University of A Coruña

Computer Architecture Group

Big Data Evaluator: User Guide

Authors:

Jorge Veiga, Roberto R. Expósito, Guillermo L. Taboada and
Juan Touriño

November 25, 2016

Contents

1 Overview

The Big Data Evaluator (BDEv) [1] is an evaluation tool to extract valuable information about the performance, scalability and resource efficiency of several Big Data frameworks. It allows to compare several solutions by means of different workloads, including micro-benchmarks and real-world applications.

BDEv uses multiple user-defined parameters to unify of the configuration the solutions, ensuring a fair comparison between them. In each experiment, the user can select the workloads and the solutions to be run. Several cluster sizes can be used in order to check the scalability of the frameworks and ensure an optimal use of the nodes of the system. The user can also determine the number of times each workload is executed in order to obtain statistical information.

This user guide aims to provide a clear explanation of the features available in BDEv, as long as to explain how to configure and run the evaluations.

2 Configuration of the experiments

The configuration of a experiment affects the following files:

- hostfile
- bdev-conf.sh
- system-conf.sh
- experiment-conf.sh
- core-conf.sh
- hdfs-conf.sh
- mapred-conf.sh
- yarn-conf.sh
- solutions-conf.sh
- solutions.lst
- benchmarks.lst
- cluster_sizes.lst

The environment variables and the configuration files are explained below, including the default values of the parameters.

Environment variables   There are two environment variables that BDEv uses to know where to find the configuration of the experiments. First, the `EXP_DIR` variable determines the directory that contains the configuration files mentioned above. This feature enables to set up different evaluations by means of several experiment directories. If this variable is not set, the value taken by default is `$BDEv_HOME/experiment`. Second, the `HOSTFILE` variable contains the path to the hostfile. If this variable is not set, the value taken by default is `$EXP_DIR/hostfile`.

```
export EXP_DIR=$BDEv_HOME/experiment
export HOSTFILE=$EXP_DIR/hostfile
```

---

[1] BDEv has evolved from MREv, a MapReduce Evaluation tool aimed to compare the performance of HPC-oriented MapReduce solutions. Apart from that kind of solutions, BDEv also evaluates new types of Big Data frameworks like Spark and Flink

hostfile This file lists the computing nodes that will be used in the experiments. The first line of the file will be the master, and the remaining lines will be the slaves. NOTE: the "localhost" host name can only be used if the evaluation is going to run on local, instead the user must specify the hostname that corresponds to the host.

bdev-con.sh This file contains some parameters that configure the behaviour of BDEv along the experiments. First, the `ENABLE_PLOT` parameter determines if BDEv will generate performance graphs of the execution of the workloads. Similarly, `ENABLE_STAT` is set to perform the stat recording when executing the workloads, and generating the graphs afterwards. `ENABLE_ILO` can be set in order to record power consumption measurements using the HP-iLO technology (requires the previous instalation of this technology). `DEFAULT_TIMEOUT` defines the maximum execution time of a workload used by default. Those which run above this limit will be killed, and the execution of the solution will be finished. Of course, this BDEv feature can be disabled by setting `DEFAULT_TIMEOUT` to 0. Finally, the directory where BDEv will write the results of the experiment can also be configured by using the `OUT_DIR` variable. If this variable is not set, the value taken by default is `$PWD/BDEv_OUT`. This file also contains some configuration parameters used when enabling the record of resource stat or HP-iLO measurements.

system-conf.sh This file contains the parameters related to the system where BDEv is being run. Some of them are automatically detected from the system, but can also be tuned by the user in order to maximize the leveraging of the system resources.

experiment-conf.sh This file sets the configuration of the benchmarks, including the problem size and additional parameters, as well as the number of times each one is executed. Moreover, it also contains the `METHOD_COMMAND` variable, which contains the action to run in batch mode during the command benchmark. Additionally, `METHOD_PREPARE_COMMAND` is called to set up the input datasets needed for `METHOD_COMMAND`. This enables to perform accurate performance and resource utilization monitoring, without taking into account the data generation or the copy to HDFS. This file also allows to configure specific timeouts for each benchmark.

core-conf.sh This file contains configuration parameters which are related to the core-site.xml file of Hadoop configuration.

hdfs-conf.sh This file contains configuration parameters which are related to the hdfs-site.xml file of Hadoop configuration.

mapred-conf.sh This file contains configuration parameters which are related to the mapred-site.xml file of Hadoop configuration.

yarn-conf.sh This file contains configuration parameters which are related to the yarn-site.xml file of Hadoop configuration.

solutions-conf.sh This file contains configuration parameters which are specific to each solution, as well as some variables for Apache Mahout and Apache Hive.

solutions.lst This file contains the solutions to be used in the experiment, specifying the framework, its version and the network interface to be configured.

benchmarks.lst This file contains the benchmarks to be used in the experiment.

cluster_sizes.lst This file contains the cluster sizes with which the user wants to run the experiments. Additionally, the cluster size can be set to the maximum number of nodes available.

3 Execution

The following command starts the experiments:

```
bash BDEv/bin/run.sh
```

4 Evaluation outcomes

The results from the execution will be found in the `$OUT_DIR` directory, having the structure shown in Figure 1.

4.1 Log & configuration

BDEv creates separate log and configuration directories for each framework and stores them at `{cluster_size}/{framework}`. For example, the configuration directory of Hadoop-2.7.2-IPoIB with 5 nodes is `report_BDEv_03_02_12-00-00/5/Hadoop-2.7.2-IPoIB/etc/hadoop` and its log directory is `report_BDEv_03_02_12-00-00/5/Hadoop-2.7.2-IPoIB/log`. Both directories can be used to check the configuration generated by BDEv and the execution of the workloads. Moreover, this feature enables to run simultaneous evaluations of the same framework using different configurations.

4.2 Performance

The performance results in terms of time are available in the `graphs` subdirectory. For example, for the Wordcount benchmark, they can be found in the `report_BDEv_03_02_12-00-00/graphs/wordcount.eps` file. For each cluster size, the graph depicts the average, maximum and minimum execution times taken by each framework to perform the workload.

4.3 Resource Utilization

The resource utilization results from the execution of a benchmark can be found at `{cluster_size}/{framework}/{benchmark}_{num_execution}/stat_records`. For example, the values of the first execution of Wordcount using Hadoop-2.7.2-IPoIB on 5 nodes are at `report_BDEv_03_02_12-00-00/5/Hadoop-2.7.2-IPoIB/wordcount_1/stat_records`. This directory contains one subdirectory for the values of each cluster node, plus another one for the average values among the slave nodes.

The resource utilization graphs are not automatically generated by BDEv in order to prevent the apparition of an excessively number of unnecessary files. The user can generate them by running the script `gen_graphs.eps`, which contains the command lines needed for generating the resource utilization graphs contained in that directory. These graphs include CPU utilization (`cpu_stat.eps`), CPU load (`cpu_load_stat.eps`), memory usage (`mem_stat.eps`), disk read/write (`dsk_sda_rw_stat.eps`), disk utilization (`dsk_sda_util_stat.eps`) and network send/recv (`net_eth1_stat.eps`, `net_ib0_stat.eps`). Disks (sda) and network interfaces (eth1, ib0) are automatically detected by BDEv. For some resources, like CPU utilization, there are different visualization modes that allow to see the results individually (with lines, `cpu_stat.eps`) or as a whole (with stacked values, `cpu_stat_stacked.eps`).
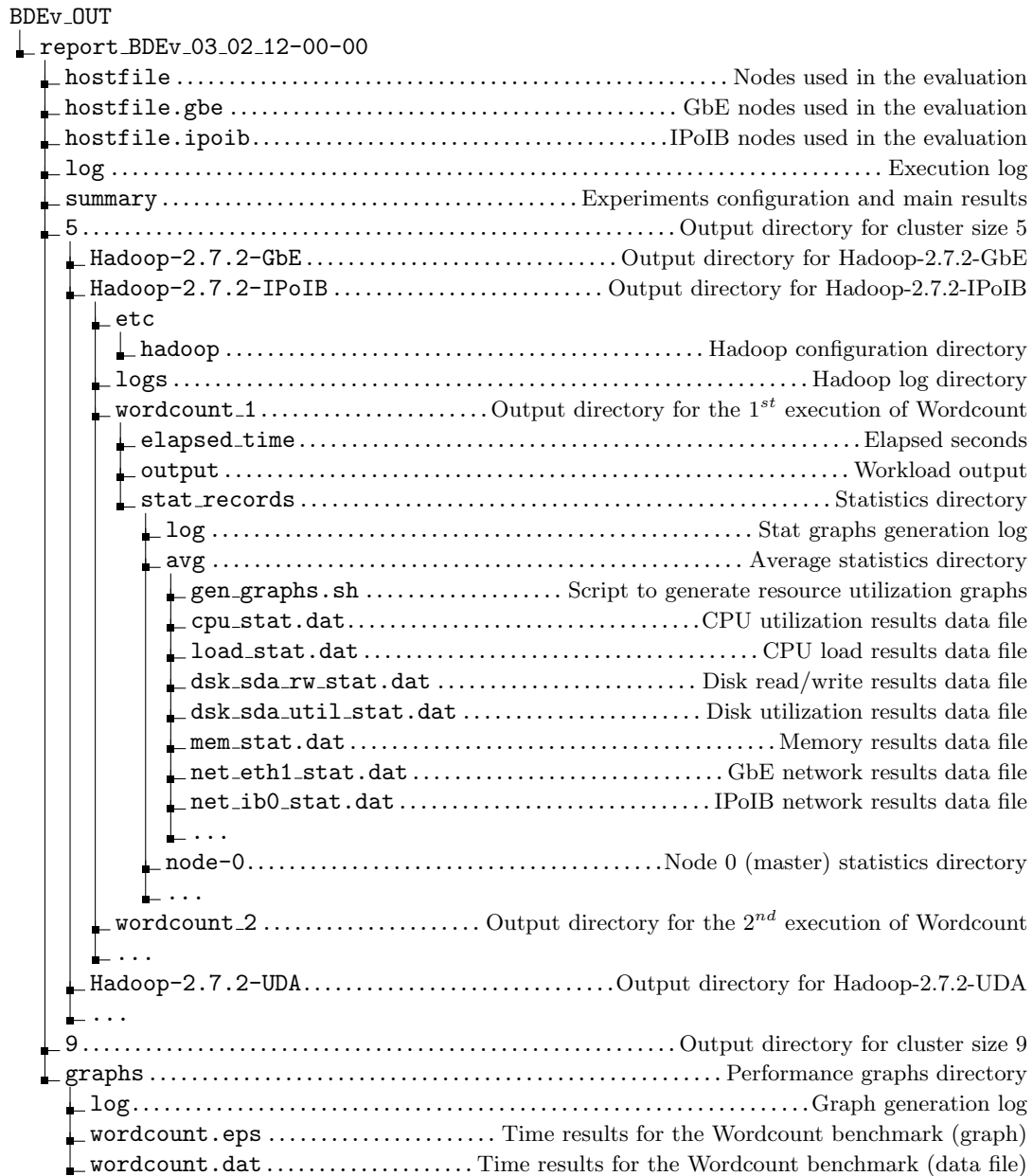
```
BDEv_OUT
└ report_BDEv_03_02_12-00-00
  ├ hostfile...................................................Nodes used in the evaluation
  ├ hostfile.gbe..........................................GbE nodes used in the evaluation
  ├ hostfile.ipoib......................................IPoIB nodes used in the evaluation
  ├ log................................................................Execution log
  ├ summary......................................Experiments configuration and main results
  ├ 5....................................................Output directory for cluster size 5
  │ ├ Hadoop-2.7.2-GbE............................Output directory for Hadoop-2.7.2-GbE
  │ ├ Hadoop-2.7.2-IPoIB.........................Output directory for Hadoop-2.7.2-IPoIB
  │ │ ├ etc
  │ │ │ └ hadoop..........................................Hadoop configuration directory
  │ │ ├ logs.............................................................Hadoop log directory
  │ │ ├ wordcount_1.....................Output directory for the 1ˢᵗ execution of Wordcount
  │ │ │ ├ elapsed_time...................................................Elapsed seconds
  │ │ │ ├ output...........................................................Workload output
  │ │ │ ├ stat_records...............................................Statistics directory
  │ │ │ │ ├ log..................................................Stat graphs generation log
  │ │ │ │ ├ avg................................................. Average statistics directory
  │ │ │ │ │ ├ gen_graphs.sh...................Script to generate resource utilization graphs
  │ │ │ │ │ ├ cpu_stat.dat..................................CPU utilization results data file
  │ │ │ │ │ ├ load_stat.dat......................................CPU load results data file
  │ │ │ │ │ ├ dsk_sda_rw_stat.dat..........................Disk read/write results data file
  │ │ │ │ │ ├ dsk_sda_util_stat.dat.......................Disk utilization results data file
  │ │ │ │ │ ├ mem_stat.dat.........................................Memory results data file
  │ │ │ │ │ ├ net_eth1_stat.dat..............................GbE network results data file
  │ │ │ │ │ ├ net_ib0_stat.dat..............................IPoIB network results data file
  │ │ │ │ │ └ ...
  │ │ │ │ ├ node-0.......................................Node 0 (master) statistics directory
  │ │ │ │ └ ...
  │ │ ├ wordcount_2.....................Output directory for the 2ⁿᵈ execution of Wordcount
  │ │ └ ...
  │ ├ Hadoop-2.7.2-UDA..............................Output directory for Hadoop-2.7.2-UDA
  │ └ ...
  ├ 9....................................................Output directory for cluster size 9
  ├ graphs..................................................Performance graphs directory
  │ ├ log..............................................................Graph generation log
  │ ├ wordcount.eps......................Time results for the Wordcount benchmark (graph)
  │ └ wordcount.dat....................Time results for the Wordcount benchmark (data file)
```

Figure 1: BDEv output directory structure

A   About Open Grid Scheduler/Grid Engine

As most supercomputers use a batch-queuing system for distributed resource management, BDEv is aware of the environment variables and connection constraints that are typically present in these cases. The correct behaviour of BDEv under this kind of systems has been tested with the Open Grid Scheduler/GE (OGS/GE).

BDEv will detect the `PE_HOSTFILE` environment and use it to read the compute nodes. In this case, no `HOSTFILE` variable will be needed, although it can also be set. Moreover, the ssh connections used to launch the different framework daemons do not work properly under OGS/GE, so BDEv performs a light modification to enable their execution.

B   About Environment Modules

BDEv is aware of the use of Modules for dynamically modifying the user's environment. If enabled in the configuration, BDEv will use it for loading the Java and MPI environment variables.

C   System Requirements

The following packages need to be installed:

- Expect 1.1 (needed if timeout is enabled)

- Java JRE 1.7 (always needed)

- Gnuplot 4.4 (needed for generating the graphs)

- MPI [2] (needed for DataMPI)

---

[2] DataMPI has been tested using Mvapich2