An energy-aware adaptation model for Big Data platforms

Emiliano Casalicchio, Lars Lundberg, Sogand Shirinbad
Department of Computer Science and Engineering
Blekinge Institute of Technology, Sweden
Email: emiliano.casalicchio@bth.se, lars.lundberg@bth.se, sogand.shirinbad@bth.se

Abstract—Platforms for big data includes mechanisms and tools to model, organize, store and access big data (e.g., Apache Cassandra, Hbase, Amazon SimpleDB, Dynamo, Google BigTable). The resource management for those platforms is a complex task and must account also for multi-tenancy and infrastructure scalability. Human assisted control of Big data platform is unrealistic and there is a growing demand for autonomic solutions. In this paper we propose a QoS and energy-aware adaptation model designed to cope with the real case of a Cassandra-as-a-Service provider.

Keywords—Autonomic computing; Cloud computing; Green computing; Apache Cassandra; Big Data

I. INTRODUCTION

This work considers a big data platform provider, offering Apache Cassandra-as-a-Service. The provider wants to minimize operational costs under the constraints imposed by Service Level Agreements (SLAs) contracted with the customers. The strategy adopted to minimize costs is to reduce energy consumption. To address this problem we propose an energy-aware adaptation model specifically designed for Cassandra consumption. To address this problem we propose an energy-aware adaptation model designed to cope with the real case of a Cassandra-as-a-Service provider.

II. OPTIMAL ADAPTATION MODEL

In order to formulate our optimal adaptation problem we defined models for: the workload and SLA; the system architecture; the throughput; and the utility function. The solution of the optimization problem returns, for each tenant: the size of the Cassandra virtual datacenter in term of number of vnodes; the configuration of vnodes in terms of CPU capacity; the placement of vnodes on the physical nodes. The periodic, or event based evaluation of the optimisation problem provides a runtime adaptation policy for the Cassandra service provider.

The system workload consist of a set of read (R), write (W) and read & write (RW) operation requests generated by the N independent applications. Each application i generate only a type \( l_i \in L = \{R, W, RW\} \) of requests. In case \( l_i = RW \) the workload is composed of 75% R and 25%W requests.

In literature have been considered more sophisticated workloads that include, for example, scan and range queries. We limit this study to the set \( L \) above defined because the model we propose can deal with any type of operation requests.

The SLA model is describe by the tuple \( (l_i, T_i^{min}, D_i) \) where: \( T_i^{min} \) is the minimum throughput (operations per second) specified by the customer, \( D_i \) is the data replications factor (a specific configuration parameters of Cassandra).

The Architecture model describes a datacenter consisting of H homogeneous physical machines (PMs) installed at the same geographical location. We assume each PM \( h \) has a nominal CPU capacity \( C_h \) measured in number of available cores. We do not model the memory capacity because we assume that our scenario is not memory bound. Each Cassandra vnode run on a VM of type \( j \) configured with \( c_j \) virtual cores. We consider a set of V VM configurations. A change from configuration \( j_1 \) to \( j_2 \) is modelled as the replacement of the VM of type \( j_1 \) with a VM of type \( j_2 \). However, in a real setting, hypervisors such as VMWare allow to change at runtime the number of cores associated to a VM without service interruption. As suggested by the Cassandra management best practice the Cassandra virtual datacenter is composed of \( n_i \) homogeneous Cassandra virtual nodes where \( n_i \geq D_i \) and at least \( D_i \) out of \( n_i \) vnodes must run on different physical machines. Those three constraints are modelled as follow:

\[
\sum_{j \in J} y_{i,j} = 1, \quad \sum_{j \in J, h \in H} x_{i,j,h} \geq D_i \quad \sum_{h \in H} s_{i,h} \geq D_i \ \forall i \in I
\]

where: \( y_{i,j} \) is equal to 1 if application \( i \) use a VM configuration \( j \) to run Cassandra vnodes, otherwise \( y_{i,j} = 0 \). \( x_{i,j,h} \) is the...
number of Cassandra vnodes serving application $i$ and running on VMs with configuration $j$ allocated on PM $h$, $s_{i,h}$ is equal to 1 if a Cassandra vnode serving application $i$ run of PM $h$. Otherwise $s_{i,h} = 0$. Finally, $\mathcal{J} = [1, |V|] \subset \mathbb{N}$ is the set of VMs configurations indexes, $\mathcal{H} = [1, |H|] \subset \mathbb{N}$ is the set of PMs indexes and $\mathcal{I} = [1, |N|] \subset \mathbb{N}$ is the set of application indexes.

The throughput $T_i$ guaranteed by the provider to application $i$ is modelled as function of $x_{i,j,h}$. From the analysis of experimental data it emerges that the throughput of a Cassandra vnode serving requests of type $l_i$ and running on a VM of type $j$ (on top of a PM $h$) can be approximated with a set of linear segment with slope $\delta_{i,j}$. $\delta_{i,j}$ is the slope of the $k^{th}$ segment and is valid for a number of Cassandra nodes between $n_{k-1}$ and $n_k$. Therefore, for $n_{k-1} \leq x_{i,j,h} \leq n_k$ is valid the following equation:

$$t(x_{i,j,h}) = t(n_{k-1}) + \frac{\delta_{i,j} - \delta_{i,j} \cdot (x_{i,j,h} - n_{k-1})}{t_{i,j}(1) - t_{i,j}(0)}$$

where $k \geq 1$, $n_0 = 1$ and $t_{i,j,h}(1) = t_{i,j}(0)$.

The power consumption model we consider is linear [8] and the power $P_h$ consumed by a physical machine $h$ is function of the CPU utilization and therefore of the system configuration $x$:

$$P_h(x) = k_h \cdot P_{h}^{max} + (1 - k_h) \cdot P_{h}^{max} \cdot U_h(x)$$

where $P_{h}^{max}$ is the maximum power consumed when the PM $h$ is fully utilised (e.g. 500W), $k_h$ is the fraction of power consumed by the idle PM $h$ (e.g. 70%), and the CPU utilization for PM $h$ is defined by $U_h(x) = \frac{1}{x} \cdot \sum_{x} x_{i,j,h} \cdot c_j$.

The optimization problem is formulated in Fig. 1. Eq. 3 guarantees the SLA is satisfied in term of minimum throughput for all the tenants. For the sake of clarity we keep this constraint non linear, but it can be linearized using standard techniques from operational research if the throughput is modelled using eq. 1. Eq. 4 guarantee that the number of vnodes allocated implement the replication factor specified in the SLAs. Eq. 5 model the assumption that for each tenant must be allocated homogeneous VMs and the number of vnodes of the Cassandra datacenter must be greater or equal than $D_i$. Eq. 6 control the maximum capacity of the physical machine is not exceeded. Eq. 7 guarantee that the Cassandra vnodes are instantiated on at least $D_i$ different physical machines. Finally, expressions 8 and 9 are structural constraints of the problem. For lack of space we omit a set of other structural constraints.

### III. Concluding Remarks

In this paper we explore the problem of energy-aware adaptation for Big Data platform providers and in the specific we consider the case of Cassandra-as-a-Service. We proposed a system model and an optimization problem that, as solution, return the optimal or sub-optimal configuration of the system. The main advantage of the model we propose is that it is based only on the knowledge of the relationship between the throughput and the number of Cassandra vnodes. This information is easy to be collect and maintained up to date at execution time.

Preliminary experimental results, not reported for lack of space, shown that: the proposed model has a correct behaviour in different dynamic scenarios. As expected, it's drawback is the entropy that successive adaptation actions can create. Further works will investigate techniques to reduce the system state perturbation due to adaptation actions. Moreover, we will consider the constraints imposed by the memory and the size of the dataset.

### Acknowledgments

Funded by the Knowledge Foundation (SE) grant n.20140032 "Scalable resource-efficient systems for big data analytics". Thanks to Jim Hakansson and Ericsson AB, Sweden for their support.

### References


