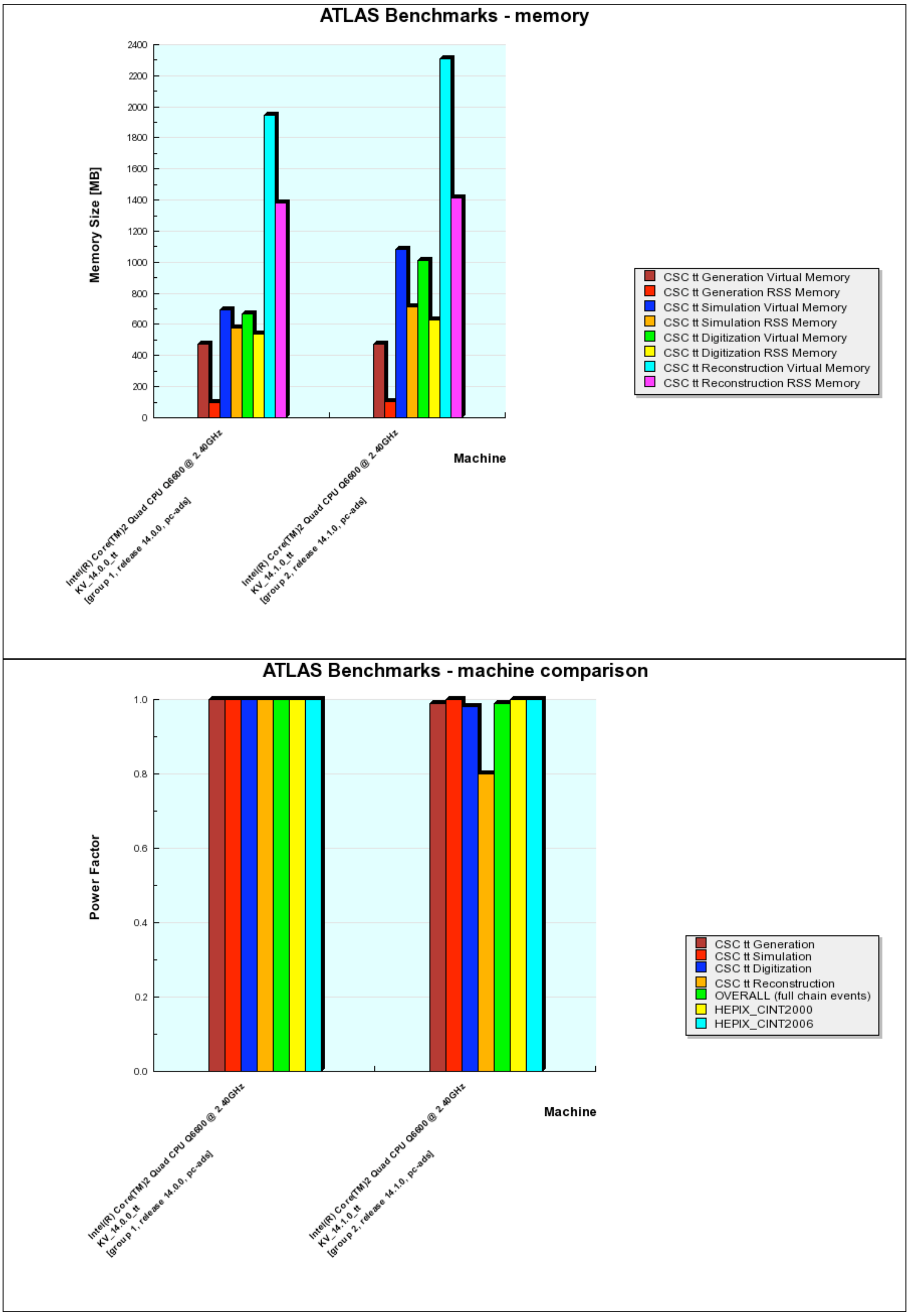

Figure 4 – Comparison of 2 different ATLAS software releases on the same hardware. The differences in both CPU performance and memory occupancy are shown in the plots.

4. Conclusions

Kit Validation has been used by the whole ATLAS collaboration since 2003 and more than 203,000 tests have been performed so far. The open architecture, the huge database of software performance measurements created so far and the advanced benchmarking features, make this tool very useful for both code testing purposes, processor benchmarks and software release comparisons.

All the users can perform benchmarks on new machines, thus incrementing the information of new processors available to the whole community, and we expect in the future that the ATLAS results will be very useful for the LHC computing centres to help in choosing the most appropriate resources to be acquired.

References

[1] Obreshkov E et al.
Organization and Management of ATLAS Software Releases,
ATL-SOFT-PUB-2006-008; CERN-ATL-SOFT-PUB-2006-008; ATL-COM-SOFT-2006-013, CERN.

[2] http://physics.bu.edu/~youssef/pacman/htmls/.

[3] De Salvo A. et al.
The ATLAS software installation system for LCG/EGEE,
International Conference on Computing in High Energy and Nuclear Physics (CHEP'07). Journal of Physics: Conference Series 119 (2008) 052013.

[4] Zhao X et al.
A Dynamic System for ATLAS Software Installation on OSG Grid site,
International Conference on Computing in High Energy and Nuclear Physics (CHEP'09).

[5] https://twiki.cern.ch/twiki/bin/view/Atlas/PerfMonComps.

[6] Michelotto M et al.,
A comparison of HEP code with SPEC benchmark on multicore worker nodes,
International Conference on Computing in High Energy and Nuclear Physics (CHEP'09).