A Power-aware Cost Model for HPC Procurement

Neha Gholkar[1], Frank Mueller[1], Barry Rountree[2]
[1]North Carolina State University, USA, mueller@cs.ncsu.edu
[2]Lawrence Livermore National Laboratory, USA, rountree1@llnl.gov

Abstract

With the supercomputing community headed toward the era of exascale computing, power has become one of the foremost concern. Today's fastest supercomputer, Tianhe-2, already consumes 17.8MW to achieves a peak performance of 33.86PFlops [1]. At least an order of magnitude improvement in performance while maintaining the power envelope is required for exascale. Yet, manufacturing variations are increasingly creating a heterogeneous computing environment, even when identical processing components are deployed, particularly when operating under controlled power ceiling.

This work contributes a procurement model to aid in the design of a capability system that achieves maximum performance while considering manufacturing variations. It appropriately partitions a single, compound system budget into the CAPEX (infrastructure cost) and the OPEX (operating power cost). Early results indicate that aggressive infrastructure procurement disregarding such operational needs can lead to severe performance degradation, or significant hidden operating cost will be incurred after procurement.

1. Introduction

As we approach exascale, a supercomputer is expected to cost about $200 million and use only 20 megawatts of power in achieving an exaflop [4]. Considering the least expensive power in the U.S. ($\approx$ 5 cents/kWh) [17], the cost of operating this machine is $8.76 million per year. Considering a lifetime of five years, its operating cost is about $45 million or nearly 25% of the total cost of ownership (TCO).

The status quo of supercomputing procurements is that the power is provisioned for the worst case (WCP). After the initial burn-in phase, which often entailed a Linpack run among other application acceptance tests, the power utilization of the machine drops to 61% of the procured power [13]. This implies that the infrastructure put in place for peak power is no longer fully utilizing this power. Such overprovisioning can be considered a bad investment of budget during steady state operation.

Our prior work on power-efficient computing under manufacturing variations shows that processors are not most power efficient at WCP [8]. Hence, a system with WCP procurement is not power efficient. Processors achieve peak power efficiency at disparate power bounds due to manufacturing variability [14]. Efficient processors achieve peak power efficiency at lower power bounds than the inefficient processors. Increasing the operating power of the processors beyond these bounds leads to diminishing returns in terms of performance. Instead, some power of these processors can be redirected to additional processors to get better performance. Given a hard power constraint, we proposed PTune, a power tuner that exploits this concept to maximize the performance of the system while staying within the power budget.

In this work, we propose a procurement strategy to design a machine that achieves maximum performance per dollar by determining the optimal partitioning of the total budget into capital expenditure (CAPEX) and operating expenditure (OPEX). For the scope of this paper, we limit the OPEX to the cost of power procured for the lifetime of the machine.

Previous research [9], [19], [20], [10] has developed cost models for predicting the total cost of ownership (TCO) for cloud computing centers and datacenters. These models do not share our objective of maximizing performance under a fixed system (dollar) budget. In recent work [11], Patki has studied the effect of adding more infrastructure to the system under a fixed power budget. System-wide solutions for power constraint systems have been proposed that aim at increasing the throughput of systems and the runtime of the jobs under a fixed power budget [7], [12], [15], [16], [6], [13], [5]. Unlike all of this work, we consider the system's power budget as a variable expenditure that we balance against the infrastructure cost to determine the break down of the system's total budget that leads to maximum performance. Furthermore, our model takes the effects of manufacturing variation on the processor's performance into account.

The paper is organized as follows. Section 2 states the problem statement. Section 3 gives an overview of the cost model. Sections 4 describes our procurement strategy. Section 5 presents the modeling results. Section 7 summarizes the contributions.

2. Problem Statement

Given a budget, Sys_Budget, for a system acquisition, design a machine for optimal performance under the assigned budget. The total budget can be divided into two main variable budgets: (1) CAPEX or the cost of the infrastructure ($Sys\_Infrastructure\_Cost$); and (2) OPEX or the cost of power (Sys_Power_Cost) .

We propose a procurement strategy of building a system that achieves maximum performance under the total budget by appropriately partitioning the budget into capital expenditure and operating expenditure. To model the system, we quantify performance in terms of instructions retired per second (IPS). System's performance is represented by $SysIPS$. Our objec-

tive is to

$$Maximize(SysIPS) \quad (1)$$

subject to $Sys\_Budget \geq Sys\_Power\_Cost + Sys\_Infrastructure\_Cost.$

3. Cost Model

To demonstrate the model, let us consider a system of Intel Ivy Bridge (12 core Xeon E5-2697 v2 2.7GHz) processors. We make a number of assumptions about the system costs to simplify the problem.

Capital Expenditure (CAPEX):. CAPEX is the cost of the physical assets ($Sys\_Infrastructure\_Cost$). We limit CAPEX to the cost of purchasing racks. The cost of a rack ($Rack\_Infrastructure\_Cost$) is assumed to be \$366K. This number is derived from the data for Jaguar [2], [3]. A rack can host a maximum of 100 server nodes, where each node has two processor sockets. In this paper, we assume all racks are identical, i.e., all the racks have the same set of processors and, hence, the same processor characteristics. We assume a variation of 30% in performance across the processors of a rack due to manufacturing variability [8].

Operational Expenditure (OPEX):. OPEX mainly consists of the fraction of the budget spent on power (Sys_Power_Cost) required to run the system for a predetermined fixed duration. For simplicity, we limit ourselves to the power associated with computing; networking and I/O is subject to future work, and so are secondary operational costs, such as maintenance and support, including staff (typically subject to a separate budget). We assume the minimum power required by the nodes ($P_{node,min}$) is 110W while the maximum power ($P_{node,max}$) corresponding to the Thermal Design Power (TDP) of the processors is 260W. We assume a flat power power of 5 cents per kWh [17]. Finally, we assume the system's lifetime of 5 years ($T = 5 \times 365 \times 24$ hours). This implies that the total cost of acquiring and operating the machine for at least 5 years should not exceed the total budget.

3.0.1. Workload:. A supercomputing workload consists of multiple and often coupled parallel scientific simulations that execute on several processors simultaneously. In the interest of simplicity, we assume that the system runs a single application over the duration of its lifetime. To assess the differences between codes, we study the effect of CAPEX and OPEX on performance for 4 different codes, viz., EP, SP, BT from the NAS parallel benchmark suite and CoMD from the Mantevo suite, in the modeling results.

4. Procurement Strategy

We propose a procurement strategy that builds a system with maximum performance under an assigned total budget. We assume that a rack is always filled to capacity with compute nodes, i.e., it hosts 200 processors. It can be powered at:

- Maximum Power: A rack is supplied with worst case power (WCP) to be able to run the processors at their Thermal Design Power (TDP).
- Minimum Power: A rack is supplied with minimum power required by the computation capability to be functional.
- Medium Power: A rack is supplied with less than required maximum power but greater than the bare minimum power required by the computation capability it hosts.

The performance (and cost) of a rack is maximal under maximum power configuration. The maximum power that can be provisioned to a rack ($Rack\_max\_power$) is $100 \times P_{(node,max)}$. Its power cost is $100 \times P_{(node,max)} \times \$0.05 perkWh \times T = \$56,940$. Therefore, $Rack\_TotalCost_{(max)} = \$56,940 + \$366,197 = \$423,137$.

$Rack\_TotalCost_{(min)}$ represents the minimum budget required to add a rack to the system. The infrastructure cost is the same as that of a packed rack. Its power cost is $100 \times P_{(node,min)} \times \$0.05 perkWh \times T = \$21,900$. Therefore, $Rack\_TotalCost_{(min)} = \$22,338 + \$366,197 = \$388,535$.

Given a fixed Sys_Budget, the number of racks in the system (num_rack) that can be procured ranges between $num\_rack_{\perp}$ and $num\_rack_{\top}$.

The lower bound on num_rack ($num\_rack_{\perp}$) is $num\_rack_{\perp} = \frac{Sys\_Budget}{Rack\_TotalCost_{(max)}}$.

The upper bound on num_rack ($num\_rack_{\top}$) is $num\_rack_{\top} = \frac{Sys\_Budget}{Rack\_TotalCost_{(min)}}$.

For num_rack racks, the $Sys\_Infrastructure\_Cost$ is $num\_racks \times Rack\_Infrastructure\_Cost$.

The Sys_Power_Cost is calculated as

$$Sys\_Power\_Cost = Sys\_Budget - Sys\_Infrastructure\_Cost.$$

The Sys_Power is calculated as

$$Sys\_Power = min(\frac{Sys\_Power\_Cost}{\$0.05 \ per \ kWh \times T}, num\_rack \times Rack\_max\_power).$$

The $Rack\_Power$ is calculated as $Rack\_Power = \frac{Sys\_Power}{num\_rack}$.

At rack-level, for an assigned power budget of $Rack\_Power$, PTune[8], our variation-aware power tuner, determines the power distribution across the processors that achieves maximum performance within this rack.

$Rack\_Performance = PTune(Rack\_Power)$

As all racks are considered to be identical (statistically, given their large processor count), the system's performance can be calculated as

$SysIPS = num\_rack \times Rack\_Performance.$

Given a $Rack\_Power$ budget, PTune[8] distributes the budget across the processors of the rack systematically in a variation-aware, i.e., processor-sensitive, manner to maximize the performance of this rack. A system configuration can be represented as a tuple over (1) the number of racks in the system, (2) power allocated to a rack, and (3) the resulting performance of a system, denoted as ($< num\_racks, Rack\_power, SysIPS >$). The winning design is the one that achieves the objective of Eq.1.

5. Experimental Setup

Characterization experiments were conducted on the *Catalyst* cluster, a 324-node Ivy Bridge cluster at Lawrence Livermore National Laboratory (LLNL). We used the performance and power data from this entire cluster to represent one rack in the machine that we designed in the previous section. Each node has two 12-core Intel(R) Xeon(R) CPU E5-2695 v2 @ 2.40GHz processors and 128 GB of memory. We used MVAPICH2 version 1.7. The codes were compiled with the Intel compiler version 12.1. The msr-safe kernel module provides direct access to Intel RAPL registers via libmsr [18]. We used the package (PKG) domain of RAPL that provided us the capability of capping power for each of the processors in an experiment. The environment was simulated in R. We used EP, BT, and SP from the NPB suite and CoMD from the Mantevo suite in their pure MPI versions.

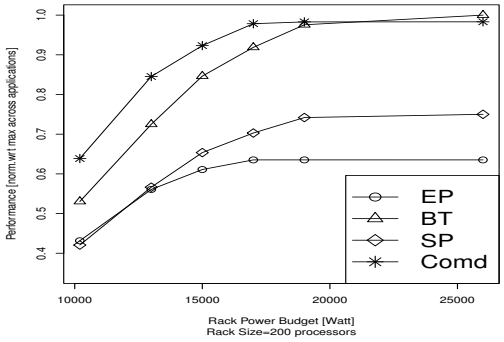

Fig. 1: PTune: Power Tuning Results for a rack at several rack power budgets.

6. Results

We observe the impact of different breakdowns of the system's budget (into CAPEX and OPEX) on the performance of the system. At the rack-level, we use PTune [8] to get the maximum performance under the rack's power budget. Fig. 1 depicts the performance of a rack at several power budgets. The x-axis represents the rack's power budget in watt and the y-axis represents the rack performance normalized to the maximum performance across applications. We show results for EP, SP, BT, and Comd. Data points corresponding to 26kW represent the performance at maximum power ($Rack\_max\_power$). Every data point represents the performance corresponding to variation-aware power tuning of the rack power across processors. We observe that the performance increases non-linearly with rack power.

Fig. 2 shows the performance of a system under a fixed budget of $102 million. The x-axis represents the number of racks and the y-axis represents the system performance normalized to the optimal performance. The optimal design at 252 racks achieves 5% better performance compared to the WCP. Purchasing more racks beyond this point (and effectively procuring less power) leads to suboptimal capability. However,

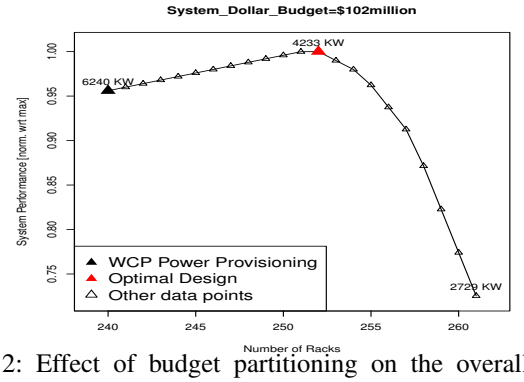

Fig. 2: Effect of budget partitioning on the overall system performance

increasing the number of racks up to 255 gives at least as much performance as the WCP design but addition capacity.

Purchasing more than 255 racks degrades the capability of the machine by up to 29% compared to the optimal design. No more racks can be added beyond this point as the system budget will not suffice to provision the bare minimum power required to power the system. It is important to note that rack $(n+1)$ is procured at the expense of the power stolen from the prior n racks. Since the racks are always fully packed, their CAPEX is fixed. Hence, the only way to accommodate an addition rack is to reduce the OPEX of the prior n racks. From this, we conclude that aggressively purchasing infrastructure under a fixed system budget by disregarding the diminishing budget for power does *not* lead to the best capability system design. A *balance* needs to be stricken between the CAPEX and the OPEX for an optimal system design.

Figures 3, 4, and 5, compare the results for 3 different codes under several system budgets. (Results for SP are similar to EP and are omitted due to space). The x-axis represents number of racks and the y-axis represents system performance. The figure shows 7 system budgets that correspond to the cost of 40, 80, 120, 160, 200, 240, 280, and 320 racks at maximum power. The first data point in each system budget curve represents the performance at WCP.

Overall, we make the following observations for these figures:

- Power-aware procurement can improve performance by up to 4%, 6%, 7%, and 4% for EP, SP, BT, and Comd, respectively, compared to worst-case power provisioning.
- Without power-aware procurement, performance can degrade up to 25%, 28%, 41%, and 38% for EP, SP, BT, and Comd, respectively, compared to worst-case power provisioning.
- Performance increases linearly with the system budget.
- Performance linearly increases with the number of racks before it reaches the peak, after which there is a steep performance degradation. Peak performance is achieved relatively early in case of BT (Fig. 4) compared to EP and CoMD (Figures 3 and 5). BT has a longer tail that follows the peak, representing the feasible system designs that lead to degraded system performance due to aggressive infrastructure procurement.
- The absolute performance achieved by the system depends

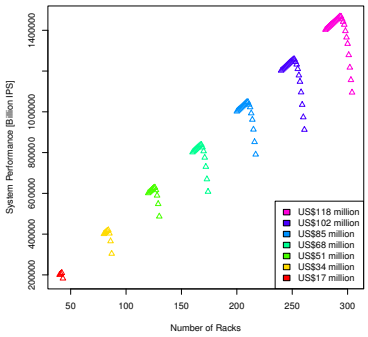

Fig. 3: EP

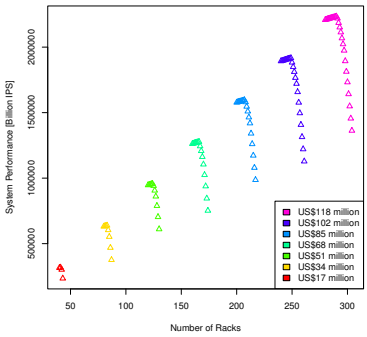

Fig. 4: BT

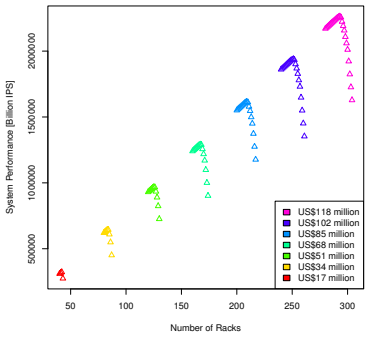

Fig. 5: Comd

upon the application, i.e., the optimal system design also depends on the application. The assumption that our system executes a single application during the course of its lifetime does not hold in reality. Hence, it is necessary to compromise on middle ground with a design that, on *average*, fits the needs of the applications when run in capability mode.

7. Summary

This work analyzed the effect of manufacturing variations on procurement and operations. It showed that when partitioning the system's budget into infrastructure cost and power cost, a balance needs to be stricken between the two to achieve an optimal ratio of performance per cost (dollar). More infrastructure does not necessarily mean more performance under a fixed total budget. Model-based analysis provides the means to optimize power-aware procurement/operation for a set of application codes. Such strategies may need to be adopted in the future to best utilize a compound budget for systems and operations. Failure to plan ahead may either result in a nominal loss in performance compared to such a balanced system, or a significant additional cost will be incurred in operating cost to efficiently utilize an overprovisioned hardware installation.

8. Acknowledgements

This work was supported in parts by NSF grants 552803, 555237, 0958311, the Consortium for Advanced Simulation of Light Water Reactors (CASL), the U.S. Department of Energy's Lawrence Livermore National Laboratory. Office of Science, under Award number DE-AC52-07NA27344 and Office of Science, Office of Advanced Scientific Computing Research (LLNL-CONF-656877).

References

[1] Top 500 list. http://www.top500.org/, June 2002.

[2] Ten of the coolest and most powerful supercomputers of all time. 2009. http://royal.pingdom.com/2009/06/11/10-of-the-coolest-and-most-powerful-supercomputers-of-all-time/.

[3] Ten of the coolest and most powerful supercomputers of all time. 2009. https://en.wikipedia.org/wiki/Jaguar_%28supercomputer%29.

[4] PCWorld. 2015. http://www.pcworld.com/article/2014715/next-up-exascale-computers-expected-to-arrive-by-2020.html.

[5] D. A. Ellsworth, A. D. Malony, B. Rountree, and M. Schulz. Pow: System-wide dynamic reallocation of limited power in hpc. In *Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '15, pages 145–148, New York, NY, USA, 2015. ACM.

[6] M. E. Femal and V. Freeh. Boosting data center performance through non-uniform power allocation. In *International Conference on Autonomic Computing*, pages 250–261, 2005.

[7] M. E. Femal and V. W. Freeh. Safe overprovisioning: using power limits to increase aggregate throughput. In *In International Conference on Power-Aware Computer Systems*, December 2005.

[8] N. Gholkar, F. Mueller, and B. Rountree. Power tuning for hpc jobs under manufacturing variations. In *Technical Report TR 2016-2, Department of Computer Science, North Carolina State Univeristy*, February 2015.

[9] M. Heikkurinen, S. Cohen, F. Karangiannis, K. Iqbal, and S. Andreozzi. Answering the Cost Assessment Scaling Challenge: Modelling the Annual Cost of European Computing Services for Research. In *Journal of Grid Computing*, 2015.

[10] J. Koomey, B. Kenneth, P. Turner, J. Stanley, and B. Taylor. A simple model for determining true total cost of ownership for data centers. In *Uptime Institute White Paper, Version 2*, 2007.

[11] T. Patki. *The Case for Hardware Overprovisioned Supercomputers*. PhD thesis, University of Arizona, July 2015.

[12] T. Patki, D. K. Lowenthal, B. Rountree, M. Schulz, and B. R. de Supinski. Exploring Hardware Overprovisioning in Power-constrained, High Performance Computing. In *International Conference on Supercomputing*, pages 173–182, 2013.

[13] T. Patki, D. K. Lowenthal, A. Sasidharan, M. Maiterth, B. Rountree, M. Schulz, and B. R. de Supinski. Practical Resource Management in Power-Constrained, High Performance Computing. In *HPDC*, 2015.

[14] B. Rountree, D. H. Ahn, B. R. de Supinski, D. K. Lowenthal, and M. Schulz. Beyond DVFS: A First Look at Performance under a Hardware-Enforced Power Bound. In *IPDPS Workshops*, pages 947–953. IEEE Computer Society, 2012.

[15] O. Sarood. *Optimizing Performance Under Thermal and Power Constraints for HPC Data Centers*. PhD thesis, University of Illinois, Urbana-Champaign, December 2013.

[16] O. Sarood, A. Langer, A. Gupta, and L. V. Kale. Maximizing throughput of overprovisioned hpc data centers under a strict power budget. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '14, New Orleans, LA, 2014. ACM.

[17] J. Shalf, S. Dosanjh, and J. Morrison. Exascale computing technology challenges. In *VECPAR*, 2010.

[18] K. Shoga, B. Rountree, M. Schulz, and J. Shafer. Whitelisting msrs with msr-safe. In *3rd Workshop on Extreme-Scale Programming Tools at SC*, Nov. 2014. http://www.vi-hps.org/upload/program/espt-sc14/vi-hps-ESPT14-Shoga.pdf.

[19] B. Tak, B. Urgaonkar, Sivasubramaniam, and Anand. To move or not to move: The economic of cloud computing. In *USENIX Conference on Hot Topics in Cloud Computing*, 2011.

[20] E. Walker. The real cost of a cpu hour. In *USENIX Conference on Hot Topics in Cloud Computing*, 2009.