

Energy Consumption Models and Predictions for Large-scale Systems

Taghrid Samak*, Christine Morin†, David Bailey*

*Lawrence Berkeley National Laboratory, Berkeley, CA

†INRIA Rennes - Bretagne Atlantique

Abstract—Responsible, efficient and well-planned power consumption is becoming a necessity for monetary returns and scalability of computing infrastructures. While there is a variety of sources from which power data can be obtained, analyzing this data is an intrinsically hard task. In this paper, we propose a data analysis pipeline that can handle the large-scale collection of energy consumption logs, apply sophisticated modeling to enable accurate prediction, and evaluate the efficiency of the analysis approach. We present the analysis of a power consumption data set collected over a 6-month period from two clusters of the Grid’5000 experimentation platform used in production. To solve the large data challenge, we used Hadoop with Pig data processing to generate a summary of the data that provides basic statistical aggregations, over different time scales. The aggregate data is then analyzed as a time series using sophisticated modeling methods with *R* statistical software. Energy models from such large dataset can help in understanding the evolution of consumption patterns, predicting future energy trends, and providing basis for generalizing the energy models to similar large-scale systems.

I. INTRODUCTION

Responsible, efficient and well-planned power consumption is becoming a necessity, if not for ethical reasons, for pure monetary returns and scalability of computing infrastructure projects. Green computing is increasingly becoming the focus of many recent studies. The planning, analysis, and design of power-aware systems is an essential part of all current and future large scale computing infrastructure projects. While there are a variety of sources from which power data can be obtained, there are several ongoing projects that deal with coalescing and enriching the available measurements. Analyzing this data is an intrinsically hard task due to the heterogeneity of the sources, and lack of proper models.

This paper analyzes energy consumption data collected over six months of operation from two clusters of an experimentation platform used in production. Energy models from such large dataset can help in understanding the evolution of consumption patterns, predicting future energy trends, and providing basis for generalizing the energy models to similar large-scale systems. The analysis performed on this dataset needs to be efficient, accurate, and easily adaptable to new data. For achieving those goals, we provide a data analysis pipeline that can handle the large-scale collection of energy consumption logs, apply sophisticated modeling to enable accurate prediction, and evaluate the efficiency of the approach.

For the large data challenge, we used Hadoop with Pig data processing to generate a summary of the data that provides

basic statistical aggregations, over different time scales. The aggregated energy data includes basic statistics and outliers detection. The aggregate data is then analyzed as a time series using sophisticated modeling methods with *R* statistical software. This analysis process provides high-level visualizations of the consumption trends and outliers. The process also provides time series models with predictions of future samples. The performance of the Hadoop data processing and aggregation pipeline is evaluated on a 55 node cluster.

The remainder of this paper is organized as follows. In Section II, we present the Grid’5000 computing infrastructure from where the dataset analyzed was collected. We then introduce in Section III our approach for analyzing the power consumption dataset and background on time series modeling. The results and models and predictions derived from them are described in Section IV. We provide performance results in Section V. Related work is discussed in Section VI. In Section VII we summarize the lessons learnt from this study and give future work directions.

II. BACKGROUND

A. Grid’5000 Testbed

Grid’5000 is a scientific instrument designed to support experiment-driven research in all areas of computer science related to parallel, large-scale or distributed computing and networking [6]. It aims at providing a highly reconfigurable, controlable and monitorable experimental platform to its users. It currently comprises of 9 sites in France and one in Luxembourg [1]. Grid’5000 platform features almost 8000 cores (in more than 2200 AMD Opteron and Intel Xeon processors) in clusters located on the different sites. Grid’5000 features both Myrinet and Infiniband network infrastructures in its clusters as well as Gigabit Ethernet. Sites are interconnected through a dedicated 10 Gb/s wide area network operated by RENATER. The platform is available to researchers through the OAR advanced reservation system [7]. A reservation consists of a list of cluster nodes that can be located on one or several sites for a given amount of time. OAR attributes a unique identifier to each reservation and records the time when it was submitted, when it will start and when it will finish. When the reservation expires, it records its final status that can be either terminated or error. Once a user gets its reserved nodes, she has their exclusive usage and she can deploy the whole software stack of her choice on the nodes from the network to application layers through the operating system, middleware, application runtime

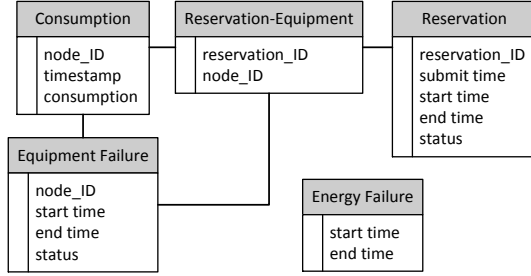


Fig. 1. Overview of Grid'5000 Data Model.

	Nodes	Reservations	Failures
Capricorne	56	90521	6815
Sagittaire	79	91159	16592

TABLE I
DATA SUMMARY.

and programming environment layers using the Kadeploy tool [14]. OAR maintains a log recording individual node failures that is to say periods during which nodes are unavailable to user requests. For each node failure, a failure type is recorded along with the failure starting and end times and node identifier.

B. Dataset

In this paper, we are focusing on energy consumption for the Grid'5000 testbed. The energy consumption data used in our analysis are collected from the Lyon site of the Grid'5000 testbed. This site consists of two main clusters: *Capricorne* and *Sagittaire* with 56 and 79 nodes respectively. *Capricorne* machines have 2 AMD Opteron CPUs and 2GB of RAM each. *Sagittaire* machines are 2 AMD Opteron CPUs and 16GB RAM each. All nodes are equipped with SCSI disks. The data spans a period of 6 months, from September'09 to March'10. Power consumption is measured every second using wattmeters manufactured by the OMEGAWATT company, the precision being 0.125W [12].

Figure 1 shows a high-level view of the available data. The structure is abstracted to show basic relationships between each data entity. In this work, we are interested in the *Consumption* data, where timestamped logs are available for each node. The original data includes an ASCII file for each node, where file name indicates the node ID. The consumption raw logs contain only two fields: timestamp and consumption in watts. To enable the analyses to only process file contents, not the metadata (file name), we appended the node ID to each line in the raw log file.

Table I summarizes the available data that is used in our modeling. Each cluster consists of a number of nodes that can be individually reserved by users. The total number of node reservations in the dataset is shown, along with the number of recorded node failures.

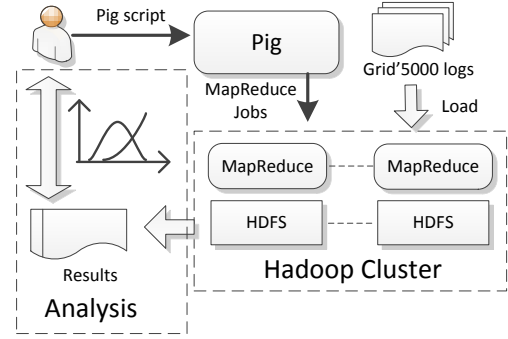


Fig. 2. Overview of analysis framework.

III. ANALYSIS APPROACH

In this section, we present explain our analysis approach and the tools and infrastructure used to generate the results.

A. Infrastructure

An overview of our analysis framework is shown in Figure 2. It uses two components of Apache's Hadoop – MapReduce and Hadoop Distributed File System (HDFS) – as the underlying distributed processing infrastructure [23], [3]. Hadoop MapReduce and HDFS are open source implementations of the MapReduce programming model [9] and Google File System [15] respectively. Hadoop provides scalability by automatically parallelizing MapReduce programs on clusters of machines, transparently handling host and application failures. At the analysis level, Apache Pig [4] was chosen as the platform for aggregating and summarizing the Grid'5000 dataset. Apache Pig has a high-level language, Pig Latin [17], that allows an analyst to easily write parallelized data analysis programs against very large datasets. The Pig programs are compiled into one or more sequenced MapReduce tasks which run on the underlying distributed processing infrastructure of HDFS and Hadoop MapReduce.

The first step is loading the Grid'5000 data into HDFS. We maintain the same directory structure and file formats available from the online repository. Once the data is available from HDFS, Pig Latin scripts can be executed on the Hadoop cluster. The results are stored back to HDFS. The range of analyses that can be performed at this step is limited by the supported built-in functions from Pig Latin. For our analysis, the main purpose of this step is to generate an aggregated view of the data that is more suitable for advanced analysis.

The output of the MapReduce phase is loaded into a desktop-scale analysis tool, which can perform advanced interactive visualization and further analysis. For this step, we used the popular *R* [2], [16] statistical analysis environment. The available results can now fit in memory for *R* to perform more comprehensive investigations.

B. Data Summarization

This section describes the methods used to perform basic statistical functions with Pig Latin scripts running on a Hadoop

```

x = LOAD '$in' USING PigStorage(',')
y = FILTER x BY watts>=0;
z = FOREACH y GENERATE
    node, watts;
g = GROUP z BY (node_ID);
s = FOREACH g GENERATE flatten(group),
    COUNT(z), MIN(z.watts),
    MAX(z.watts), AVG(z.watts),
STORE s INTO '$out' USING PigStorage(',');

```

Fig. 3. Example Pig Latin script. The variable names were omitted from the LOAD statement.

cluster. The results of the Pig Latin analysis phase will be fed to R statistical tools for more advanced, interactive data exploration; results from this detailed analysis are presented in the following section IV.

Computing Basic Statistics. A typical Pig Latin script describes a series of data transformations, or a data pipeline. Scripts begin with one or more LOAD operations that read the data from HDFS into a table-like structure where basic data types are supported. The loaded data can be processed by a FILTER operator. A GENERATE operator is available to select only a subset of the table columns. Statistics can be computed for the entire dataset, or the GROUP operator can be used in a similar fashion to the SQL GROUP BY operator. The example in Figure 3 groups the data by node, and computes the minimum, maximum and average energy consumption values in watts. Finally, the script uses the STORE operation to save the results to HDFS.

Pig Latin supports a limited set of aggregation operators, e.g. count, average, minimum and maximum. These do not include such common statistical measures as variance and correlation. Therefore, we wrote our own Pig Latin scripts for these functions. We show the variance computation here, as it was essential for our analysis. The correlation computation was done, but not applied to the energy consumption data, so it will not be presented here.

Variance. In our analysis, we used the variance calculation formula that is more parallel in nature, $s^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - (\frac{1}{n} \sum_{i=1}^n x_i)^2$, as opposed to $s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$, which first computes the population mean, $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$, then applies the difference and summation for the entire dataset. This can be performed by modifying the GENERATE statement in the original script where the projection will first generate $watts$ and $watts \times watts$, then after grouping, the equation will be added to the final GENERATE statement to compute the variance.

Time-series aggregation. The energy consumption data is collected every second by energy monitors at each node, resulting in a massive number of records. It is impractical to model the entire dataset at that granularity, especially that not every data point would be of interest to administrators or users.

When the time-series consists of consecutive fixed time intervals, an efficient and easily parallelized technique for this is called time *masking*. In time masking, given a certain time precision (minutes, seconds, milliseconds), each time stamp

will be rounded to the nearest value for that precision. The masked time stamp can then be used as part of the *key* in the GROUP statement, allowing different measurements to fall in the bucket of the specific time stamp. Time masking, sometimes called binning, is not a novel mechanism. What our framework offers is the ability to efficiently reduce the data and aggregate over multiple time scales for further analysis. Masking here is used as the means to exploit the parallelism.

We applied the statistical functions described above to perform the following analyses on the *Grid'5000* dataset:

- **Stats:** This includes minimum, maximum, average and standard deviation.
- **Outliers:** Each measured value is compared to the mean and marked as positive (negative) outlier if it is higher (or lower) than two standard deviations from the mean.
- **Histogram:** To approximate the real value of consumption (which could generate infinite number of values), the value is approximated to a certain precision by the same *masking* as the time variable.

Each analysis was expressed as a Pig Latin script, and executed on the entire dataset. Different levels of aggregations over time were performed (1 minute, 5 minutes, 1 hour, and 1 day). The results of those scripts are saved to HDFS, and retrieved for additional analysis using R .

C. Time Series Modeling

The goal of this work is to compute models for predicting energy consumption at the Grid'5000. We investigate here possible time series approaches to model the energy consumption over time. Time series analysis comes natural to the raw data, as a consumption value is measured every second. It is also applicable to data summarization and aggregation results from Pig analysis, as it is also following the format of time stamped measures with different periods.

Given a dataset of N samples over time X_1, X_2, \dots, X_N , the goal of time series modeling is to predict or forecast future samples by learning model from X series. Some regression techniques can be applied using the entire series, but the modeling step in this case will be extremely costly, and the resulting model cannot be generalized to other datasets. General time series analysis aims at finding the model that can approximate an observation X_t at time t using the minimum number of previous observations, X_i where $i < t$. Model identification is the process of finding out how many historical samples are sufficient to accurately predict future values. For this, a number of diagnostic functions are available and can identify what is called model order.

Our analysis will use an ARMA (AutoRegressive Moving Average) process. An $ARMA(p, q)$ process is used to model the series, where a sample at time t can be calculated using two components: Auto Regressive and Moving Average components.

$$X_t = \sum_{i=1}^p \alpha_i X_{t-i} + \sum_{j=0}^q \beta_j \epsilon_{t-j}$$

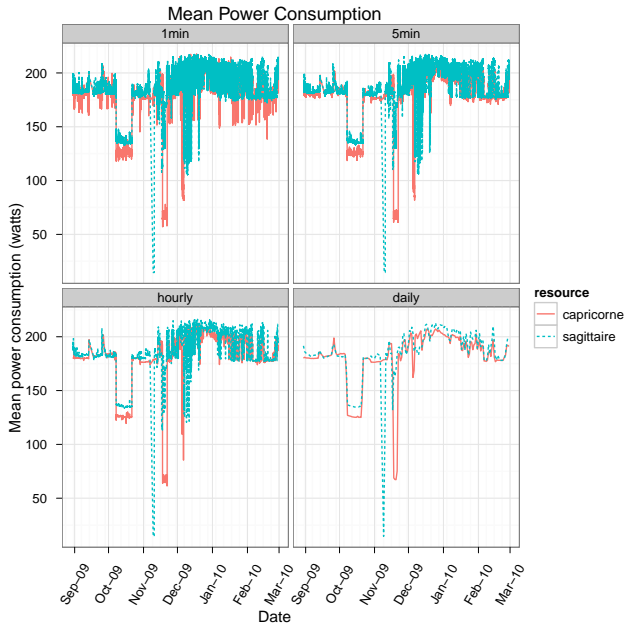


Fig. 4. Mean power consumption for each cluster, aggregated over different time scales.

Model identification is used to find the model order (p, q) , and the available data is used to compute the coefficient. The first component uses previous samples to compute the summation, while the second component incorporates some random process reflecting noise in the data. The diagnostic step to compute (p, q) uses autocorrelation for the entire series, as will be seen in the next section with the results.

IV. MODELS AND PREDICTIONS

This section presents the detailed analysis of energy consumption data, after basic aggregations have been computed using Hadoop and Pig. The results in the following subsections are computed using R .

A. Preliminary Results

We first start by inspecting the overall energy consumption trend for the two clusters. This step produces visual representations for the summary statistics across all nodes. The summaries enable quick exploration of the data, to detect areas of interest for further analysis.

Figure 4 shows the mean energy consumption of *Capricorne* and *Sagittaire* clusters for the period of 6 months. Each panel of the figure corresponds to different aggregation scales; 1-minute, 5-minute, hour, or day. We can see that the trends are very similar, where peaks and valleys are consistent across different time scales. Also, for the period after December'09, the measurements seem to have a stable trend. This might be attributed to overcoming any transient configuration problems related to monitoring and data collection. Also, many of the drops might reflect node failures, or missing data from the energy monitoring equipment failures. For example, the drop around December'09 for 1-minute, 5-minute and hour

aggregation was diluted at the daily aggregations. This might indicate that only a subset of the nodes have low consumption, or did not report the consumption. The consistent drops may correlate with system failures.

We now proceed with showing the outliers computed for each individual node for each cluster. We only show the results for hourly and daily aggregation, as the minutes aggregation was over-detecting many values. Figure 5 shows an overview of the detected outliers for *Capricorne* and *Sagittaire* clusters. The time is shown on the x-axis, and each node name on the y-axis. The positive outliers (high energy consumption with respect to the mean) are marked with red flags. The green flags show negative outliers (low consumption values). For high consumption, we wish to identify the cause, and possibly build models that can automatically flag the outliers in real-time. For low consumption, in many cases, the value is lower than the consumption of the machine in idle state. For this case, we wish to identify if the consumption prior to the outlier correlates with failure events on the node.

Upon identifying outliers, detailed analysis can be triggered at the specific nodes. For example, detailed analysis should be performed for nodes with high consumption; *capricorne-21*, *capricorne-22*, *capricorne-23*, *capricorne-25*, *sagittaire-1*, *sagittaire-10*, and *sagittaire-73*.

B. Overall Energy Model

We first start by modeling the average energy consumption for the entire cluster for both *Capricorne* and *Sagittaire*. We perform the analysis first on the daily aggregation, corresponding to the least level of granularity, where samples are grouped daily, and the final series contains the fewest samples. Accurate prediction of future samples from the daily level will save a lot of processing time, but cannot provide near-real-time prediction.

The model identification process starts by plotting the autocorrelation function (ACF) and partial autocorrelation (PACF) for the dataset. Those autocorrelation measures reflect the correlation between samples of the time series, and the same series when shifted back in time (lagged). By looking at the correlation values at each lag value (how many steps back in time), the series can be modeled by an order where this correlation disappears. In other words, the correlation vanishes when the specific historical sample no longer affect the prediction. For the Auto Regressive part of the model p , the autocorrelation is used. The partial autocorrelation can identify the order of the Moving Average component q .

Figure 6a shows the autocorrelation plot of the average daily energy consumption for *Capricorne*. The blue dashed lines show the confidence window, where the correlation can be considered as zero value. The first lag step where the ACF almost vanishes is 10. For the PACF in Figure 6b, the first lag where the PACF vanishes, or start cycling around zero is 5. Therefore, the series can be modeled as an $ARMA(10, 5)$ process. This means 10 historical samples are needed to forecast new daily consumption levels on *Capricorne*. Figure 6c shows

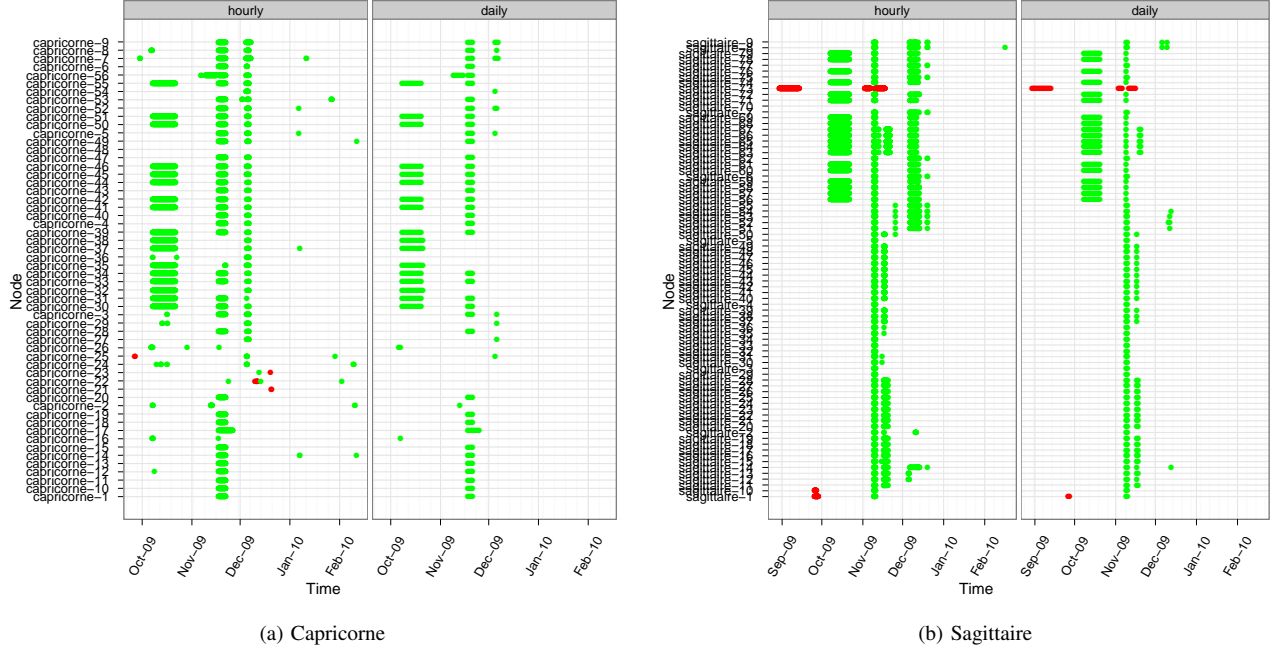


Fig. 5. Detected energy consumption outliers for *Capricorne* and *Sagittaire*. The red points show positive outliers (high consumption in watts), while the green show negative outliers (potential failures).

the prediction results from the fitted model. Note here that the time on the x-axis is shown as a time step, not the actual time. The right-most 7 samples are the model prediction, and the figure shows that they fall inside the confidence envelop (orange window). The dashed red line across the entire series shows a single prediction at each time step using the resulting model. This shows the accuracy of the model, where the predictions match the original data.

We proceed by inspecting the models of the average hourly consumption levels for *Capricorne*. For this, more samples are used in the modeling step, and the prediction can be made for hourly steps. Figure 7 shows the ACF, PACF, and the final forecast from the model. The ACF plot shows a different result from the daily aggregation. In the hourly aggregation, the ACF does not reach zero, which suggests that an Auto Regressive process cannot produce accurate model for the series. On the other hand, the PACF reaches zero at 5 as the daily aggregation PACF. We fitted a similar *ARMA* model for comparison, and predicted the future 7 days (7×24 steps for the hourly model). From Figure 7c, we can see that the dashed fitted line accurately captures the series, and the predictions of the future 7 days are within the confidence envelop. We notice also that the dashed line fits the series better than the daily predictions from Figure 6c.

It is worth mentioning that each lag value for the daily aggregation reflects 24 steps in the hourly aggregation. We inspected the ACF for the hourly aggregation with maximum lag of 300, which showed to have converged. This reflects the nature of the data, but unfortunately would be extremely inefficient to model. As the forecast plot showed, the low order

can fit the data reasonably well, without the the extreme fitting.

Similar results are shown for *Sagittaire*, where the *ARMA* model is able to accurately predict both daily and hourly consumption. Figure 8 shows the results (ACF and PACF plots are omitted for space considerations). We notice that the red-dashed line on both daily and hourly models accurately fits the original series. The forecasts are also within the confidence envelops.

C. Per node Energy Model

The next step in the analysis is zooming in on nodes where outliers have been detected. In this section, we only show the results for *Capricorne-21*. A time series model was computed for the average energy consumption for that node using both daily and hourly aggregation. We used an *ARMA* model, and performed the same diagnosis; ACF, PACF, and forecast. The forecast/prediction results are shown in Figure 9. We can see that the hourly series has significantly more variations than the daily series, and the red dashed predictions does not fit the original series as well as the daily prediction. It seems in this case that the outliers are affecting the model accuracy, where smoothing them helps the daily case. To better understand the outliers, more analysis should be performed. It is clear that the applied time series model does not accurately describe single node consumption, especially with outliers.

V. PERFORMANCE

In our analysis, we relied on Hadoop to perform the large-scale data aggregation, then used *R* to do the advanced modeling and visualization. The most expensive step in the

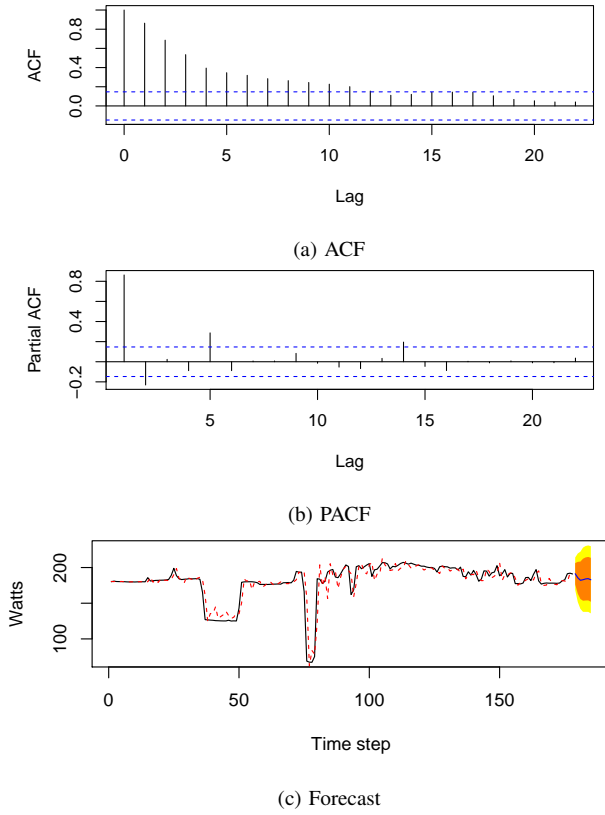


Fig. 6. Time series analysis for energy consumption on *Capricorne*, aggregated daily.

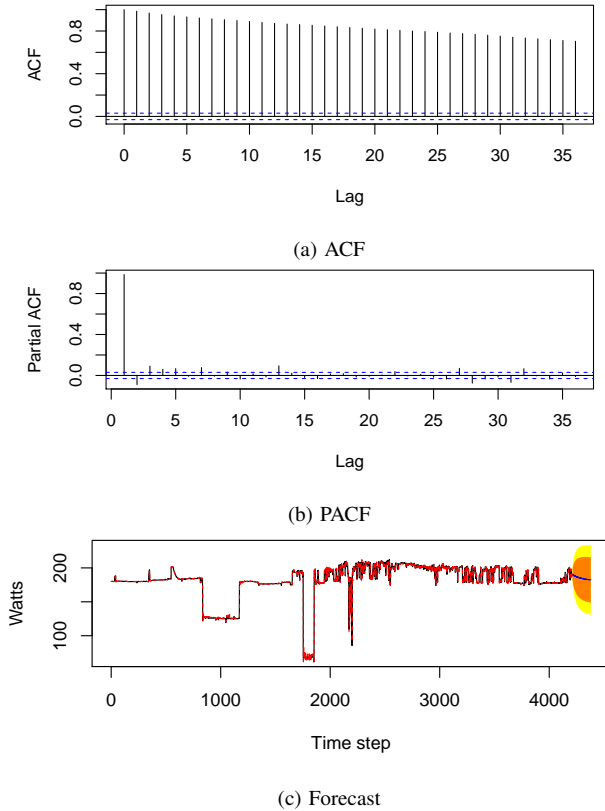


Fig. 7. Time series analysis for energy consumption on *Capricorne*, aggregated hourly.

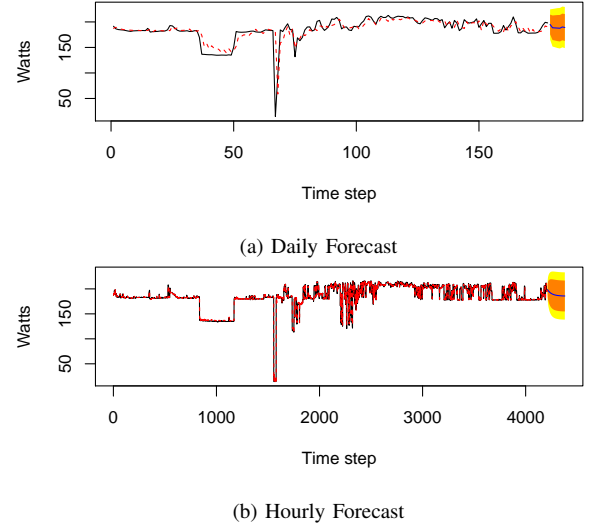


Fig. 8. Model evaluation and forecast for *Sagittaire*, both daily and hourly aggregations.

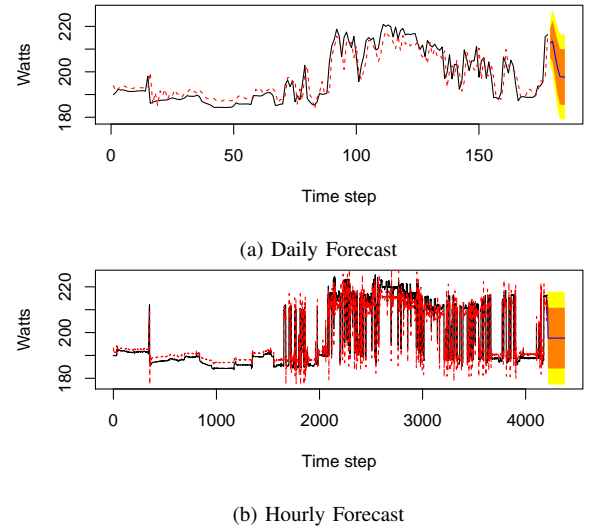


Fig. 9. Model evaluation and forecast for *Capricorne-21*, both daily and hourly aggregations.

process is data summarization using Hadoop. Once the data is summarized, analysis with *R* takes seconds to finish. This section shows the performance of Pig on Hadoop for the summarization described in Section III-B. We performed the analysis on NERSC Hadoop [22] cluster with 55 active nodes. The total data size used in the analysis consists of 41GB. This consists of a total of 135 energy consumption files.

The analysis scripts were run on the data for each cluster (*Capricorne* or *Sagittaire*) separately. We provide here the running time of the different evaluation scripts (statistics, outliers, and histograms) at different time scales (1-min, 5-min, hour, and day). It was expected that operations on the *Sagittaire* dataset would take more time, as the number of nodes are larger. This is due to the way the energy monitors recorded data, where each monitor saves a measure every second, resulting in each node recording almost the same number of samples. Figure 10 confirms the expectations. It

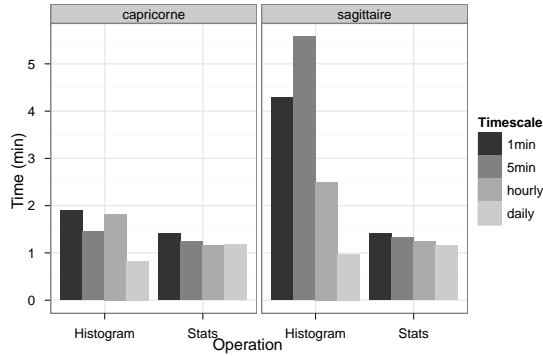


Fig. 10. Time for processing overall statistics and histograms for different time scales.

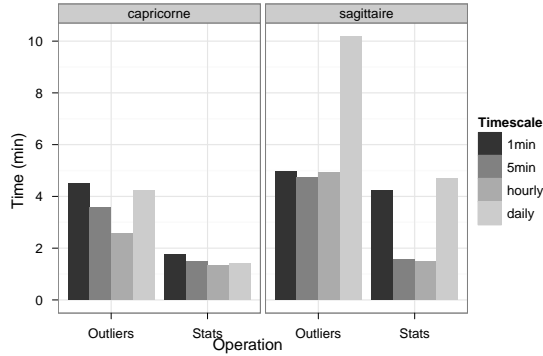


Fig. 11. Time for processing statistics for each individual node at different time scales.

also shows that high-level aggregations takes less time. This shows that the dominant factor in the Hadoop process is not the computation, rather the storing and moving of the results according to the keys. For example, one minute aggregation will result in more samples in the outcome, but performs less computations. The histogram operation on *Sagittaire* takes a significant amount of time compared to other operations. Recall that the histogram computation is performed by binning on the consumption values, which means that if there is high variation in the results, more values will be processed.

It is also important to evaluate the effect of performing the computation with respect to individual nodes. Figure 11 shows the resulting numbers, where the total time almost doubles over the time to average for the entire dataset. Again, *Sagittaire* dataset take more time, as there are more nodes. It is also evident that outlier detection takes more time, as a result of computing the statistics then performing a comparison to flag the outlier for each sample. An interesting observation is shown at the daily aggregation, where it was expected for it to be the fastest. In our analysis, we ran each script with the time scale as a parameter. It might be the case that while the script was running computing the daily aggregation, the Hadoop cluster was running other jobs. This needs to be verified. It is not affecting the analysis here, as this step is performed once for the entire dataset. When the statistics are available, models are built and optimized in *R*, which is much more efficient.

VI. RELATED WORK

The instrumentation of energy consumption on the Grid'5000 testbed has inspired work in modeling energy consumption on different aspects. Some work focused on developing the framework to enable basic correlations and predictions for consumptions and user reservations for better resource utilization. The Energy-Aware Reservation Infrastructure EARI [18] provides a decision system for the users to act on predictions performed using the Grid5000 infrastructure. The average value is used in the prediction, along with estimated reservation time, and the current utilization of resources. Different modes of operations are proposed that enable switching on and off resources, or explicitly asking for user input. EARI was integrated with GREEN-NET framework [8] to provide a more comprehensive set of options for the user in managing energy consumptions for grids and clouds.

In [5], the same dataset was used to investigate the relationship between energy consumption on servers and user reservation. The work found a small correlation between power and resource utilization, while idle periods of servers consumed a significant amount of energy. Another study focused on comparing different checkpointing and fault tolerance mechanisms, in terms of energy consumption [13]. The approach used the Grid'5000 data at the Lyon site. Comparisons of RAM and HDD checkpointing showed that RAM consume more energy, but HDD takes more time to log, making HDD less efficient.

Our work on the Grid'5000 dataset models the entire collection of data as a time series process, which enables higher granularity modeling and more flexibility in fitting model parameters, in contrast to previous work that focused more on the platform and used basic models. The pipeline can be used to model new dataset, and the models can be compared for similar systems.

Another category of work on the energy modeling problem compared different platforms in terms of energy efficiency. The work in [20] compared energy efficiency of data centers with two different processors architecture; ARM and Intel workstations. In [10] a system of hierarchical measurements is proposed to perform real-time scheduling of virtual machines to optimize energy. The testbed was implemented and various workloads were investigated. In [21] a control-theoretic approach was used to enable coordinated power management for data centers. This provided a general framework to model multiple architectures and incorporate multiple components.

For the purpose of predicting power consumption, Gaussian Mixture Models (GMM) were used in [11]. The approach models power while the system is utilized. Several architectural metrics are collected at the physical resource level, where different workloads are evaluated. A central server learns a model for each physical machine, and predicts consumption for the resource when certain workloads are running.

The study in [19] answers some basic questions and addresses assumptions and misconceptions in energy consumption modeling for large-scale grids and clouds. Many assumptions were proved wrong (linear CPU consumption versus

load, constant consumption for homogeneous nodes, among others...). OS versions have different impact on energy, while virtualization has negligible cost for most cases. Switching on/off nodes was shown to be very energy-efficient.

As opposed to the previous work focusing on small testbeds and workstations, our large-scale analysis framework enables extending the work to handle large systems.

VII. CONCLUSION

In this paper we described a generic approach to analyze large power consumption datasets collected from computing infrastructures. For an accurate and efficient analysis, the first step of the proposed data analysis pipeline consists of computing basic statistics and aggregating data at different time-scales using Pig data processing on Hadoop. In the second step, using the R framework, we exploit aggregated data seen as time-series to detect outliers and derive hourly and daily power consumption predictive models. We have applied our approach to the analysis of the power consumption of two clusters of the Grid'5000 experimentation platform used in production. We demonstrated the accuracy of the predictive models and the efficiency of the data processing performed on a 55-node cluster at NERSC.

Our future plan for this work will start by applying the analysis to datasets from other sites on the testbed, and comparing the time series models from different sites for possible generalizations. We will also investigate the correlation between energy consumption and failures both from nodes and energy equipments. Outlier detection combined with failure analysis might provide accurate root-cause analysis for both failures and high energy consumption. User information and reservation will be considered and correlated with energy models as well.

ACKNOWLEDGMENT

This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under contract DE-AC02-05CH11231.

REFERENCES

- [1] Grid'5000 Experimentation Platform. www.grid5000.fr.
- [2] R. www.r-project.org.
- [3] Apache hadoop. <http://hadoop.apache.org/>.
- [4] Apache pig. <http://pig.apache.org/>.
- [5] Marcos Dias de Assuncao, Anne-Cécile Orgerie, and Laurent Lefevre. An analysis of power consumption logs from a monitored grid site. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, GREENCOM-CPSCOM '10, pages 61–68, Washington, DC, USA, 2010. IEEE Computer Society.
- [6] Raphaël Bolze, Franck Cappello, Eddy Caron, Michel Dayde, Frédéric Desprez, Emmanuel Jeannot, Yvon Jégou, Stéphane Lanteri, Julien Leduc, Nouredine Melab, Guillaume Mornet, Raymond Namyst, Pascale Primet, Benjamin Quétier, Olivier Richard, Talbi El-Ghazali, and Iréa Touche. Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed. *International Journal of High Performance Computing Applications*, 20(4):481–494, 2006.
- [7] N. Capit, G. Da Costa, Y. Georgiou, G. Huard, C. Martin, G. Mounie, P. Neyron, and O. Richard. A batch scheduler with high level components. In *Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on*, volume 2, pages 776 – 783 Vol. 2, may 2005.
- [8] Georges Da Costa, Marcos Dias de Assunção, Jean-Patrick Gelas, Yannis Georgiou, Laurent Lefèvre, Anne-Cécile Orgerie, Jean-Marc Pierson, Olivier Richard, and Amal Sayah. Multi-facet approach to reduce energy consumption in clouds and grids: the green-net framework. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, pages 95–104, New York, NY, USA, 2010. ACM.
- [9] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. In *Proceedings of the 6th conference on Symposium on Operating Systems Design & Implementation - Volume 6*, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association.
- [10] Gaurav Dhiman, Giacomo Marchetti, and Tajana Rosing. vgreen: A system for energy-efficient management of virtual machines. *ACM Trans. Des. Autom. Electron. Syst.*, 16(1):6:1–6:27, November 2010.
- [11] Gaurav Dhiman, Kresimir Mihic, and Tajana Rosing. A system for on-line power prediction in virtualized environments using gaussian mixture models. In *Proceedings of the 47th Design Automation Conference, DAC '10*, pages 807–812, New York, NY, USA, 2010. ACM.
- [12] Marcos Dias de Assuncao, Jean-Patrick Gelas, Laurent Lefvre, and Anne-Cécile Orgerie. The green grid5000: Instrumenting a grid with energy sensors. In *5th International Workshop on Distributed Cooperative Laboratories: Instrumenting the Grid (INGRID 2010)*, Poznan, Poland, May 2010.
- [13] M. Diouri, O. Gluck, L. Lefevre, and F. Cappello. Energy considerations in checkpointing and fault tolerance protocols. In *2nd Workshop on Fault-Tolerance for HPC at Extreme Scale (FTXS 2012)*, Boston, USA, 06 2012.
- [14] Y. Georgiou, J. Leduc, B. Videau, J. Peyrard, and O. Richard. A tool for environment deployment in clusters and light grids. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, page 8 pp., april 2006.
- [15] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pages 29–43, New York, NY, USA, 2003. ACM.
- [16] Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.
- [17] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1099–1110, New York, NY, USA, 2008. ACM.
- [18] Anne-Cécile Orgerie, Laurent Lefèvre, and Jean-Patrick Gelas. Save watts in your grid: Green strategies for energy-aware framework in large scale distributed systems. In *Proceedings of the 2008 14th IEEE International Conference on Parallel and Distributed Systems*, ICPADS '08, pages 171–178, Washington, DC, USA, 2008. IEEE Computer Society.
- [19] Anne-Cécile Orgerie, Laurent Lefevre, and Jean-Patrick Gelas. Demystifying energy consumption in grids and clouds. In *Proceedings of the International Conference on Green Computing*, GREENCOMP '10, pages 335–342, Washington, DC, USA, 2010. IEEE Computer Society.
- [20] Zhonghong Ou, Bo Pang, Yang Deng, Jukka K. Nurminen, Antti Ylä-Jaaski, and Pan Hui. Energy- and cost-efficiency analysis of arm-based clusters. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, CCGRID '12, pages 115–123, Washington, DC, USA, 2012. IEEE Computer Society.
- [21] Ramya Raghavendra, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. No "power" struggles: coordinated multi-level power management for the data center. In *Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, ASPLOS XIII, pages 48–59, New York, NY, USA, 2008. ACM.
- [22] The magellan project. <http://www.nersc.gov/research-and-development/cloud-computing/>.
- [23] Tom White. *Hadoop: The Definitive Guide*. O'Reilly Media, original edition, June 2009.