International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 6 Issue 10|| October 2017 || PP. 13-24

Design & Development of a Dynamic Cloud Computing Architecture to achieve QoS Levels and reduce Energy Consumption

Prof.Dr.G.Manoj Someswar[1], Naresh Alapati[2]
[1]*Visiting Professor & Research Supervisor, VBS Purvanchal University, Jaunpur, U.P., India*
[2]*Research Scholar, VBS Purvanchal University, Jaunpur, U.P., India*
Corresponding Author: Prof.Dr.G.Manoj Someswar[1]

Abstract: *Cloud computing is an emerging computing paradigm which is gaining popularity in IT industry for its appealing property of considering "Everything as a Service". The goal of a cloud infrastructure provider is to maximize its profit by minimizing the amount of violations of Quality-of-Service (QoS) levels agreed with service providers and at the same time, by lowering infrastructure costs. Among these costs, the energy consumption induced by the cloud infrastructure, for running cloud services, plays a primary role. Unfortunately, the minimization of QoS violations and at the same time, the reduction of energy consumption is a conflicting and challenging problem. In this research paper, we propose a framework to automatically manage computing resources of cloud infrastructures in order to simultaneously achieve suitable QoS levels and to reduce as much as possible the amount of energy used for providing services. We show through simulation, that our approach is able to dynamically adapt to time-varying workloads (without any prior knowledge) and to significantly reduce QoS violations and energy consumption with respect to traditional static approaches.*

Keywords: *MegaJoules (MJ), Reference Machine Multiplier (RMM), Response Time(RT), Squandered Energy.*

Date of Submission: 06-10-2017 Date of acceptance: 26-10-2017

I. INTRODUCTION

Performance Evaluation with VM Migration

In this research paper, we evaluate the performance of our resource management frame-work for the case where: all components of the resource management framework are used and the lifetime of each hosted application can potentially be shorter than the length of a replication.

Put in another way, in this case study, VMs can migrate to different physical machines and applications can start and finish in the middle of the simulation.

The research paper is organized as follows: First, in research paper, we describe specific settings used to setup this case study. We report and discuss experimental results.

Experimental Setup

In addition to settings presented in research paper, we provide the following configuration setup.

Table 1: Experimental setup – Physical machines characteristics.

# Instances	CPU Capacity	Reference Machine		Power Model		
		Multiplier	$^{w}0$	$^{w}1$	$^{w}2$	r
3	1000	x1	86:7 119:1 69:06			0:400
2	2000	x2	143:0	258:2	117:2	0:355
2	3000	x3	178:0	310:6	160:4	0:311
2	4000	x4	284:0	490:1	343:7	0:462

Physical Infrastructure Configuration

Unlike the case study presented in Chapter 8, the evaluation of our resource management framework is done in a heterogeneous environment in order to better take advantage of the potential offered by VM migration.

Specifically, we consider a set of nine heterogeneous physical machines whose characteristics are reported in Table 1 In this table, there is one row for each type of physical machines. In particular, the first column (named

"# Instances") shows the number of instances (of a particular type of physical machines) used in the experiments. The second column (labeled "CPU Capacity") reports the CPU capacity. The third column (called "Reference Machine Multiplier") indicates the relative CPU capacity (in terms of a multiplicative factor) with respect to the reference machine; for instance, a value of x2 means that the capacity of that type of physical machines is twice the one of the reference machine. The fourth to last columns (grouped under the label "Power Model") reports the co-efficients of the power consumption model, as defined by eq.

For the power consumption model, we estimate the values of the parameters w_0, w_1, w_2 and r through a statistical regression analysis over data collected by the SPECpower_ssj2008 benchmark [1], and the resulting fit is shown in the last columns of Table 1. A graphical comparison of these power models is shown in Fig.1 Interestingly, from the figure we can observe that, assuming a proportional relationship between capacities and utilizations of different physical machines (as we do in this thesis), it is not always effective (in terms of power consumption) to aggregate the largest number of VMs on the smallest number of physical machines, mostly because of idle power consumption. For instance, suppose that there are 3 identical VMs, each of which using the 100% of a physical

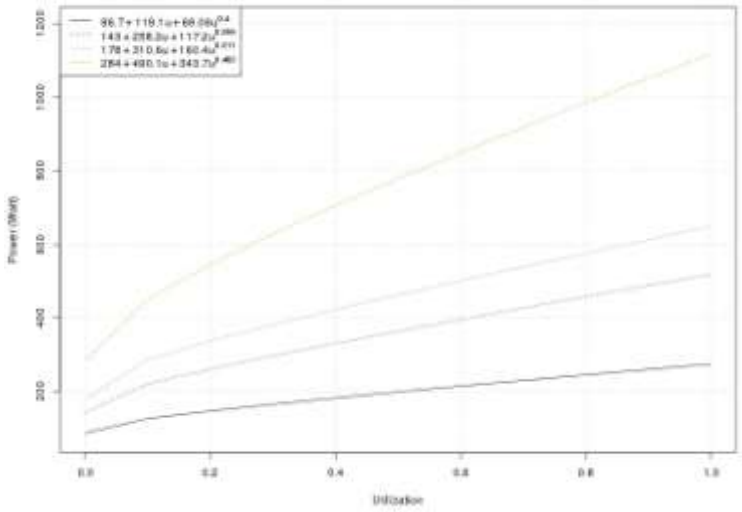

Figure 1: Experiments setup – Utilization vs power consumption of physical machines

machine of CPU capacity 1000; this means, that (according to our assumptions on proportional relationship between CPU capacities) if each VM is deployed on a physical machine of capacity 2000, 3000, or 4000, it will lead to a utilization of 50%, 33%, or 25%, respectively. Thus the resulting aggregated power consumption will amount to:

824:58 Watt, if 3 physical machines of capacity 1000 are used;
882:14 Watt, if 2 physical machines of capacity 2000 are used;
649:00 Watt, if 1 physical machine of capacity 3000 is used;
952:50 Watt, if 1 physical machine of capacity 4000 is used.

In this case, it is much more effective to consolidate the 3 VMs on the physical machine of capacity 3000 rather than put them on two or more physical machines with less capacity. However, this is not true if a physical machine of capacity 4000 is used in place of the one with capacity 3000.

Application Configuration

We use the same three types of applications, namely A_1, A_2, and A_3, as defined in this research paper. However, we allow each application to start and finish in the middle of each replica of the simulation experiment. Specifically, for each type of application, we create two instances, one with a lifetime that spans for the entire length of each replication, and the other with a shorter duration, in order to study system configurations in which the number of applications that are simultaneously present dynamically varies over time.[1]

In Table 2, we report the lifetime specifications for each application instance, inside each simulation replica. In the table, there is a row for each instance of application. Specifically, the first column (named "Type") shows the type of application. The second column (called "ID") indicates the identifier associated to a particular application instance, that will be used later in the discussion of experiment results. The third to last columns (grouped under the label "Lifetime") reports the start time and the duration of the execution of a particular application

Table 2: Experiments setup – Lifetime of each application instance

Application		Lifetime	
Type	ID	Start Time	Duration
A_1	$^{A}11$	0	–
A_1	$^{A}12$	10000	70000
A_2	$^{A}21$	0	–
A_2	$^{A}22$	70000	70000
A_3	$^{A}31$	0	–
A_3	$^{A}32$	130000	70000

instance inside each simulation replica; the symbol "–", used for the duration time, means that the application instance stops at the end of each replication.

From the above table, it can be noted that the maximum number of simultaneously running application instances is 5, which comprises the 3 "always running" instances, and 2 instances with shorter lifetime. This happens during the time intervals [70000;80000], when both A_{12} and A_{22} are executing, and [130000;140000], when both A_{22} and A_{32} are running.

Migration Manager Configuration

The configuration of the Migration Manager includes the smoothing factor of the EWMA filters, the parameters specific to the optimization problem, and the sampling time.[2] In the following, we provide some detail about the choice of each of them.

EWMA Filters

The Migration Manager uses two EWMA filters: one for computing $\hat{u}_j(k)$ (i.e., the expected contribution to the mean CPU utilization of a physical machine with a capacity equivalent to the one of the reference machine, that will be brought by VM j at control interval k – see Table 2 and Eq., and the other for computing $\hat{s}_j(k)$ (i.e., the expected mean CPU share that is assumed VM j will demand to a physical machine with a capacity equivalent to the one of the reference machine, at control interval k – see Table 2 .

For such filters, we need to specify their respective smoothing factors, namely **b** and g. For both of them, we choose a value of 0:70 so that the influence of past observations does not vanish too fast.

Optimization Problem

For the parameters of the optimization problem, we choose the following values: Maximum aggregated CPU share demand s^{max}_i for physical machine i: we choose the value 1 for all the physical machines.

Minimum CPU share $\bar{s}^{min}_t$ assignable to tier t on the reference machine: we choose the value 0:2 for all the tiers of all the applications. We derive it by means of offline system identification experiments. Specifically, for every experiment, we excite each application with three randomly generated and uniformly distributed in $[\bar{s}^{min};1]$ signals (each of which representing the CPU share assigned to each tier), by varying $\bar{s}^{min}$ in the range (0;0:5] with a step increment of 0:05. From these experiments, we find that, for values of $\bar{s}^{min}$ below 0:2, the behavior of an application becomes unstable due to too many queueing phenomena, with the result that the response time diverges (theoretically) to the infinity. CPU utilization threshold u^{max}_i for physical machine i: we set it to 1 for every physical machine.

Weights w_e, w_m, and w_p for the objective function J: we choose the value 1 for all the weights, so that the three costs J_e, J_m and J_p, of the objective function, are equally weighted.

For what concerns the value of the mean CPU share $\bar{s}_t$ of each tier t (of every application) on the reference machine, we use the same benchmark-like approach (similar to the one described in for application profiling), already employed for computing the set-point for the LQ control design of the Application Manager.[3]

Sampling Time

The value of the sampling time T is derived by means of offline trial-and-error experiments, from which we find that a reasonable value is to set it to 1800 ticks of simulated times (e.g., if one tick of the simulated time represents 1 second, the sampling time amounts to half hour).

Performance Metrics

The performance of our resource management framework is assessed by means of simulation, by using the independent replications output analysis method, where the length of each replication is fixed to 2100000 ticks of simulated time, and the number of total replicas is fixed to 5.[4]

We use these fixed values since the simulated system never reaches the steady-state due to the presence of applications that can start and stop in the middle of the simulation.

Experimental Scenarios and Resource Management approaches

Experiments performed for this case study, use the same four scenarios described in Chapter, that is S-DMPP, S-PMPP, S-MMPP, and S-MIX. Also, as discussed in this research paper, for each scenario, we evaluate our approach with three commonly used techniques, that is STATIC-SLO, STATIC-ENERGY, and STATIC-TRADEOFF. However, in order to evaluate the efficacy of VM migration, we consider two different variants of our approach (instead of only one), namely OUR-APPROACH-NM and OUR-APPROACH-M. Specifically, in OUR-APPROACH-NM, we do not use the Migration Manager component, while in the OUR-APPROACH-M variant, this component is employed.

For what concerns the implementation of the Migration Manager, as discussed in Section 6.4, we propose two possible implementations based on approximated algorithms, that is a greedy algorithm and a strategy based on local optimization.[6]

In order to evaluate both implementations, we divide the experiments in two different groups, namely MM-GREEDY and MM-LOCOPT. In the MM-GREEDY group, the Migration Manager is implemented by means of the greedy algorithm, while in the MM-LOCOPT group, the Migration Manager is driven by local optimization.[5]

II. RESULTS AND DISCUSSION

This research paper is organized as follows. In this research paper, we present and discuss the results obtained by each individual experiments groups, that is-GREEDY and MM-LOCOPT, respectively. Finally, we present some concluding remark about the convenience of using VM migration and the behavior two different implementations of the Migration Manager.

Results for the MM-GREEDY Experiments Group

The results of the various scenarios, in the MM-GREEDY group, are presented in four separate tables: Table 3 for S-DMPP, Table 4 for S-PMPP, Table 5 for S-MMPP, and finally Table 6 for S-MIX. For the sake of readability, we limit to report only those results deriving from the best combination of RLS algorithm and LQ control design, among all of their variants we considered in this research paper.

In each table, every column reports the results obtained by the various applications, under a specific resource management approach (i.e., STATIC-SLO, STATIC-ENERGY, STATIC-TRADEOFF, OUR-APPROACH-NM, and OUR-APPROACH-M).

A column filled with the symbol "n/a" (which stands for "result not available") means that the use of the corresponding resource management approach made the simulation unable to converge.[7] Numbers inside parenthesis (when present) represent the standard deviations of the related measures, while letters inside parenthesis represent unit of measures (e.g., "(s)" means seconds). The rows of each table have instead the following meaning. Rows labeled by $A_{i\,j}$, for i = 1;:::;3 and j = 1;2, report the 99^{th} percentile of the response time (label "Response Time"), expressed in seconds (s), and the mean percentage of SLO violations (label "% SLO Violations") for each application instance; lower values correspond to better results. The row labeled by "Uptime" reports the sum of the mean uptime of all the physical machines, where the uptime of a physical machine is defined as the total time (from the beginning of each simulation replica) that the physical machine has been powered on.[8] This metric quantifies the efficiency of a given approach in using the physical machines of the cloud infrastructure, since lower "Uptime" values indicate the usage of a lower amount of physical resource capacity to serve a given workload.

Table 3: Experimental evalResults for the S-DMPP scenario in the MM-GREEDY group

			Approach				
			STATIC-SLO	STATIC-ENERGY	STATIC-TRADEOFF	OUR-APPROACH-NM	OUR-APPROACH-M
A_{11}	Response Time	(s)	1:17 (0:02)	3:21 (0:23)	1:53 (0:02)	1:16 (0:01)	1:17 (0:03)
	% SLO Violations	(%)	0:62	19:19	2:72	0:63	0:63
A_{12}	Response Time	(s)	1:17 (0:00)	3:08 (0:04)	1:57 (0:01)	1:17 (0:00)	1:17 (0:00)
	% SLO Violations	(%)	0:63	19:73	2:84	0:63	0:63
A_{21}	Response Time	(s)	0:62 (0:01)	1:59 (0:03)	0:81 (0:01)	0:62 (0:01)	0:62 (0:01)
	% SLO Violations	(%)	0:76	14:35	2:72	0:76	0:76
A_{22}	Response Time	(s)	0:63 (0:02)	1:66 (0:01)	0:84 (0:02)	0:63 (0:02)	0:63 (0:02)
	% SLO Violations	(%)	0:76	14:65	2:75	0:77	0:76
A_{31}	Response Time	(s)	0:61 (0:01)	1:70 (0:01)	0:82 (0:00)	0:61 (0:01)	0:61 (0:01)
	% SLO Violations	(%)	0:77	19:40	3:19	0:77	0:77
A_{32}	Response Time	(s)	0:61 (0:00)	1:69 (0:04)	0:83 (0:00)	0:61 (0:00)	0:61 (0:00)
	% SLO Violations	(%)	0:80	19:29	3:20	0:80	0:80
Uptime		(Ms)	0:97	1:25	1:03	0:97	0:85

Table 4: Experimental evaluation – Results for the S-PMPP scenario in the MM-GREEDY group.

			STATIC-SLO	STATIC-ENERGY	STATIC-TRADEOFF	OUR-APPROACH-NM	OUR-APPROACH-M
A_{11}	Response Time	(s)	1:19 (0:02)	3:05 (0:04)	1:59 (0:02)	1:19 (0:02)	1:19 (0:02)
	% SLO Violations	(%)	0:58	19:60	2:57	0:58	0:59
A_{12}	Response Time	(s)	1:29 (0:00)	3:41 (0:10)	1:73 (0:01)	1:29 (0:01)	1:29 (0:00)
	% SLO Violations	(%)	0:89	33:89	4:17	0:90	0:90
A_{21}	Response Time	(s)	0:81 (0:23)	1:64 (0:01)	0:86 (0:00)	0:81 (0:23)	0:81 (0:23)
	% SLO Violations	(%)	0:68	16:04	2:68	0:68	0:68
A_{22}	Response Time	(s)	0:68 (0:02)	1:36 (1:29)	0:90 (0:02)	0:69 (0:00)	0:68 (0:01)
	% SLO Violations	(%)	0:82	9:33	3:24	0:83	0:82
A_{31}	Response Time	(s)	0:59 (0:00)	1:56 (0:01)	0:79 (0:00)	0:59 (0:00)	0:59 (0:00)
	% SLO Violations	(%)	0:53	15:62	2:37	0:53	0:53
A_{32}	Response Time	(s)	0:59 (0:01)	1:55 (0:05)	0:82 (0:03)	0:59 (0:01)	0:59 (0:01)
	% SLO Violations	(%)	0:47	11:57	1:84	0:47	0:47
Uptime		(Ms)	1:35	1:83	1:79	1:32	1:04
Energy Consumption	Total Energy	(MJ)	393:42	419:24	427:40	386:33	353:78
	Wasted Joules	(MJ)	2:51	74:33	11:64	2:47	2:26

Table 5: Experimental eval Results for the S-MMPP scenario in the MM-GREEDY group

			Approach				
			STATIC-SLO	STATIC-ENERGY	STATIC-TRADEOFF	OUR-APPROACH-NM	OUR-APPROACH-M
A_{11}	Response Time	(s)	4:29 (0:17)	n/a	43:63 (0:34)	4:44 (0:12)	4:43 (0:17)
	% SLO Violations	(%)	1:01	n/a	28:31	1:12	1:14
A_{12}	Response Time	(s)	4:45 (0:45)	n/a	24:03 (1:55)	4:08 (0:10)	4:39 (0:32)
	% SLO Violations	(%)	0:86	n/a	19:61	0:88	0:88
A_{21}	Response Time	(s)	2:04 (0:18)	n/a	35:18 (3:83)	2:04 (0:17)	2:04 (0:19)
	% SLO Violations	(%)	0:89	n/a	21:83	0:90	0:90
A_{22}	Response Time	(s)	1:73 (0:35)	n/a	10:93 (0:16)	1:76 (0:39)	1:73 (0:34)
	% SLO Violations	(%)	0:36	n/a	9:90	0:37	0:36
A_{31}	Response Time	(s)	1:82 (0:06)	n/a	21:57 (0:36)	1:82 (0:06)	1:82 (0:06)
	% SLO Violations	(%)	0:67	n/a	15:68	0:67	0:66
A_{32}	Response Time	(s)	1:95 (0:58)	n/a	15:92 (2:81)	2:02 (0:71)	2:00 (0:66)
	% SLO Violations	(%)	0:62	n/a	15:68	0:62	0:63
Uptime		(Ms)	0:97	n/a	1:17	0:97	0:89

TalTable 6 : Experimental evaluation Results for the S-MIX scenario in the MM-GREEDY group

			STATIC-SLO	STATIC-ENERGY	STATIC-TRADEOFF	OUR-APPROACH-NM	OUR-APPROACH-M
A_{11}	Response Time	(s)	0:61 (0:00)	n/a	0:83 (0:00)	0:61 (0:00)	0:61 (0:00)
	% SLO Violations	(%)	0:81	n/a	3:23	0:81	0:81
A_{12}	Response Time	(s)	0:62 (0:00)	n/a	0:83 (0:00)	0:62 (0:00)	0:89 (0:38)
	% SLO Violations	(%)	0:79	n/a	3:14	0:79	0:79
A_{21}	Response Time	(s)	1:98 (0:03)	n/a	35:14 (19:65)	2:00 (0:05)	1:96 (0:06)
	% SLO Violations	(%)	0:87	n/a	20:86	0:84	0:82
A_{22}	Response Time	(s)	1:71 (0:19)	n/a	16:19 (2:95)	1:90 (0:36)	1:79 (0:28)
	% SLO Violations	(%)	0:44	n/a	15:09	0:51	0:52
A_{31}	Response Time	(s)	0:59 (0:03)	n/a	0:79 (0:04)	0:59 (0:03)	0:59 (0:03)
	% SLO Violations	(%)	0:55	n/a	2:36	0:55	0:56
A_{32}	Response Time	(s)	0:61 (0:02)	n/a	0:83 (0:02)	0:61 (0:02)	0:61 (0:02)
	% SLO Violations	(%)	0:65	n/a	2:75	0:65	0:66
Uptime		(Ms)	0:97	n/a	1:08	0:97	0:86
Energy Consumption	Total Joules	(MJ)	316:75	n/a	329::06	315:27	297:53
	Wasted Joules	(MJ)	2:25	n/a	25:63	2:23	2:10

The "Uptime" metric is expressed in million of seconds (Ms). The row labeled by "Energy Consumption" reports two energy-related metrics: the total energy consumed, on average, by the cloud infrastructure (label "Total Joules") and an estimate of the total consumed energy that the cloud infrastructure spend, on average, to serve out-of-SLO requests (label "Wasted Joules"). In particular, the "Wasted Joules" metric provides an indication of how efficiently a given approach used physical resources of the cloud infrastructure in order to lower the number of SLO violations and, at the same time, to reduce energy consumption; thus, the lower is its value, the better is the result. Both metrics are expressed in Mega Joules (MJ).

By looking at the results reported in these tables, we can observe that both variants of our approach (columns OUR-APPROACH-NM and OUR-APPROACH-always achieve a lower number of SLO violations (with respect to the 1% threshold defined by SLO specifications), but the S-MMPP scenario, where our approach exceeds the 1% threshold. Moreover, we can note that while both OUR-APPROACH-M and OUR-APPROACH-NM practically result in identical values of SLO violations for all the scenarios, the former always leads to a more efficient usage of physical resources. As a matter of fact, OUR-APPROACH-M always results in lower values of both "Total Energy" and "Wasted Energy" metrics: this means that the exploitation of VM migration allows to both save energy (lower "Total Energy" values) and to better use the energy that is consumed (lower "Wasted Energy" values). Furthermore, the lower value of "Uptime" exhibited by OUR-APPROACH-M means that a smaller amount of physical resource capacity is required to meet SLOs: this means that, potentially, more VMs can be consolidated on the same number of physical machines without increasing the percentage of violated SLOs.[9]

With respect to the other approaches, both variants of our approach always outperform the STATIC-ENERGY and STATIC-TRADEOFF approaches. Further-more, STATIC-ENERGY can be considered worse than STATIC-TRADEOFF since it results in a moderate improvement of (unit) energy consumption at the price of much higher values of SLO violations. Therefore, we can conclude that, with respect to these two approaches, our approach is able to satisfy SLOs for a greater number of requests with a lower energy consumption, in a lower amount of time, and, more importantly, without resulting in any penalty to be paid by the provider.

The comparison with the STATIC-SLO approach needs more attention. First of all, we can observe that, for all scenarios, both our approaches practically exhibit the same values of SLO violations as STATIC-SLO. Second, both approaches keep the percentage of SLO violations under the 1% threshold (as defined by SLO specifications) for all scenarios, except for the S-MMPP one, where, however, also the STATIC-SLO approach exceeds this threshold. Specifically, in the S-MMPP scenario, our approach results in a higher number of SLO violations only for application A_{11}, while, for the remaining applications, our approach and STATIC-SLO practically show the same performance for this metric.[10]

If, however, we look at the efficiency-related metrics, we can observe that our approach, and in particular OUR-APPROACH-M, always results in lower values of "Uptime", "Total Energy" and "Wasted Energy" than STATIC-SLO, indicating that the former is able to consume less energy and to use physical resources more effectively. For instance, in the S-DMPP scenario, the STATIC-SLO approach leads to an energy consumption which exceeds by nearly 7% (i.e., by approximately 20:73 MJ) the one obtained with OUR-APPROACH-M.

Moreover, it is important to observe that STATIC-SLO requires an over commitment of resources, whereby a larger fraction of CPU capacity is assigned to each VM regardless of the fact that this fraction will be actually used by the VM. As a consequence, this implies that the number of VMs that can be consolidated on the same physical machine is lower than those attained by our approach (that, instead, allocates to each VM just the fraction of CPU capacity it needs).[11] Therefore, STATIC-SLO potentially requires, for a given number of VMs, a larger number of physical resources than the our approach one, thus yielding a larger energy consumption.

Finally, it is worth noting that a possible reason to the inability of our approach to always keep the percentage of SLO violations under the 1% threshold in the S-MMPP scenario (e.g., see application A_{11}), is the combination of the temporal burstiness (caused by the MMPP arrival process) with the dynamic execution of application instances. Together, these two factors contribute to the increase of the amount of physical resource demanded, so that if a VM has received a low or medium CPU share (with respect to the reference machine) during a low-intensity workload period, it is possible that it is unable to react in time to such bursts (even if a greater share has been assigned to it), due to the aggregation of too many queueing phenomena and to delays in the reaction time. This is also demonstrated by the behavior of the STATIC-TRADEOFF approach, whereby neither a fixed share of 90% (with respect to the capacity of the reference machine) per VM is able to avoid these situations.[12]

Results for the MM-LOCOPT Experiments Group

The results of the various scenarios, in the MM-LOCOPT group, are presented in four separate tables: Table 7 for S-DMPP, Table 8 for S-PMPP, Table 9 for S-MMPP, and finally Table 10 for S-MIX. As done for the MM-GREEDY group, for the sake of readability, we limit to report only those results deriving from the best combination of RLS algorithm and LQ control design, among all of their variants we considered in this research work..

Results tables are structured as the one presented for the MM-GREEDY group. Specifically, in each table, every column reports the results obtained by the various applications, under a specific resource management approach (i.e., STATIC-SLO, STATIC-ENERGY, STATIC-TRADEOFF, OUR-APPROACH-NM, and OUR-APPROACH-M). A column filled with the symbol "n/a" (which stands for "result not available") means that the use of the corresponding resource management approach made the simulation unable to converge. Numbers inside parenthesis (when present) represent the standard deviations of the related measures, while letters inside parenthesis represent unit of measures (e.g., "(s)" means seconds). The rows of each table have instead the following meaning. Rows labeled by $A_{i\,j}$, for i = 1;:::;3 and j = 1;2, report the 99th percentile of the response time (label "Response Time"), expressed in seconds (s), and the mean percentage of SLO violations (label "% SLO Violations") for each application instance; lower values correspond to better results. The row labeled by "Uptime" reports the sum of the mean uptime of all the physical machines.

Table 7: Experimental evaluation – Results for the S-DMPP scenario in the MM-LOCOPT group.

			Approach				
			STATIC-SLO	STATIC-ENERGY	STATIC-TRADEOFF	OUR-APPROACH-NM	OUR-APPROACH-M
A_{11}	Response Time	(s)	1:17 (0:02)	3:02 (0:02)	1:41 (0:04)	1:18 (0:02)	1:18 (0:02)
	% SLO Violations	(%)	0:62	19:87	1:86	0:62	0:64
A_{12}	Response Time	(s)	1:17 (0:00)	3:08 (0:04)	1:57 (0:01)	1:18 (0:03)	1:19 (0:03)
	% SLO Violations	(%)	0:63	19:73	2:84	0:63	0:64
A_{21}	Response Time	(s)	0:62 (0:01)	1:65 (0:03)	0:81 (0:01)	0:62 (0:01)	0:62 (0:01)
	% SLO Violations	(%)	0:76	14:69	2:72	0:76	0:77
A_{22}	Response Time	(s)	0:63 (0:02)	1:66 (0:01)	0:84 (0:02)	0:63 (0:02)	0:63 (0:02)
	% SLO Violations	(%)	0:76	14:65	2:75	0:76	0:77
A_{31}	Response Time	(s)	0:61 (0:01)	1:68 (0:00)	0:82 (0:00)	0:61 (0:01)	0:61 (0:01)
	% SLO Violations	(%)	0:77	19:30	3:19	0:77	0:77
A_{32}	Response Time	(s)	0:61 (0:00)	1:69 (0:40)	0:83 (0:00)	0:61 (0:00)	0:61 (0:00)
	% SLO Violations	(%)	0:80	19:29	3:20	0:80	0:80
Uptime		(Ms)	0:97	1:18	0:97	0:97	0:87
Energy Consumption	Total Joules	(MJ)	321:67	340:14	319:20	319:47	303:87
	Wasted Joules	(MJ)	2:33	54:79	8:66	2:32	2:23

Table 8: Experimental evaluation – Results for the S-PMPP scenario in the MM-LOCOPT group

			Static-SLO	Static-Energy	Static-TradeOff	Our-Approach-NM	Our-Approach-M
A_{11}	Response Time	(s)	1:19 (0:02)	3:26 (0:03)	1:46 (0:01)	1:19 (0:02)	1:19 (0:01)
	% SLO Violations	(%)	0:58	32:17	1:74	0:59	0:60
A_{12}	Response Time	(s)	1:29 (0:00)	3:41 (0:10)	1:73 (0:01)	1:29 (0:01)	1:29 (0:01)
	% SLO Violations	(%)	0:89	33:89	4:17	0:89	0:88
A_{21}	Response Time	(s)	0:81 (0:23)	1:89 (0:02)	0:86 (0:00)	0:81 (0:23)	0:81 (0:23)
	% SLO Violations	(%)	0:68	18:27	2:68	0:68	0:68
A_{22}	Response Time	(s)	0:68 (0:02)	2:04 (0:32)	0:90 (0:02)	0:68 (0:02)	0:68 (0:01)
	% SLO Violations	(%)	0:82	19:42	3:24	0:82	0:82
A_{31}	Response Time	(s)	0:59 (0:00)	1:46 (0:07)	0:79 (0:00)	0:59 (0:00)	0:59 (0:00)
	% SLO Violations	(%)	0:53	10:45	2:37	0:53	0:53
A_{32}	Response Time	(s)	0:59 (0:01)	1:55 (0:05)	0:82 (0:03)	0:59 (0:00)	0:59 (0:01)
	% SLO Violations	(%)	0:47	11:57	1:84	0:47	0:47
Uptime		(Ms)	0:97	1:24	0:97	0:97	0:90
Energy Consumption	Total Joules	(MJ)	349:47	387:45	345:10	346:23	337:17
	Wasted Joules	(MJ)	2:22	13:32	8:73	2:22	2:16

Table 9: Experimental evaluation – Results for the S-MMPP scenario in the MM-LOCOPT group.

			Static-SLO	Static-Energy	Approach Static-TradeOff	Our-Approach-NM	Our-Approach-M
A_{11}	Response Time	(s)	4:29 (0:17)	n/a	21:10 (1:04)	15:82 (16:00)	4:29 (0:19)
	% SLO Violations	(%)	1:01	n/a	20:11	1:83	1:00
A_{12}	Response Time	(s)	4:45 (0:45)	n/a	24:03 (1:55)	5:12 (1:10)	4:76 (0:90)
	% SLO Violations	(%)	0:86	n/a	19:61	1:21	0:94
A_{21}	Response Time	(s)	2:04 (0:18)	n/a	10:43 (0:16)	2:05 (0:18)	2:04 (0:18)
	% SLO Violations	(%)	0:89	n/a	8:19	0:91	0:89
A_{22}	Response Time	(s)	1:73 (0:35)	n/a	10:93 (0:16)	1:81 (0:45)	1:74 (0:36)
	% SLO Violations	(%)	0:36	n/a	9:90	0:36	0:36
A_{31}	Response Time	(s)	1:82 (0:06)	n/a	15:03 (1:38)	1:82 (0:06)	1:82 (0:06)
	% SLO Violations	(%)	0:67	n/a	13:59	0:67	0:67
A_{32}	Response Time	(s)	1:95 (0:58)	n/a	15:92 (2:81)	1:77 (0:31)	1:96 (0:61)
	% SLO Violations	(%)	0:62	n/a	15:07	0:74	0:64
Uptime		(Ms)	0:97	n/a	1:17	0:97	0:89
Energy Consumption	Total Joules	(MJ)	327:46	n/a	355:89	326:41	315:22
	Wasted Joules	(MJ)	2:63	n/a	17:73	3:48	2:55

Table 10: Experimental evaluation – Results for the S-MIX scenario in the MM-LOCOPT group.

			Static-SLO	Static-Energy	Static-TradeOff	Our-Approach-NM	Our-Approach-M
A_{11}	Response Time	(s)	1:25 (0:00)	n/a	1:54 (0:02)	1:25 (0:00)	1:25 (0:00)
	% SLO Violations	(%)	1:08	n/a	3:12	1:09	1:09
A_{12}	Response Time	(s)	1:25 (0:01)	n/a	1:69 (0:01)	1:26 (0:01)	1:26 (0:01)
	% SLO Violations	(%)	1:03	n/a	4:38	1:03	1:03
A_{21}	Response Time	(s)	2:02 (0:03)	n/a	35:28 (19:60)	2:29 (0:40)	2:29 (0:45)
	% SLO Violations	(%)	0:90	n/a	20:91	0:95	0:92
A_{22}	Response Time	(s)	1:76 (0:19)	n/a	16:65 (2:41)	1:74 (0:21)	1:71 (0:20)
	% SLO Violations	(%)	0:47	n/a	15:12	0:46	0:40
A_{31}	Response Time	(s)	0:59 (0:03)	n/a	0:79 (0:04)	0:59 (0:03)	0:59 (0:03)
	% SLO Violations	(%)	0:55	n/a	2:36	0:56	0:56
A_{32}	Response Time	(s)	0:61 (0:02)	n/a	0:83 (0:02)	0:61 (0:02)	0:61 (0:02)
	% SLO Violations	(%)	0:65	n/a	2:75	0:66	0:65
Uptime		(Ms)	0:97	n/a	0:89	0:97	0:89
Energy Consumption	Total Joules	(MJ)	334:05	n/a	318:72	333:46	320:01
	Wasted Joules	(MJ)	2:69	n/a	25:10	2:73	2:58

III. RESULTS & DISCUSSION

This research work emphasizes on the fact wherein the uptime of a physical machine is characterized as the aggregate time (from the earliest starting point of every reproduction imitation) that the physical machine has been controlled on. This metric measures the productivity of a given approach in utilizing the physical machines of the cloud foundation, since bring down "Uptime" values show the utilization of a lower measure of physical asset ability to serve a given workload. The "Uptime" metric is communicated in million of seconds (Ms). The line marked by "Vitality Consumption" reports two vitality related measurements: the aggregate vitality devoured, by and large, by the cloud framework (name "Add up to Joules") and a gauge of the aggregate expended vitality that the cloud foundation spend, all things considered, to serve out-of-SLO asks for (name "Squandered Joules"). Specifically, the "Squandered Joules" metric gives a sign of how proficiently a given approach utilized physical assets of the cloud infrastructure so as to bring down the quantity of SLO infringement and, in the meantime, to decrease vitality utilization; therefore, the lower is its esteem, the better is the outcome. The two measurements are communicated in Mega Joules (MJ).

By taking a gander at the outcomes revealed in these tables, we can watch that lone in the S-DMPP and S-PMPP situations (see Table 7 and Table 8, respectively), the two variations of our approach dependably accomplish a lower number of SLO infringement (as for the 1% limit characterized by SLO details). For the other two situations, the OUR-APPROACH-M displays preferred execution over OUR-APPROACH-NM. In actuality, in the S-MMPP situation, the OUR-APPROACH-M accomplishes great execution, while that of OUR-APPROACH-NM are exceptionally poor since (1) there is a high fluctuation in the circulation of reaction time (e.g., see application A11), and (2) the quantity of SLO infringement exceeds the endorsed edge of 1% (e.g., see applications A11 and A12). For what concerns the S-MIX situation, the execution acquired by OUR-APPROACH-M is superior to the one of OUR-APPROACH-NM, since, regardless of the two methodologies about show similar outcomes in term of SLO infringement, the previous likewise prompts a superior lessening of vitality utilization.

Concerning the STATIC-ENERGY and STATIC-TRADEOFF approaches, the two variations of our approach are constantly ready to beat them. In actuality, regarding these two methodologies, our approach can fulfill SLOs for a more prominent number of solicitations with a lower vitality utilization and, all the more imperatively, without coming about (more often than not) in any punishment to be paid by the supplier.

The correlation with the STATIC-SLO approach needs more investigation. Right off the bat, regarding our approach, the STATIC-SLO approach is constantly ready to accomplish comparable (and once in a while better) execution as far as rate of SLO violations. In any case, for all situations, such better conduct is not helpful since it doesn't give any profit to the IaaS cloud supplier due to the more noteworthy measure of devoured vitality. This can be watched both from the "Aggregate Energy" and the "Squandered Energy" measurements.[13] For example, in the S-DMPP situation, the STATIC-SLO approach prompts a vitality utilization which surpasses by about 6% (i.e., by roughly 17:73 MJ) the one got with OUR-APPROACH-M. In this case, behavior like the one showed by OUR-APPROACH-M is more attractive, since, overall, permits to spare vitality, while as yet keeping the quantity of SLO infringement under the recommended edge.

GAt long last, it is imperative to take note of that between the two variations of our approach there is no a reasonable champ. All in all, them two can keep the quantity of SLO infringement low, while restricting the vitality utilization. Be that as it may, as officially watched for the MM-GREEDY examinations gathering, the OUR-APPROACH-M variation is more viable in diminishing the vitality utilization and in attempting to accomplish SLO imperatives, particularly in high-force workload situations like S-MMPP (see Table 9).

Concluding Remarks

In this area, we think about the viability of utilizing VM relocation and the conduct of the two unique executions of the Migration Manager, as for the analyses displayed in the above areas.

Viability of VMs Migration. From the aftereffects of the examinations just de-scribed, we can presume that the utilization of the Migration Manager (and in this way of VM movement) keeps the rate of SLO infringement under (or practically to) the endorsed edge of 1%, even under high-power bursty workloads (e.g., like in the S-MMPP situation of the MM-GREEDY gathering – see Table 5).

Also, notwithstanding when the subsequent execution are similar with the one obtained without the Migration Manager, the utilization of the Migration Manager prompts a superior decrease of vitality utilization (e.g., like in the S-PMPP situation of the MM-GREEDY gathering – see Table 4).

Insatiable versus Local Optimization Implementation. From the consequences of the above analyses, we can take note of that the conduct of the two usage is fundamentally the same as. Be that as it may, we can attempt to play out an exhaustive correlation by watching the accompanying realities:

S-DMPP situation: for roughly a similar rate of SLO violations, the voracious usage can prompt a marginally bring down vitality utilization, along these lines bringing about a (somewhat) bring down misuse of Joules. [14]

S-PMPP situation: the execution of the two usage are somewhat tantamount, regardless of the possibility that the neighborhood advancement usage is by all accounts ready to marginally squander less Joules.

S-MMPP situation: the points of interest brought by one usage over the other one are not all that reasonable. On the hand, for the OUR-APPROACH-NM approach, the ravenous usage performs superior to the nearby streamlining one, since it can accomplish a lower number of SLO infringement and, in the meantime, to prompt a lower misuse of Joules. Then again, for the OUR-APPROACH-M approach, the nearby advancement usage demonstrates preferable execution over the avaricious one, since it can keep the rate of SLO infringement under (or practically to) the endorsed limit and, in the meantime, to prompt a lower misuse of Joules.

S-MIX situation: the ravenous execution performs superior to the neighborhood improvement one, since it can simply keep the rate of SLO infringement under the predefined edge and, in the meantime, to prompt a more prominent decrease of vitality consumption. From these certainties, it comes about that, for the situations mulled over, the eager usage performs superior to anything the one in view of nearby streamlining.

This can be credited to the way the target work has been defined. In reality, curiously, the neighborhood improvement calculation begins the scan for the (nearby) ideal just from the arrangement processed by the ravenous calculation.[15] Since the arrangement acquired by the nearby improvement technique is never more terrible than the one gave as beginning stage, this may imply that the target work does not catch all the required flow that permit to endeavor the joint objective of energy utilization minimization and execution enhancement.

REFERENCES

[1.] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In Proc. of the 2nd Symposium on Networked Systems Design & Implementation (NSDI'05), pages 273–286. USENIX Association, 2005.

[2.] D.W. Clarke and R. Hastings-James. Design of digital controllers for ran-domly disturbed systems. In Proc. of the Institution of Electrical Engineers, pages 1503–1506, Oct 1971.

[3.] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association, 74(368):829– 836, 1979.

[4.] William S. Cleveland and Susan J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. Journal of the American Statistical Association, 83(403):596–610, 1988.

[5.] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms. MIT Press, 3rd edition, 2009.

[6.] R. J. Creasy. The origin of the VM/370 time-sharing system. IBM Journal of Research and Development, 25(5):483–490, Sep 1981.

[7.] R.J. Dakin. A tree search algorithm for mixed integer programming prob-lems. The Computer Journal, 8(3):250–255, 1965.

[8.] Richard C. Dorf and Robert H. Bishop. Modern Control Systems. Prentice Hall, 12th edition, 2010.

[9.] Arne S. Drud. CONOPT: A large-scale GRG code. Journal on Computing, 6(2):207–216, 1994.

[10.] Marco Duran and Ignacio Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming, 36(3):307–339, 1986.

[11.] ENERGY STAR Program. Report to congress on server and data center energy efficiency. Technical report, U.S. EPA, Aug 2007.

[12.] Thomas Erl. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005.

[13.] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. Power provisioning for a warehouse-sized computer. In Proc. of the 34th International Symposium on Computer Architecture (ISCA'07), pages 13–23, 2007.

[14.] R. Figueiredo, P. A. Dinda, and J. Fortes. Resource virtualization renaissance. In Proc. of the IEEE Internet Computing, volume 38, May 2005.

[15.] Władysław Findeisen, F.N. Bailey, M. Bryds, K. Malinowski, P. Tatjewski, and A. Wozniak. Control and Coordination in Hierarchical Systems. John Wiley & Sons, Ltd, 1980.

[16.] Wolfgang Fischer and Kathleen Meier-Hellstern. The Markov-modulated Poisson Process (MMPP) cookbook. Performance Evaluation, 18(2):149– 171, 1993.

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with Sl. No. 3822, Journal no. 43302.