INFLUENCE OF YARN SCHEDULERS ON POWER CONSUMPTION AND PROCESSING TIME FOR VARIOUS BIG DATA BENCHMARKS

KRZYSZTOF DRYPCZEWSKI[1], JERZY PROFICZ[2] AND ANDRZEJ STEPNOWSKI[1]

[1] *Faculty of Electronics, Telecommunications and Informatics*
Gdansk University of Technology
Narutowicza 11/12, 80–233, Gdańsk, Poland

[2] *Centre of Informatics – Tricity Academic Supercomputer & Network*
Gdansk University of Technology
Narutowicza 11/12, 80–233, Gdańsk, Poland

(received: 20 July 2018; revised: 22 August 2018;
accepted: 19 September 2018; published online: 4 October 2018)

Abstract: Climate change caused by human activities can influence the lives of everybody on the planet. The environmental concerns must be taken into consideration by all fields of study including ICT. Green Computing aims to reduce negative effects of IT on the environment while, at the same time, maintaining all of the possible benefits it provides. Several Big Data platforms like Apache Spark or YARN have become widely used in analytics and High-Performance Computing systems due to the reliability and usability of Map Reduce implementations. The authors research the power consumption and energy efficiency of Hadoop YARN schedulers using Apache Spark under three different workloads. The test cases include: sorting large binary files, counting unique words in large text files and processing satellite imagery from the Sentinel-2 mission. The presented results show small (2%–11%) but distinct differences in the power consumption of FIFO and FAIR schedulers.

Keywords: Apache Spark, YARN, Big Data, Green Computing, Sentinel, Tera Sort, word count, benchmarks, scheduler

DOI: https://doi.org/10.17466/tq2018/22.4/c

1. Introduction

Taking care of our planet and its environment is one of the biggest challenges facing humanity today [1]. Climate change has received a lot of scientific focus in last years and its causes are understood much better than in the previous

decade. Slow growth of the average temperature of the Earth's atmosphere and its possible influence on the global economy and everyday lives of everybody makes it a crucial problem. The ICT community takes part in fighting climate change already now [2]. This endeavor is known as Green Computing which brings together multiple fields of study and aims to: minimize the use of hazardous and environmentally unsafe elements, biodegradability of materials, recycling and improvement of energy efficiency [3].

Big Data is a term used to describe large sets of digital information. Processing this kind of data provides multiple challenges especially in the context of resources required for its analysis and storage. The Big Data characteristics, often described as "5V" are as follows [4]: (i) Variety: data can be structured, semi-structured and unstructured; data sets can contain multiple types of information, (ii) Velocity: fast growth of the data size, also rapid changes in data, (iii) Volume: the amount of data to be stored, processed, analyzed and disseminated, (iv) Veracity: precision of data, the ability to "trust" the information contained in the data set, (v) Value: the market value of the data.

The Big Data technologies have evolved greatly due to the rapid growth of the Internet, cheap and easily available hardware and popularity of real-time data streaming (sensors, camera feeds *etc.*). Multiple platforms (such as Apache Hadoop [5] or Apache Spark [6] *etc.*) that are able to ease the Big Data processing by distributing both data and computations between the nodes in the cluster have been proposed and widely adopted by the relevant community.

From the point of view of Big Data researchers, Green Computing can be described (in a general sense) as the usage of computers in a way that provides maximum possible benefits (computing power, performance, calculation speed) while being environmentally friendly and sustainable. It is worth mentioning that Green Computing techniques can match the company or organization business goals. Lowering the energy usage of a data cluster without considerably reducing the overall performance can lead to a decrease in the operational cost.

The main contribution of the paper is analysis of the energy consumption of Apache YARN FIFO and FAIR schedulers under different workloads. Three Big Data benchmarks are presented: sorting large binary files, counting unique words found in texts in a digital library and processing large amounts of satellite data. Each of the benchmarks (dataset, algorithm *etc.*) is examined in detail and then the results are presented and discussed. In the last section, the authors present possible future works on developing a new, improved YARN scheduler that will take into account the overall cluster energy usage while scheduling jobs and managing resources.

2. Related works

Almeida *et al.* claim that [7] one of the biggest challenges in providing ultra-scale HPC systems is access to information on energy consumption. Each component in the computer stack should be thoroughly tested in the context

of possible and existing power inefficiencies. Nonetheless, as for now, there are no widely agreed and accepted power efficiency management standards. The available tools lack appropriate capabilities or, if these are available, they offer different functionalities that are difficult to compare, which makes it hard or even impossible to provide cross-platform statistics and analysis. The authors perform a survey of most often used energy measurement techniques, pointing out that most of those tools are just software developed for administrative purposes and they have not been designed for power consumption analysis.

Czarnul *et al.* in [8–10] proposed a solution for modeling the cluster behavior, including its electric power level and energy consumption. Their MERPSYS simulator enables extensive tests of the existing and even future cluster and grid solutions, thus proving the way to check possible hardware configurations and their evaluation according to the user requirements.

Appuswamy *et al.* [11] compare the performance of scale out (many commodity hosts) and scale up (smaller number of high-end machines) clusters. In the article, the authors point out that the statistics of performance (normalized) by watt is a very important metric for data centers. As the power draw serves as a limiting factor for data-centers, the power budget (cost of calculations) needs to be taken into account during the cluster design. The results of the performed test runs suggest that scale-up clusters are much more energy efficient, and this outcome was observed for all checked workloads.

Schall & Härder [12] focus on the concept of the energy proportionality which is a rate of the power usage to the amount of work (processing, *etc.*) done. The authors checked if adjusting the number of active servers in a cluster to the current workload could improve the overall energy efficiency. The article presents a Database Management System with an energy-aware scheduler, so called WattDB. The idea is to provide a DBMS that works on a set of lightweight nodes which are dynamically turned on and off, depending on the current load. The WattDB approach reduces power consumption in comparison with using only high-end hardware that consumes much more energy even when idle. Schall & Hudlet [13] describe a WattDB demo that provides a graphical user interface (GUI) for a dynamically adjusted cluster.

Haidar *et al.* [14] also write about the importance of power consumption of data centers, especially for peta and exa-scale clusters. The authors analyze the energy efficiency of the Intel Xeon Phi Knights Landing architecture and search for a way of developing power-aware algorithms. Power Application Programming API (PAPI) [15] is a library that allows access to the processor's hardware counters. A new concept is introduced to the PAPI: the power management tool called powercap. Powercap lets the user set the limits of power usage for the hardware that supports this functionality and then allows monitoring the influence of the "powercap" on the hardware. This functionality is provided by Intel RAPL – Running Average Power Limit. The authors give the power consumption of the Xeon Phi processor using a selected set of benchmarks and under different kernels.

The results acquired indicate that it is possible to reduce the energy used while keeping the performance unaffected. It can be achieved by tailoring the processor booting models and the memory modes to the selected benchmark. Another analysis of power capping covering a wider collection of CPUs (Intel Phi, Ceon with server and mobile architectures) and emphasizing the possible energy savings is provided in [16].

To the best of the authors' knowledge, there are no studies focusing primarily on the energy efficiency of Apache Hadoop clusters. There are also no works considering Spark Standalone (Spark's own built-in resource manager [17]) or Spark on YARN clusters related to this subject.

3. Environment setup

3.1. Apache Yarn

Apache YARN is a data processing platform developed as the Apache Foundation Top Level Project. It is a central point of Apache Hadoop utilizing its core functionalities (core libraries), storage capabilities (HDFS) and processing utilities (MapReduce). YARN manages resources in a cluster and schedules jobs in a workload. Both those functionalities are split between the global Resource Manager (RM), the Node Manager (NM) located at each host and the Application Master (AM) assigned to each application. The Resource Manager is responsible for assigning resources to applications; the Node Manager monitors the usage of resources (CPU, RAM) and creates, modifies and deletes application containers. The Application Master cooperates with the Resource Manager to obtain resources and with the Node Manager to execute a task specified by a given workload [18, 19].

The YARN scheduler decides what resources will be allocated to a job as well as when and where it will be executed. There are three scheduling algorithms provided in Apache Hadoop: FIFO (First In First Out) allocates resources on the basis of the job arrival time, FAIR where jobs are assigned to predefined pools and applications in different pools will share the cluster resources fairly, the Capacity Scheduler assigns jobs to different queues (*e.g.* per organization) and each queue is scheduled according to the FIFO rules [20].

YARN can serve as a resource manager for multiple applications including Apache Hadoop MapReduce and Apache Spark. YARN is used as the resource allocator for benchmarks described in the next sections.

3.2. Apache Spark

Apache Spark is an open platform for the processing of large data sets (Big Data). It enables efficient analysis and processing of data from a variety of sources, including block and object file systems. Apache Spark has the following features: high processing speed; ease of use – direct support for the Scala, Java, Python and R languages; generality – combining a variety of applications: SQL, machine learning, graphical modeling, data streaming and processing; portability – integration with Hadoop YARN, Apache Mesos, Kubernetes and the use of external data sources: HDFS, Swift, HBase, Cassandra.

The proposed benchmarks are developed for Apache Spark which in turn uses Apache YARN as the resource manager and the HDFS as a data source. Spark supports applications and appropriate processing routines written in Java, Python, R, or Scala [5, 6]. It was Scala that was used during the tests. Scala is both an object-oriented and functional language that allows the use of existing classes and libraries from Java due to its operation on the Java Virtual Machine (JVM). It also greatly simplifies functional programming by supporting stateless and state-full functions, with extensive use of mechanisms such as (tail) recursion and parallel processing.

3.3. HDFS

The Hadoop Distributed Files System (HDFS) is a Java based distributed file system designed to run on commodity hardware. The HDFS is a part of the Apahe Hadoop project providing cost-effective and easily scalable big data storage with hierarchal file organization. It supports fault tolerance, throughput and can easily store large datasets. HDFS has a mechanism that protects against hardware failure. It is based on redundancy by data replication. Moreover, it provides easier data streaming and since data is available on multiple nodes, instead of moving data, the resource manager (YARN) can move processing to the node, where the data is physically located [20].

The HDFS was implemented using the master-slave paradigm. The master, the so called Name Node (NN), manages file system namespaces and user access rights. In earlier HDFS versions there was only one Name Node instance running in the cluster, which made it a single point of failure. Since Hadoop version 2.4.1, the HDFS has provided tools [21] to overcome this by the possibility of starting a secondary Name Node in the stand-by mode. Each machine in the HDFS cluster runs a Data Node instance which is responsible for the data stored in the current host and communicates with the native file system. Data is split into blocks that, as instructed by the Name Node, are replicated between Data Nodes [20].

3.4. Big Data cluster

The Centre of Informatics – Tricity Academic Supercomputer & networK (CI TASK) provides a laboratory for the research on Green Computing. The laboratory is a separated part of the Triton supercomputer [22, 23] and consists of 16 servers; each of the nodes is equipped with Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30 GHz processor with 24 physical (48 logical HT) cores, 128 GB RAM (8×16 GiB DIMM Synchronous 2133 MHz), two 900 GB HDD managed by Logical Volume Management (LVM) and visible (through virtualization) by the operating system as one 1.7 TB disk. To provide redundancy and continuous availability, the machines are connected to two independent power distribution units (PDU) and the networking is supported by a 1 GB/s Ethernet switch.

The Simple Network Management Protocol (SNMP), a protocol designed for gathering information about devices in the computer network and modifying their settings, is used for energy measurement. PDUs and both server power adapters

support the SNMP measurement API. The master host queries them on a regular basis and the current power usage and time-stamp of measurement are recorded. Then the result is calculated and given in Joules.

4. Experimental description

4.1. Word count

Counting unique words in a given text is a classical algorithm used in benchmarking the Big Data platforms. Files are read one line at a time and each line is split into words. The text is converted into lowercase and then punctuation marks and number literals are removed. In the next step, a dictionary structure is created in which keys are the words and values are the number of occurrences of the key in the text (so called reduce-by-key operation). For example, in the sentence: "A cat caught a mouse" there are 4 unique words: "a", "cat", "caught", "mouse".

Project Gutenberg is an open-source initiative and the oldest digital library. Its aim is to provide ebooks in open format of books that were printed in paperback [24]. For the word count benchmark scenario, Project Gutenberg was selected for its freely available data set. 74887 books in text format were downloaded from the project's repository and copied to the HDFS. The entire dataset (without replication) is 30.1 GB in size.

4.2. Tera Sort

TeraSort is a widely used benchmark that measures the performance of Big Data platforms. In this benchmark, the time it takes a cluster to sort a randomly distributed binary data is measured. In recent years, TeraSort has been used to compare the performance of platforms such as Cloudera CDH, Apache Hadoop MapReduce or Apache Spark.

The TeraSort algorithm has three main steps: 1) random data generation, 2) data sort, 3) data validation (checking if the results are correct).

The TeraSort implementation based on the works of Reynold Xin (https://github.com/rxin/spark/tree/terasort) and Ewan Higgs (https://github.com/ehiggs/spark-terasort) is used in the performed tests.

4.3. Sentinel-2 Satellite data

Copernicus is a European Union Earth Observation (EO) program aimed at creating services based on the combination of satellite remote sensing and terrestrial sensor data. The European Space Agency has developed a Sentinel mission consisting of a pair of Sentinel 2 satellites for the needs of the Copernicus program. The Sentinel-2A and Sentinel-2B satellites were deployed in orbit in 2015 and 2017, respectively, and their main task is acquisition and transfer of digital imagery of the Earth's surface. Sentinel-2 provides 14 band imagery including spectral ranges such as visible light, infrared, *etc.* [22].

CI TASK maintains an archive [22] of satellite data from the two satellite missions: Sentinel-1 and Sentinel-2. The stored data covers the territory of the

Republic of Poland and the Southern Baltic. In the benchmark, Sentinel-2 data is processed to create RGB images of the Earth in a visual spectrum. 4, 3 and 2 band imagery, *i.e.*, corresponding to the RGB channels is downloaded from the HDFS and then the data is merged into one image and saved in the file system. The dataset for this benchmark consists of 300 Sentinel-2 Level 2 data products (207 GB). A more detailed description of this algorithm can be found in [22].

5. The results and discussion

The aim of the article is to test the influence of the YARN schedulers on various types of Big Data processing. In the previous section, three Big Data benchmarks (TeraSort, Sentinel, WordCount) are described. Each of the benchmarks was executed 25 times to provide more reliable results. Each of the test cases has different resource requirements (CPU, memory, disk operations) so the amount of energy used differs between the workloads.

As mentioned before, energy consumption is measured on power supplies and PDUs. The results from PDUs have higher values: the overhead arises from the additional energy used by the switch as well as from the loses caused by the power supplies. In the discussion below, we use measurements from PDUs as they seem to be more reliable. Tab. 1 presents average results of 25 benchmark iterations.

Table 1. Average benchmarks results for FIFO and FAIR schedulers

	FAIR				FIFO			
Benchmark	Energy avg. (MJ)	Energy std. (MJ)	Execution time avg. (s)	Execution time std. (s)	Energy avg. (MJ)	Energy std. (MJ)	Execution time avg. (s)	Execution time std. (s)
WordCount	11.298	0.487	2660.713	305.77	12.242	2.623	2660.713	306.072
Sentinel	4.037	0.056	985.367	37.121	3.639	0.101	985.367	36.953
TeraSort	34.109	0.627	10025.386	428.83	35.175	1.609	10025.386	428.854

Fig. 1 and Fig. 2 present data from Table 1. As Fig. 2 shows, both schedulers averaged very similar execution times for all benchmarks. However, as Fig. 1 shows there are small but distinct differences between energy consumption of FIFO and FAIR schedulers.

The results for WordCount and TeraSort benchmarks show that the FAIR scheduler requires between 1.8%–8.3% less energy than the FIFO scheduler. Interestingly, the FIFO scheduler is more energy efficient for the Sentinel benchmark (11%). Variation in measurements seems to be influenced by the type of the resource used for various test cases. Tab. 2 shows which resources are used heavily during benchmark execution. Processing satellite data requires the highest input and output (I/O) operation rates from the considered benchmarks and the results obtained for the Sentinel benchmark indicate that FIFO (in the Green Computing context) is better quipped for I/O intensive workloads. On the other hand,

CPU and RAM intensive workloads are better when managed by the FAIR scheduler. Additionally, the FAIR scheduler causes much lower variation in results in comparison to the FIFO scheduler (Tab. 1).

In general, the differences between performances of scheduling algorithms are not constant and fluctuate between 1% to 5% depending on the type of workload and resources used.

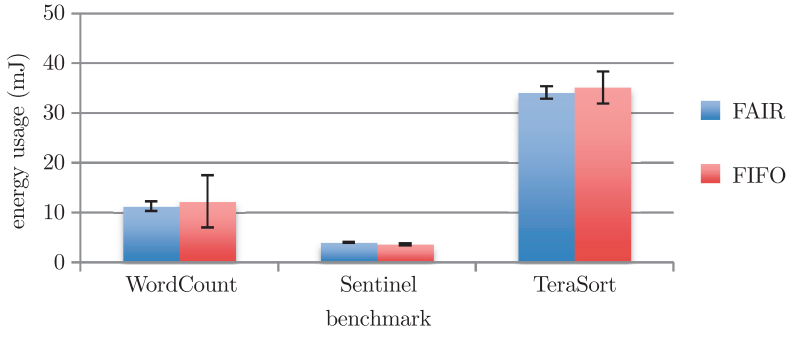

Figure 1. YARN schedulers average energy consumptions. The error bars are set to 2δ (95% of the measurements for the normal distribution)

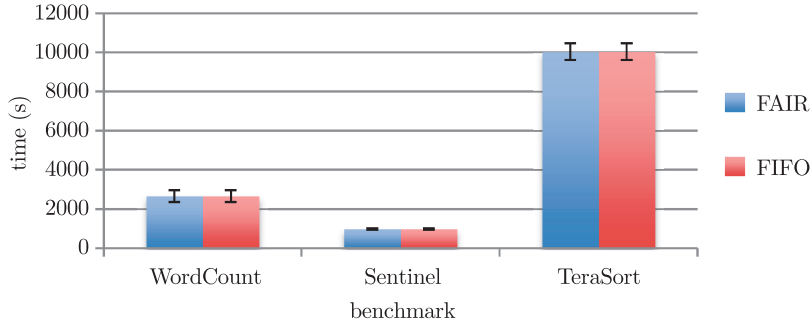

Figure 2. Average duration of processing. The error bars are set to 2δ (95% of the measurements for the normal distribution)

Table 2. Benchmark resource usage

Resource usage	CPU	RAM	I/O
WordCount		✓	
Sentinel	✓	✓	✓
TeraSort	✓	✓	

6. Final remarks

Due to the growing quantity of information in the current world, Big Data technologies are widely used in HPC solutions. Obviously, the main aim of the platforms such as Apache Spark or Apache Hadoop is to provide relevant benefits or market value for its users. Even though the IT community should think not

only about performance, optimization or speed of calculations but also about the environmental aspects. The influence of data-centers on the environment should be taken into consideration, especially in the context of power draw and energy efficiency.

Green Computing is an endeavor that searches for ways to providew energy sustainable computing. As shown by the results presented in the previous section, YARN schedulers have different energy efficiency depending on the given workload. I/O intensive workloads perform better in the context of energy used on the FIFO than FAIR scheduler. On the other hand, CPU and RAM intensive workloads, that did not make extensive use of I/O operations, achieve better results on the YARN FAIR scheduler. It can be inferred from these results that it is possible to design a scheduler that takes into account workload characteristics and manages jobs in such a way that there will be a reduction in energy consumption for a wide range of workloads. Thus, in the future, the authors plan to develop such a scheduler. Firstly, we will try to change the active scheduler depending on the history of workload energy consumption and in the next step we are going provide a machine learning solution that will adapt to the executed workload. We also plan to research the possibility of a scheduler that turns on and off machines in the cluster depending on the current load, similar to the solution proposed by Schall [12, 13].

References

[1] USGCRP 2017 *Climate Science Special Report: Fourth National Climate Assessment*, U. S. Global Change Research Program, Washington, **I** 470 doi: https://doi.org/10.7930/J0J964J6

[2] Biswajit S 2014 *Green Computing. International Journal of Computer Trends and Technology (IJCTT)* **14** (2)

[3] Molla A, Cooper V and Pittayachawan S 2009 *IT and Eco-sustainability: Developing and Validating a Green IT Readiness Model*, *ICIS 2009 Procs.*

[4] Pence H 2014 *Journal of Educational Technology Systems* **43** (2) 159 doi: https://doi.org/10.2190/ET.43.2.d

[5] Apache Hadoop Homepage [online] http://hadoop.apache.org/ [Accessed: 28-March-2018]

[6] Apache Spark Homepage [online] https://spark.apache.org/ [Accessed: 28-March-2018]

[7] Almeida F, Arteaga J, Blanco V and Cabrera A 2005 *Supercomputing Frontiers And Innovations* **2** (2) 64 doi: http://dx.doi.org/10.14529/jsfi150204

[8] Proficz J and Czarnul P 2016 *Performance and Power-Aware Modeling of MPI Applications for Cluster Computing*, in Wyrzykowski R, Deelman E, Dongarra J, Karczewski K, Kitowski J and Wiatr K (eds.), Parallel Processing and Applied Mathematics. Lecture Notes in Computer Science **9574**, Springer, Cham, https://link.springer.com/chapter/10.1007%2F978-3-319-32152-3_19

[9] Czarnul P, Kuchta J, Rościszewski P and Proficz J 2016 *Procs. 2016 Federated Conference on Computer Science and Information Systems* Ganzha M, Maciaszek L and Paprzycki M (eds.) **8** 855

[10] Czarnul P, Kuchta J, Matuszek M, Proficz J, Rościszewski P, Szymański J and Wójcik M 2017 *Simulation Modelling Practice and Theory* **77** 124 doi: 10.1016/j.simpat.2017.05.009

[11] Appuswamy R, Gkantsidis C, Narayanan D, Hodson O and Rowstron A 2013 *Scale-up vs scale-out for Hadoop: time to rethink?*, *SoCC'13, Santa Clara, California, USA. ACM 978-1-4503-2428-1* doi: http://dx.doi.org/10.1145/2523616.2523629
[12] Schall D, Hudlet V and Härder T 2010 *Procs. Third C* Conference on Computer Science and Software Engineering*, ACM, New York 1 doi: 10.1145/1822327.1822328
[13] Schall D and Hudlet V 2011 *WattDB: an energy-proportional cluster of wimpy nodes*, in Proceedings of the 2011 ACM SIGMOD International Conference on Management of data, SIGMOD'11 1229, publisher ACM
[14] Haidar A, Jagode H, YarKhan A, Vaccaro P, Tomov S and Dongarra J 2017 *Power-aware Computing: Measurement, Control, and Performance Analysis for Intel Xeon Phi*, IEEE High Performance Extreme Computing Conference (HPEC'17)
[15] PAPI homepage [online] http://icl.cs.utk.edu/papi/
[16] Krzywaniak A, Proficz J and Czarnul P 2018 *Analyzing energy/performance trade-offs with power capping for parallel applications on modern multi and many core processors* **15** 339 doi: https://doi.org/10.15439/2018f177
[17] Karau H 2013 *Fast data processing with spark*, Packt Publishing Ltd.
[18] Yarn and MapReduce Schedulers [online] https://www.cloudera.com/documentation/enterprise/5-8-x/topics/admin_schedulers.html [Accessed: 28-March-2018]
[19] Apache Hadoop YARN – ResourceManager, Vinod Kumar Vavilapalli, Hortonworks Homepage [online] https://hortonworks.com/blog/apache-hadoop-yarn-resourcemanager/ [Accessed: 24-Februrary-2018]
[20] HDFS Introduction [online] https://hortonworks.com/apache/hdfs/ [Accessed: 20-March-2018]
[21] HDFS design [online] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html [Accessed: 10-March-2018]
[22] Drypczewski K and Proficz J 2017 *TASK Quarterly* **21** (4) 365
[23] Krawczyk H, Nykiel M and Proficz J 2015 *Polish Maritime Research* **22** (3) 99 doi: https://doi.org/10.1515/pomr-2015-0062
[24] Hart M 2004 *Project Gutenberg Mission Statement* [online] https://www.gutenberg.org/wiki/Gutenberg:Project_Gutenberg_Mission_Statement_by_Michael_Hart [Accessed: 20-March-2018]