Power-capping Aware Checkpointing: On the Interplay among Power-capping, Temperature, Reliability, Performance, and Energy

Kun Tang[1], Devesh Tiwari[2], Saurabh Gupta[2], Ping Huang[1], Qiqi Lu[1], Christian Engelmann[2], and Xubin He[1]

[1]Virginia Commonwealth University [2]Oak Ridge National Laboratory

***Abstract*—Checkpoint and restart mechanisms have been widely used in large scientific simulation applications to make forward progress in case of failures. However, none of the prior works have considered the interaction of power-constraint with temperature, reliability, performance, and checkpointing interval. It is not clear how power-capping may affect optimal checkpointing interval. What are the involved reliability, performance, and energy trade-offs? In this paper, we develop a deep understanding about the interaction between power-capping and scientific applications using checkpoint/restart as resilience mechanism, and propose a new model for the optimal checkpointing interval (OCI) under power-capping. Our study reveals several interesting, and previously unknown, insights about how power-capping affects the reliability, energy consumption, performance.**

I. Introduction

The continuous growth in computing capability has expedited the scientific discovery and enabled scientific applications to simulate physical phenomena for increased problem sizes [1], [2]. However, as the computing scale becomes larger, the likelihood of failures also increases. Failures prevent scientific applications from making forward progress. To address this problem, scientists typically employ checkpoint-restart mechanisms to guarantee forward progress of the simulation in case of failures [3]–[5]. Checkpointing is a periodic process that writes the "required-to-recover" application state to the permanent storage system. When a failure occurs, the application can restart from the latest checkpoint. Although checkpoint and restart mechanisms can keep scientific simulations moving forward, writing and reading application state incurs huge I/O overhead, which also impedes the scientific productivity [6], [7]. At exascale, this overhead is anticipated to increase further. In fact, it is estimated that in some cases applications may end up spending more than 50% of total execution time on checkpoint, restart, and lost work [8], [9].

The checkpointing process has its own trade-off in terms of performance and I/O overhead. A small checkpointing interval leads to high checkpointing I/O overheads while a large interval checkpointing may result in high wasted work if a failure occurs. It is important to checkpoint at the optimal checkpointing interval (OCI) – a problem well-studied for the last several decades. Young [10] proposed a first order approximation to the optimal checkpointing interval. Daly [11], [12] derived a high order estimation of the optimal checkpointing interval. There have been several other studies proposing finer refinements to these models, but none of the prior works have considered the interaction of power-constraint with checkpointing interval. Power consumption is becoming a first-order concern for high performance computing (HPC) facilities. Therefore, efficient operation of these facilities requires power-constraint to be taken into account at all layers. Power capping essentially limits the maximum allowable power consumption of a platform, potentially impacting temperature, reliability, performance and energy-efficiency. However, it is not clear how does power capping affect OCI. What are the involved temperature, reliability, performance, and energy trade-offs?

To the best of our knowledge, no prior study has investigated how power capping affects the checkpointing decisions for scientific applications in a large-scale HPC computing facility. Therefore, the goal of this paper is to develop an understanding about the interaction between power capping and scientific applications relying on checkpoint/restart. Our study is based on real-system experiments, analytical models, and statistical techniques. This work is driven by data obtained from the real large-scale computing facility. In particular, this work makes the following contributions.

Contributions: First, we study the effect of power capping on compute and checkpointing phase for a variety of scientific applications using a widely-used checkpoint library (Berkeley Lab Checkpoint/Restart) [3]. We also demonstrate and quantify how power capping affects the system reliability due to change in temperature. Second, we propose a new model for optimal checkpointing interval (OCI) under power capping effects. Our model derives OCI for both execution time and energy consumption. Third, we validate our model, and present model and simulation driven results for a wide range of scenarios. We show that the proposed model is significantly more accurate than previously proposed power capping unaware models. Compared to the previously proposed models, our model results in significant time and energy savings for both peta- and exa-scale systems, with different checkpointing costs, wide range of power caps, and for different application and system characteristics. Our evaluation also shows that the proposed model can save up to 18% of energy and execution time for a set of leadership applications run at the Oak Ridge National Laboratory. It also reduces the amount of data movement by up to 57% for these large-scale applications.

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan. This work is sponsored in part by U.S. National Science Foundation grants CCF-1547804, CNS-1320349 and CNS-1218960. This work is also supported by the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory.

Our study also reveals several interesting, and previously unknown, insights about how power capping affects temperature, reliability, energy consumption, and performance of large-scale leadership applications[1] in the presence of system failures and checkpoint/restart. We believe that insights derived from this work carry significant implications for data center facilities, researchers focusing on resilience, and end users.

II. Related Work

This paper investigates how the power capping affects the application performance, system reliability, and the optimal checkpointing decisions. There are many ways to achieve power capping effects. Power capping is generally achieved through either Dynamic Voltage and Frequency Scaling (DVFS) [13], or throttling by idle cycle insertion [14]. The power capping method we used is the Intel Power Governor, which utilizes the throttling technique [15] [16]. There have been studies to utilize power capping for enterprise workloads. Power gating is a possible way to reduce the power consumption through shutting down certain cores. For example, Ma et al. [17] proposed to shut down idle cores and boost performance of the other cores through frequency scaling. Recent works have also explored applying DVFS for I/O intensive task such as garbage collection [18].

Recognizing the importance of checkpointing, researchers have long-investigated methods to reduce the checkpointing overheads. Some studies have focused on reducing checkpoint data size itself, for example via designing incremental checkpointing schemes [19]. Some researchers have proposed diskless checkpointing through redundancy, such as replication [20]. Compared with the disk-based checkpointing techniques, diskless checkpointing consumes excessive computing resources, including processor, memory, network, etc. Finally, one of the most widely used ways to reduce checkpointing overhead has been to derive optimal checkpointing interval. Young [10] proposed a first order approximation to the optimum checkpointing interval based on the assumption that system failures follow a Poisson process. Based on the first order model, Daly [11], [12] proposed a high order estimation of the optimum checkpointing interval. The high order model can predict the OCI more accurately when mean time between failures (MTBF) becomes smaller. Recently, some works have taken failure characteristics into account to tune optimal checkpointing interval [8], and make checkpointing more energy-efficient [21]. However, none of these works investigate the impact of power capping on checkpointing for large-scale HPC applications.

III. Background and Methodology

Our work is primarily modeled after and based on Titan, No. 2 on the Top 500 supercomputer list. Titan consists of 18,688 compute nodes (CPU and GPU) and more than 700 TB memory capacity. Titan's theoretical peak performance is approx. 27 Petaflops. We have also included various system design points to show the relevance and impact of our insights, and proposed model for the future exascale systems.

[1]Leadership-scale computing refers to supercomputers facilitated by the Department of Energy, and we refer to the large-scale application run on these supercomputers as leadership applications.

System failure related data to validate our model and drive our simulation studies has been collected from the Oak Ridge Leadership Computing Facility. This data represents Titan's failure log data for over two years since production. We evaluate the impact of our proposed model on different leadership applications. In Table I, we show the checkpoint size and run time for such applications based on traditional hourly checkpoints.

TABLE I: Characteristics of leadership applications [8].

Application Name	Scientific Domain	Checkpoint Data Size	Application Run-time
CHIMERA	Astrophysics	160 TB	360 Hours
GTC	Fusion	20 TB	120 Hours
GYRO	Fusion	50 GB	120 Hours
POP	Climate	26 GB	480 Hours
S3D	Combustion	5 TB	240 Hours
VULCUN/2D	Astrophysics	0.83 GB	720 Hours

In order to study the impact of power capping on the optimal checkpointing interval, we combine experiments, simulations, and model analyses in this paper. First, we obtain performance and temperature data under power capping on small-scale machines. Second, we develop our OCI model based on the regression analysis. Then, we validate our OCI model through simulations, and evaluate the model. Finally, we show a case study for our model at large scale based on leadership application runs.

We point out that performing power capping experiments is not possible on the Titan supercomputer's AMD CPUs because power-capping is not supported on these platforms. To overcome this limitation, we choose two Xeon platforms to drive our study and gain insights. The two Xeon platforms are E5-2670 and E5-2630. E5-2670 platform has 8 cores each clocked at 2.6 GHz with 115 watts Thermal Design Power (TDP) and 166.4 flops of double floating point peak performance. On the other hand, E5-2630 platform has only 6 cores each clocked at 2.3 GHz with 95 watts TDP and 110.4 flops of double floating point peak performance. Both platforms have 64 GB of DRAM and running Linux 2.6.32 kernel with GCC 4.4.6 compiler installed. Also, since these platforms cannot run large-scale applications that depend on Cray linux environment and platform specific libraries, we run a wide variety of scientific applications for understanding power capping effects, taken from Rodinia benchmark suite and NPB benchmark suite (Table II). These benchmarks cover a variety of science domains and were characterized using TAU profiling tool [22] and PAPI counters [23] to ensure that they represent a wide range of architectural characteristics.

We use BLCR [3], a widely used system-level checkpointing library, to perform system level checkpointing on scientific applications. Performance is measured as the reciprocal of execution time. Each processor runs at its full capacity and utilizes all available cores. Librapl [24] and Intel Power Governor [15] are utilized to profile CPU power consumption and cap the package power consumption respectively. Linux-monitoring sensors (lm_sensors) are used to measure CPU temperature. We recognize that our work and findings are bounded by the assumptions and scope, therefore, we also point out the threats to validity when discussing our results.

TABLE II: Benchmark domain and problem size

Rodinia	Problem size	**NPB**	Problem size
LUD (LU Decomposition)	4M	pseudo applications	
LavaMD	4K	LU, SP & BT	B
CFD (CFD Solver)	97K	kernels	
		FT, MG, IS, EP & CG	B

IV. Power Capping Effects on Performance

The first step toward obtaining the optimal checkpointing interval under power-capping is to understand how power-capping affects: (a) the execution time of simulation (computation time), (b) the execution time of checkpointing, and (c) system reliability. In this section, we focus on first two goals, i.e., how power capping affects the performance/execution time of application computation phase and checkpointing phase.

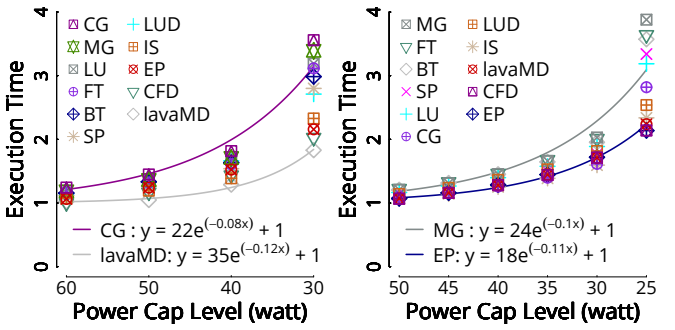
First, we present results that help us understand how different power capping affects the execution time of application computation phase. Fig. 1 shows the normalized execution time for a set of scientific benchmarks from linear algebra, computational fluid dynamics, and molecular dynamics domains (Table II), on two different platforms (Section III). We observe that the execution time increases non-linearly across all the benchmarks on both platforms. This indicates that power capping affects the computation time significantly, although the degree of effect may vary across benchmarks and platforms. We point out that the average power consumption for the benchmarks on Xeon E5-2670 platform ranges from 63 watts to 78 watts, and the minimum package power consumption that Intel power governor can enforce is approximately 23 watts on this platform. This implies that the range for reasonable power caps should between 23 watts and 63 watts to observe the effect on performance. Therefore, we choose power capping levels of 60, 50, 40, and 30 watts for Xeon E5-2670 platform. Similarly for Xeon E5-2630 platform, we choose power caps 50, 45, 40, 35, 30, and 25 watts taking average power consumption of the benchmarks into consideration.

To take this effect into consideration toward obtaining optimal checkpointing interval, we attempt to capture this trend mathematically. We find that normalized execution time under power capping for a given benchmark can be fitted using an exponential function. The R-squared values of regression functions are above 0.97 for all the benchmarks on both platforms indicating statistically sound fit. Since the benchmarks are affected differently by power capping in terms of execution time, the parameters or regression coefficients are different for each application. The exponential regression functions can be generalized as Equation 1.

$$\overline{T_{comp}}(P_i)/T_{comp} = A \times e^{B \times Pi} + 1 \quad (1)$$

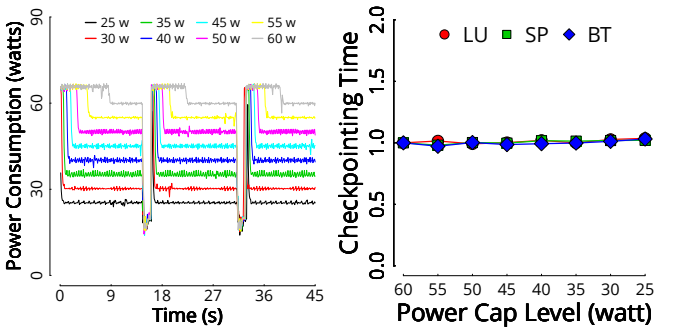
T_{comp} represents computation time without power capping, and $\overline{T_{comp}}(P_i)$ denotes the computation time under power cap P_i. e is Euler's number.

The upper bound and lower bound regression functions for both platforms are shown in Fig. 1. From these results we note that applications and platforms both have impact on the coefficients in fitted exponential functions. We study the impact of these co-efficients in later sections; in particular how these co-efficients affect the optimal checkpointing interval and total execution time under different power capping scenario.



(a) Xeon E5-2670 (b) Xeon E5-2630

Fig. 1: Effect of power capping on compute phase of benchmarks on different platforms.



(a) Power Consumption (b) Checkpointing Time

Fig. 2: Effect of power capping on checkpointing phase of benchmarks: BLCR checkpointing library is used and Checkpointing time is normalized by the checkpointing time without any power capping.

Next, we present results that help us understand how different power capping affects the execution time of checkpointing phase. Checkpointing is an I/O intensive operation in contrast to computation intensive scientific simulation applications. Therefore, one can reasonably expect to observe different power capping effects on checkpointing phase than on compute phase. We use BLCR to perform checkpointing on three benchmarks (LU, SP, and BT). We find similar results for other benchmarks and platforms, but due to space limitation we only present representative results that capture trends for all the benchmarks. Fig. 2 shows the computing and checkpointing power consumption and execution time on Xeon E5-2630 platform under different power caps. Notice the two dips in the Fig. 2 (a) which is corresponding to two checkpoint phases. Checkpointing power consumption under all power caps are similar (approximately 21.4 watts). The effect of power capping on checkpointing time can be captured by the Eq. 2.

$$\overline{\beta}(P_i) = \beta \quad (2)$$

β represents time needed to take a checkpoint without power capping, and $\overline{\beta}(P_i)$ denotes time needed to take a checkpoint

under power cap P_i. As expected, the effect of power capping on duration of checkpointing phase is minimal because checkpointing is an I/O intensive operation and throttling CPU performance to save instantaneous power does not affect I/O performance significantly, similar to observed by previous studies too.

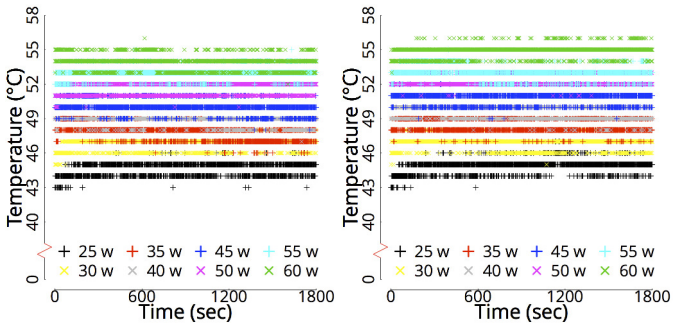
Finding 1: *Power capping affects the execution time of compute phase significantly across all benchmarks on different platform. Further, We show that this effect can be captured by an exponential function fitting.*

V. Power Capping Effects on MTBF

As discussed earlier, the next step toward obtaining the optimal checkpointing interval under power-capping is to understand how power-capping affects system reliability. However, deriving this relationship is more challenging. We show that this process consists of two steps. First, we show that power-capping directly affects the temperature of the system. Second, we show how temperature affects the system reliability.

A. Power Capping Effects on Temperature

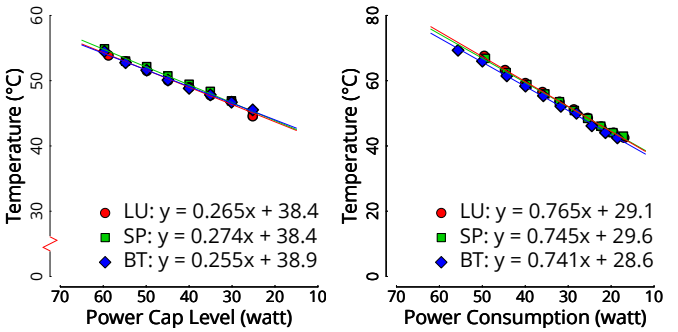
In this section, we show that different power-capping levels result in different steady-state temperature. First, we perform power capping and temperature measurement on the Xeon E5-2630 platform and E5-2670 platform. We run each benchmark for 1800 seconds under ten different power caps. Steady power consumption and temperature are calculated using the average of last 30 seconds. Fig. 3 shows the representative trend for two benchmarks on the Xeon E5-2630 platform. Both benchmarks show almost same behavior in terms of temperature profile under different power caps. We obtained similar results for other benchmarks and platforms, indicating that power-capping has a direct impact on the temperature and is largely independent of benchmark characteristics, unlike the effect of power-capping on execution time of compute phase.



(a) LU Temperature (b) SP Temperature

Fig. 3: Effect of power capping on the processor temperature (Xeon E5-2630 platform).

To take this effect into consideration toward obtaining optimal checkpointing interval, we attempt to capture this trend mathematically. We find that temperature under power capping can be fitted using a linear function. The R-squared values of regression functions are above 0.99 for different platforms indicating a statistically sound fit. We find the regression coefficients to be different across platforms, but



(a) Xeon E5-2630 (b) Core i7-2600

Fig. 4: Temperature variation with power capping level, and fitted regression functions on Xeon E5-2630 platform (a), and Core i7-2600 platform (b).

same for different benchmarks on the same platform. Fig. 4 (a) shows this for Xeon E5-2630 platform, we find similar results for other benchmark and platform combinations (e.g., Xeon E52670 and E52603). The linear regression functions can be expressed more generically as Eq. 3.

$$TEMP(P_i) = C \times Pi + D \tag{3}$$

$TEMP(P_i)$ denotes temperature under power cap P_i.

We recognize that the relationship between power cap and processor temperature can also depend on the cooling infrastructure. Heat is dissipated in the form of both heat exchange and radiation, which grows faster as processor temperature increases. When processor heat generation equals to heat dissipation, processor temperature becomes steady. A powerful cooling infrastructure can maintain a low steady processor temperature such as in servers. Therefore, in addition to Xeon server platforms, we also a chose a desktop-like processor (Core i7-2600) to be able to compare and contrast our findings with two different types of cooling infrastructures (Fig. 4(a) and (b)). We note that repeating the exactly same experiments for core i7-2600 is not feasible because power capping is not supported on desktop processors. Therefore, different clock rates are utilized to generate distinct power consumption and temperature pairs. Steady power consumption and temperature data of Core i7-2600 are shown in Fig. 4(b). These data points can still be fitted with linear functions as shown in the figure. The R-squared values of regression functions are higher than 0.99 for all the applications. As PC cooling infrastructure has less capacity as compared to server cooling infrastructure, Core i7-2600 temperature increases much faster than Xeon E5-2630 when power consumption is increased. This result also illustrates that the regression co-efficients are dependent on the platform and cooling infrastructure.

Finding 2: *Power capping level directly impacts the temperature of the processor. This relationship can be captured by a linear function, and is largely independent of the application characteristics for the different platform and benchmarks pairs we tested. However, the regression co-efficients are platform-specific, potentially indicating dependent on the cooling infrastructure itself.*

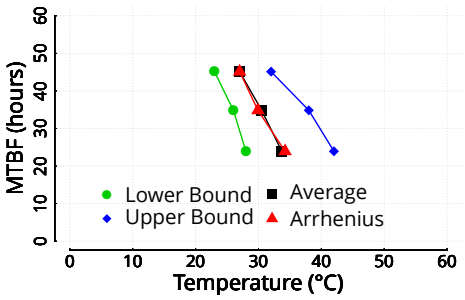


Fig. 5: Temperature and MTBF data of top, middle, and bottom cages on Titan. MTBF is calculated according to the total number of failures across cabinets, while temperature is calculated based on each individual cabinet.

B. Temperature Effects on MTBF

As discussed in the previous section, we relate power capping and system reliability by understanding how power capping affects temperature and then how temperature impacts the MTBF of the system. Previous works have shown the evidence that temperature can affect the overall system reliability [25]–[29].

In this section, we establish how temperature affects the MTBF of the system. For this purpose, we take advantage of Arrhenius Equation [30], which has been shown to fit computing systems [31] defining mean time between failure (MTBF) dependence on temperature.

$$\overline{MTBF}(P_i) = MTBF_{base}/F_A(TEMP(P_i)) \quad (4)$$

$F_A(x)$ is the acceleration factor under a given temperature x, as defined in Equation 5.

$$F_A(x) = e^{\frac{E_a}{k} \times (1/TEMP_{base} - 1/x)} \quad (5)$$

k is Boltzmann constant which equals to 8.617×10^{-5} eV/°K. E_a represents activation energy. Using the Titan supercomputer's data, we demonstrate that this relationship holds true in a large-scale HPC computing facility. Fig. 5 shows the temperature and MTBF data for different levels on cages in the cabinet for the Titan supercomputer. Each server cabinet in Titan consists of three cages, top, middle, and bottom. Cold air flows from bottom cage to top cage, which creates a gradient in ambient temperature. The temperature increases as we go from bottom cage to the top cage and hence, lower cages tend to have shorter MTBF. We set $MTBF_{base}$ and $TEMP_{base}$ as MTBF and average temperature of bottom cage. Then, we calculate temperature for middle and top cages based on MTBF of corresponding cage level using Equation 4. As shown in Fig. 5, temperature data for middle and top cages closely match the field data for the empirical value of activation energy, $E_a = 0.7eV$. We also plot the variance in the temperature data to show that it falls within the range and has similar trend.

This mathematical relationship can be used to model the system's reliability behavior and its impact on the optimal checkpointing interval. However, it is important to note the potential limitation and scope of this approach. We recognize that power-capping alone may not be responsible for temperature of different computing components. The inefficiencies in power/cooling infrastructure may cause temperature variability, in addition to what may be caused by the power-capping alone. Our approach doesn't directly and explicitly model such variance caused by the power/cooling infrastructure itself. Focus of this paper is to understand the impact of power-capping on checkpointing decisions, although other environmental conditions may also contribute toward such decision. We also note that power/cooling infrastructure can not completely mitigate the temperature's impact on system MTBF without dynamically changing the cooling infrastructure load. However, current HPC facilities often do not react dynamically to load-changes in order to adjust cooling resources. They are typically designed for a fixed load and therefore, power capping effect on the temperature will exist in such systems. Therefore, it is important to explicitly model and understand the power-capping's effect on checkpointing decisions, performance and energy consumption. Finally, we also note that we do not model the effect of variance in temperature on failures [29] since the presence of such effects in the Titan supercomputer's failure and temperature logs was not statistically significant.

Finding 3: *The system MTBF decreases with increase in temperature. The effect of temperature on the system MTBF can be modeled by Arrhenius Equation. We also show that the field data obtained on Titan validates this relationship.*

VI. Power Capping Effects on the OCI

TABLE III: Symbols and Definitions

Symbols	Definitions
P_i	power cap
α	checkpointing interval
β, $\overline{\beta}(P_i)$	time to take a checkpoint
γ	time to restart from a failure
ϵ	fraction of lost work
$TEMP_{base}$	baseline temperature
$MTTF_{base}$	baseline MTTF under $TEMP_{base}$
$TEMP(P_i)$	temperature under power cap P_i
$F_A(x)$	acceleration factor under temperature x
T_{total}, $\overline{T_{total}}(P_i)$	total execution time
T_{comp}, $\overline{T_{comp}}(P_i)$	total computation time
T_{chkp}, $\overline{T_{chkp}}(P_i)$	total time in taking checkpoints
T_{waste}, $\overline{T_{waste}}(P_i)$	total wasted time
P_{comp}	computing power consumption
P_{chkp}	checkpointing power consumption
E_{total}, $\overline{E_{total}}(P_i)$	total energy consumption
E_{comp}, $\overline{E_{comp}}(P_i)$	total computation energy
E_{chkp}, $\overline{E_{chkp}}(P_i)$	total energy in taking checkpoints
E_{waste}, $\overline{E_{waste}}(P_i)$	total wasted energy
T^{comp}_{waste}, $\overline{T^{comp}_{waste}}(P_i)$	total wasted computation time
T^{chkp}_{waste}, $\overline{T^{chkp}_{waste}}(P_i)$	total wasted checkpoint time
$T_{restart}$, $\overline{T_{restart}}(P_i)$	total time in restarting

* Symbols with overlines have the same meanings as the ones without overlines, except that they are under power cap P_i.

In section VI-A, first, we revisit how the first order model calculates the OCI [10], as shown in Equations 6 to 11. Then we introduce our power capping aware OCI model based on the first order model in section VI-B. In section VI-C, we revisit the high order model [12] and develop our high order power-aware OCI model using the same approach in first order model. Table III lists all the parameters used in the models.

A. First Order Model

According to the first order model, when considering checkpoint and restart, the total execution time is composed of successful computation time, successful checkpoint time and wasted time caused by failures, as shown in Eq. 6.

$$T_{total} = T_{comp} + T_{chkp} + T_{waste} \tag{6}$$

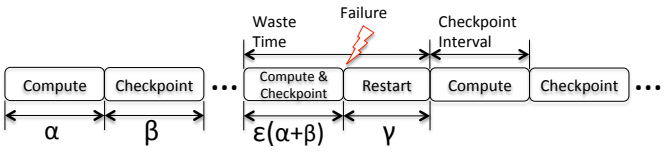


Fig. 6: An example of computation, checkpoint, failure, and restart. Computation is divided into multiple segments and a checkpoint is taken at the end of each segment.

An example to illustrate computing, checkpoint, failure, and restart is given in Fig. 6. Checkpointing interval is denoted as α. Time to take a single checkpoint is denoted as β. ϵ represents the fraction of lost computation and checkpoint. γ is the time to restart from a failure. Total amount of time spent in checkpointing can be expressed as Eq. 7.

$$T_{chkp} = (\frac{T_{comp}}{\alpha} - 1) \times \beta \tag{7}$$

Time wasted due to failures consists of lost computation (T^{comp}_{waste}), lost checkpoint (T^{chkp}_{waste}), and time spent in restart process ($T_{restart}$), which can be expressed as Eq. 8.

$$\begin{aligned} T_{waste} &= T^{comp}_{waste} + T^{chkp}_{waste} + T_{restart} \\ T^{comp}_{waste} &= \frac{T_{comp}}{\alpha} \times (e^{\frac{\alpha+\beta}{MTBF}} - 1)(\epsilon \times \alpha) \\ T^{chkp}_{waste} &= \frac{T_{comp}}{\alpha} \times (e^{\frac{\alpha+\beta}{MTBF}} - 1)(\epsilon \times \beta) \\ T_{restart} &= \frac{T_{comp}}{\alpha} \times (e^{\frac{\alpha+\beta}{MTBF}} - 1)(\gamma) \end{aligned} \tag{8}$$

Based on Eq. 6, total energy consumption of the first order model is shown in Eq. 9.

$$\begin{aligned} E_{total} =& P_{comp} \times T_{comp} + P_{chkp} \times T_{chkp} \\ &+ P_{comp} \times T^{comp}_{waste} + P_{chkp} \times (T^{chkp}_{waste} + T_{restart}) \end{aligned} \tag{9}$$

Replacing T_{chkp} and T_{waste} in Eq. 6 and Eq. 9, we can get the expressions of total time (T_{total}) and energy consumption (E_{total}) in the first order model.

The OCI optimized for execution time (α_t) is achieved when $\frac{d}{d\alpha}(T_{total}) = 0$. Similarly, the OCI optimized for energy consumption (α_e) is achieved when $\frac{d}{d\alpha}(E_{total}) = 0$. When $\alpha+\beta \ll MTBF$, expression $e^{\frac{\alpha+\beta}{MTBF}} - 1$ can be approximated as $\frac{\alpha+\beta}{MTBF}$. Solving the differential equations, we can get the expressions for the OCI optimized for execution time (α^-_t) and energy consumption (α^-_e) for the first order model, which are shown in Eq. 10 and Eq. 11 respectively. Note the superscript "$-$" signifies that OCI is from a power-unaware model.

$$\alpha^-_t = \sqrt{\beta^2 + \frac{\beta \times \gamma}{\epsilon} + \frac{MTBF \times \beta}{\epsilon}} \tag{10}$$

$$\alpha^-_e = \sqrt{\frac{P_{chkp}}{P_{comp}} \times (\beta^2 + \frac{\beta \times \gamma}{\epsilon} + \frac{MTBF \times \beta}{\epsilon})} \tag{11}$$

Finding 4: *Even without applying power-capping, the OCI optimized for energy can be smaller than the OCI optimized for performance. The difference between these OCIs gets larger as the ratio of power consumption during checkpointing to power consumption during computing becomes smaller.*

B. First Order Power-aware Model

Taking the first order model described here as a baseline, we propose a power-aware OCI model. The goal is to express T_{total} and E_{total} as functions of power cap P_i. Similar to Equation 6 and 9, we can write the following equations for execution time and energy consumption under a given power cap P_i (i.e., $\overline{T_{total}}(P_i)$, $\overline{E_{total}}(P_i)$).

$$\overline{T_{total}}(P_i) = \overline{T_{comp}}(P_i) + \overline{T_{chkp}}(P_i) + \overline{T_{waste}}(P_i) \tag{12}$$

$$\begin{aligned} \overline{E_{total}}(P_i) =& P_i \times \overline{T_{comp}}(P_i) + P_{chkp} \times \overline{T_{chkp}}(P_i) \\ &+ P_i \times \overline{T^{comp}_{waste}}(P_i) \\ &+ P_{chkp} \times (\overline{T^{chkp}_{waste}}(P_i) + \overline{T_{restart}}(P_i)) \end{aligned} \tag{13}$$

Equations 7 and 8 are also applicable here if we replace T_{comp} by $\overline{T_{comp}}(P_i)$ and $MTBF$ by $\overline{MTBF}(P_i)$. According to Eq. 1, T_{comp} can be expressed as functions of P_i (i.e., $\overline{T_{comp}}(P_i)$). Also, based on Eq. 3, Eq. 4 and Eq. 5, MTBF can also be expressed as a function of P_i (i.e., $\overline{MTBF}(P_i)$). Note that β is dominated by writing checkpoints to storage system, and γ is dominated by reading checkpoints from storage system. Since we find in our experiments that time to write checkpoints does not vary significantly with power capping, as shown in Eq. 2, it is reasonable to assume that time to read checkpoints is also independent of power capping. Therefore, we assume γ to be independent of power capping. After performing these substitutions we can obtain detailed expressions for $\overline{T_{total}}(P_i)$ and $\overline{E_{total}}(P_i)$. We do not show them here due to space limit.

The OCI optimized for execution time (α^+_t) and optimized for energy consumption (α^+_e) are achieved when $\frac{d}{d\alpha}(\overline{T_{total}}(P_i)) = 0$ and $\frac{d}{d\alpha}(\overline{E_{total}}(P_i)) = 0$ respectively. Note that superscript "+" signifies that the OCI includes the power capping aware model. When $\alpha + \beta \ll MTBF$, solving the differential equations, we can get the functional relationship between power and OCI, as shown in Eq. 14 and 15.

$$\alpha^+_t = \sqrt{\beta^2 + \frac{2 \times \beta \times \gamma}{\epsilon} + \frac{MTTF_{base} \times \beta}{F_A(C \times P_i + D) \times \epsilon}} \tag{14}$$

$$\alpha^+_e = \sqrt{\frac{P_{chkp}}{P_i} \times [\beta^2 + \frac{2 \times \beta \times \gamma}{\epsilon} + \frac{MTTF_{base} \times \beta}{F_A(C \times P_i + D) \times \epsilon}]} \tag{15}$$

We note that Power capping aware OCI has no dependence on the regression co-efficients A and B, which show how power capping affects compute phase performance.

Finding 5: *Power capping aware OCI is not determined by how the compute phase performance is affected by power capping. That is, two applications with varying sensitivity toward power capping on their compute phase performance, will have the same OCI if all else is the same.*

C. High Order Models

The high order model refines T_{waste} in the first order model in the following three steps. First, high order model introduces the fraction of lost work over a time interval $\phi(\Delta t)$ to replace ϵ, which is shown in Eq. 16, similar to [12]. Second, high order model defines the number of failures as $\frac{T_{total}}{MTBF}$ to consider multiple failures in a computing segment. Finally, high order model considers failures during restart processes. Refined T_{waste}^{comp}, T_{waste}^{chkp}, and $T_{restart}$ are shown in Eq. 17 and Eq. 18.

$$\phi(\Delta t) = \frac{MTBF}{\Delta t} + \frac{1}{1 - e^{\Delta t / MTBF}} \tag{16}$$

$$\begin{aligned} T_{waste}^{comp} =& \frac{T_{total}}{MTBF} \times [\phi(\alpha+\beta) \times \alpha \times e^{-\frac{\alpha+\beta+\gamma}{MTBF}} \\ &+ \phi(\alpha+\beta+\gamma) \times \alpha \times (1 - e^{-\frac{\alpha+\beta+\gamma}{MTBF}})] \\ T_{waste}^{chkp} =& \frac{T_{total}}{MTBF} \times [\phi(\alpha+\beta) \times \beta \times e^{-\frac{\alpha+\beta+\gamma}{MTBF}} \\ &+ \phi(\alpha+\beta+\gamma) \times \beta \times (1 - e^{-\frac{\alpha+\beta+\gamma}{MTBF}})] \end{aligned} \tag{17}$$

$$\begin{aligned} T_{restart} =& \frac{T_{total}}{MTBF} \times [\gamma \times e^{-\frac{\alpha+\beta+\gamma}{MTBF}} \\ &+ \phi(\alpha+\beta+\gamma) \times \gamma \times (1 - e^{-\frac{\alpha+\beta+\gamma}{MTBF}})] \end{aligned} \tag{18}$$

Replacing T_{waste} in the first order model, the expression of T_{total} for the high order model is shown in Eq. 19.

$$\begin{aligned} T_{total} =& MTBF \times (\frac{T_{comp}}{\alpha} - \frac{\beta}{\alpha+\beta}) \\ &\times e^{\frac{\gamma}{MTBF}} \times (e^{\frac{\alpha+\beta}{MTBF}} - 1) \end{aligned} \tag{19}$$

High order model can also be extended to power-aware OCI model using same approach in Section VI-B. Since the high order model only refines T_{waste}, we can derive $\overline{T_{total}}(P_i)$ and $\overline{E_{total}}(P_i)$ from $\overline{T_{waste}}(P_i)$. Similar to Eq. 17, Eq. 18, and Eq. 19, $\overline{T_{waste}^{comp}}(P_i)$, $\overline{T_{waste}^{chkp}}(P_i)$, and $\overline{T_{restart}}(P_i)$ can be expressed if T_{total}, $MTBF$, and ϕ is replaced by $\overline{T_{total}}(P_i)$, $\overline{MTBF}(P_i)$ and $\overline{\phi}(\Delta t)$. Similar to Section VI-B, $\overline{MTBF}(P_i)$ can be obtained using Eq. 3, 4 and 5. Expression for $\overline{T_{total}}(P_i)$ is shown in Eq. 20 and Eq. $\overline{\phi}(\Delta t)$ is defined in Eq. 21.

$$\begin{aligned} \overline{T_{total}}(P_i) =& \overline{MTBF}(P_i) \times (\frac{\overline{T_{comp}}(P_i)}{\alpha} - \frac{\beta}{\alpha+\beta}) \\ &\times e^{\frac{\gamma}{\overline{MTBF}(P_i)}} \times (e^{\frac{\alpha+\beta}{\overline{MTBF}(P_i)}} - 1) \end{aligned} \tag{20}$$

$$\overline{\phi}(\Delta t) = \frac{\overline{MTBF}(P_i)}{\Delta t} + \frac{1}{1 - e^{\Delta t / \overline{MTBF}(P_i)}} \tag{21}$$

Based on Eq. 13, we can also get the expression of $\overline{E_{total}}(P_i)$ for power capping aware high order model. The OCI optimized for execution time (α_t^+) and optimized for energy consumption (α_e^+) are achieved when $\frac{d}{d\alpha}(\overline{T_{total}}(P_i)) = 0$ and $\frac{d}{d\alpha}(\overline{E_{total}}(P_i)) = 0$ respectively. The analytical solution of the OCI can be found in [11] [12]. However, the analytical solutions are approximation to or estimation of OCIs based on certain conditions. In order to accurately predict OCIs under all conditions, we use the numeric solver "vpasolve" in MATLAB to calculate OCIs for both first order power-aware model and high order power-aware model.

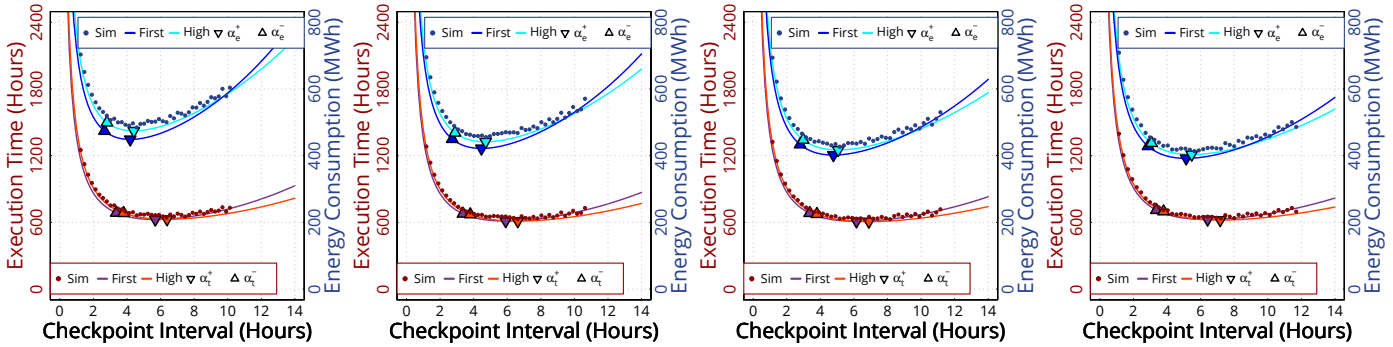
VII. Model Validation and Model-driven Study

In this section, we perform simulations using an event-driven simulator and validate our model against simulation results. Then, we conduct model-driven study to compare our power-aware OCI models with the prior power capping unaware OCI models (i.e., first order and high order models).

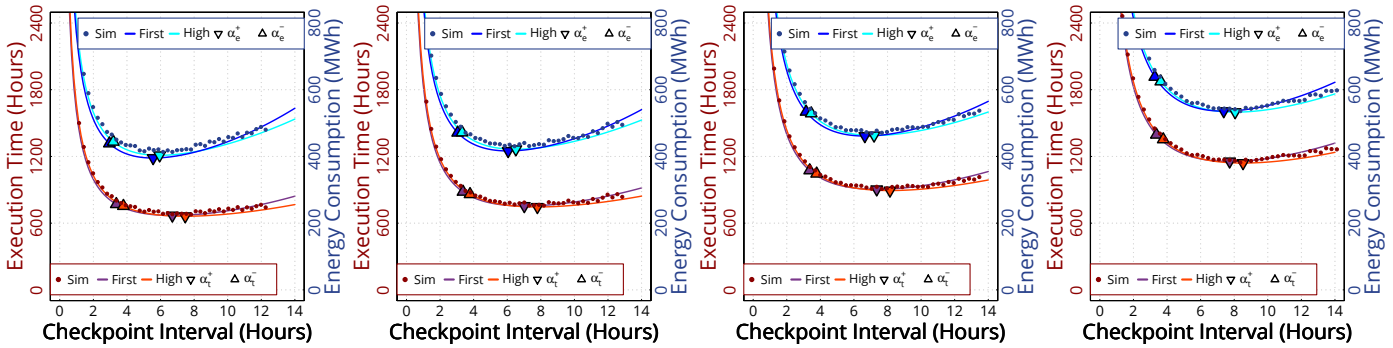
To validate our power capping aware OCI model, we use a simulation based approach to compare against. We developed an event-driven simulator to simulate the compute phase, checkpointing phase, and failure events. The simulator generates random failures which follow a Poisson process, and intervals between failures follow an exponential distribution. The execution time, checkpointing time, and MTBF is adjusted in the simulation based on the input power capping level in accordance to relationships derived in previous sections.

Threats to validity: We recognize that our findings are bounded by the assumptions and parameter settings. To mimic real-world scenario, our simulation based evaluation is driven by parameters obtained from real-system experiments, large-scale application characteristics, empirical parameters obtained from HPC facility. The simulation based study uses power capping related coefficients that are experimentally obtained from the different Intel Xeon platforms for a variety of scientific applications. Simulation based study is driven by failure and I/O data obtained from the Titan supercomputer and the temperature dependence of MTBF has also been simulated based on the Titan supercomputer data. At the same time, we also acknowledge that it is not always possible to obtain real-world data to drive simulation based studies. In such cases, we have used a range of parameters to simulate the impact of such factors. We simulated failure events using Weibull distribution to mimic real-world scenario and obtained similar accuracy and results, but results are omitted due to space restrictions.

Fig. 7 shows the total execution time and energy consumption under different power-caps for a peta-scale system, similar to the Tian supercomputer. The figure curves correspond to our power capping aware models and the simulation. We also mark the OCIs obtained by previous models and OCIs obtained by new power capping aware model. The simulation setup assumes a 120 hour long application running on a Titan-like supercomputer, which is composed of 20,000 nodes. It assumes to have the same MTBF as Titan under the same temperature. The power capping effect is modeled after Xeon E5-2630 processor. The regression co-efficients corresponds to that platform and pseudo applications from the NAS benchmark suite (LU, SP, BT) on that platform. Note that these applications have similar regression co-efficients. We later perform sensitivity analysis w.r.t. such co-efficients as well. The checkpointing power is taken as 21.4 watts as on measured on this platform. The baseline power consumption is 64.1 watts under no power capping. Baseline temperature is calculated based on Eq. 3, and baseline MTBF is calculated based on MTBF and temperature data from Titan logs using Arrhenius Equation with the empirical value of activation energy. The checkpointing time is taken as to be 3.6% of the compute time as obtained from our experiments from the BLCR checkpointing library. We assume that time to restart equals to time to checkpoint, since the former one primarily reads checkpoints from storage system and the latter one



(a) Power Cap 60 watts (b) Power Cap 55 watts (c) Power Cap 50 watts (d) Power Cap 45 watts



(e) Power Cap 40 watts (f) Power Cap 35 watts (g) Power Cap 30 watts (h) Power Cap 25 watts

Fig. 7: Execution time and energy consumption under various checkpointing intervals. Legends at the bottom represent execution time (T_{total}), and legends at the top represent energy consumption. "Sim" is simulation results, "First" represents first order model, and "High" denotes high order model. OCIs derived from prior models (i.e., α_t^- and α_e^-) are marked with a triangle facing upwards and OCIs calculated from our model (i.e., α_t^+ and α_e^+) are marked with a triangle facing the downwards.

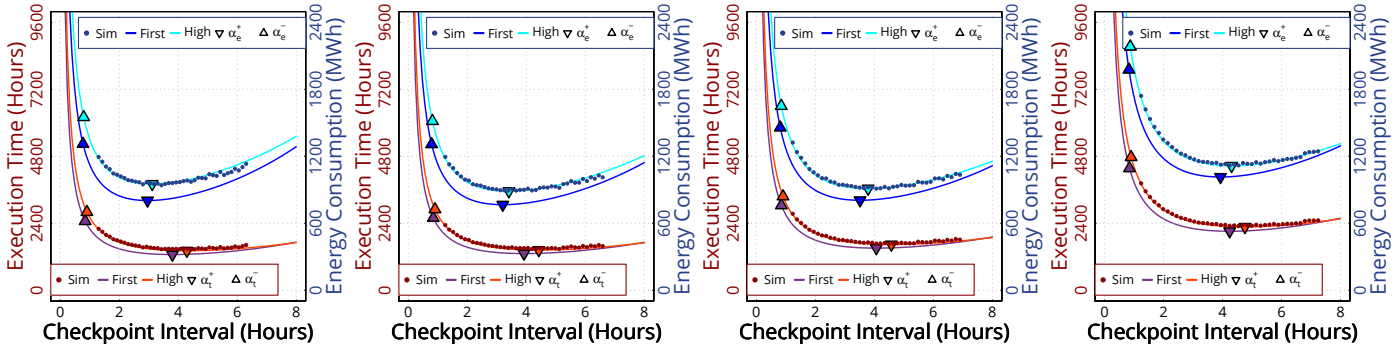
mainly writes checkpoints to storage system. We simulate more than 40 checkpointing intervals under each power cap.

We make several observations from the Fig. 7. First, we find that across various power caps, the power capping aware model predicted OCI closely matches with the simulation results corresponding to minimum execution time and energy consumption. Second, the OCI predicted by previous models, which do not take power capping effects into account, are significantly far from optimal OCI points. In most cases, this results in more than 10% performance loss and additional energy consumption. We also notice that the difference between the first and high order model is not significant in the cases presented here. Finally, we also observe that the power capping aware model results in significant savings as the power cap becomes smaller. For example, the performance difference between power capping aware OCI model and high order model increases from 8.8% to 17.2% when power cap drops from 60 watts to 25 watts. This is primarily because the new model captures the MTBF change due to power capping better as the power cap drops.

Finding 6: *Our model predicted OCI closely matches the minimal execution time and energy consumption achieved by the simulation runs. The power capping aware model results in significant performance and energy savings compared to previous models. Also, these savings increase significantly as the power cap gets tighter.*

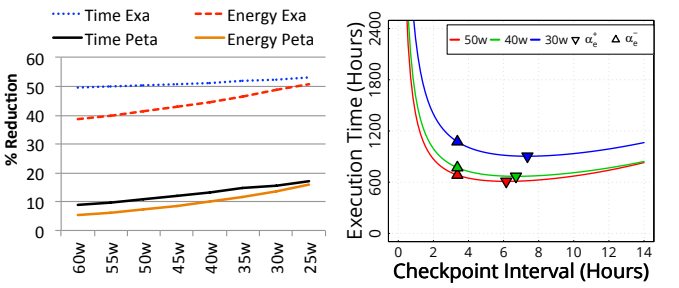
Next, we show that our model is validated for an exascale-like system as well. Fig. 8 shows that the model closely follows the simulation and power capping aware model predicts OCI accurately. Interestingly, the improvements in performance and energy due to new model is higher compared to the petascale system. This is because at exascale the MTBF becomes smaller and hence, previous models take checkpoint more frequently and incur very high I/O overhead. However, the power capping aware model adjusts the OCI taking both the system scale and power capping into account. It estimates the OCI to be a bit higher and hence, results in significantly less I/O overhead. As a key summary, Fig. 9(a) shows that applying our power capping aware OCI model can reduce the total execution time and energy consumption, compared to prior OCI models at both peta- and exa-scale. We observe that for a petascale system execution time can be improved between 8.8% to 17.2% using the high order power capping aware OCI model. This effect is even more pronounced for exascale system where execution time can be improved between 49.4% to 52.9% using the high order power capping aware OCI model. Similar savings can be observed for energy consumption as well when applying power capping aware OCI model.

In Fig. 7 and 8, it can be noticed that the execution time and energy curves shift upward when the power cap is reduced. To illustrate and understand this trend better, we show Fig. 9(b) where the execution time curve is plotted for a power cap of 50 watt, 40 watt, and 30 watt. α_t^+ and α_t^- are also marked



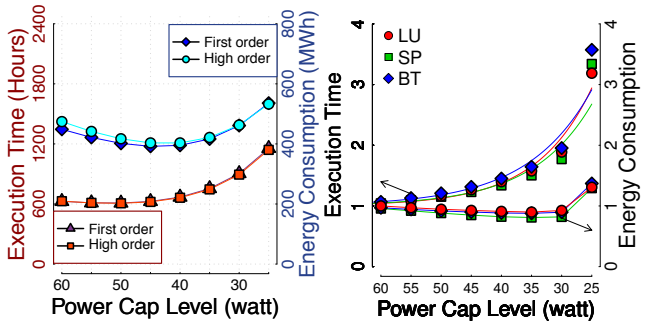
(a) Power Cap 60 watts (b) Power Cap 50 watts (c) Power Cap 40 watts (d) Power Cap 30 watts

Fig. 8: Execution time and energy consumption under various checkpointing intervals for exascale system. Legends have the same meanings as in Figure 7.



(a) (b)

Fig. 9: Improvement in performance and energy consumption at petascale and exascale under different power caps (a), and execution time curves under different power capping levels with OCIs for power aware model and prior model without power capping awareness (b).



(a) Optimal checkpointing interval (b) No checkpointing

Fig. 10: Total execution time and energy consumption under various power caps with the optimal checkpointing interval (a), and when checkpointing overhead is ignored (b).

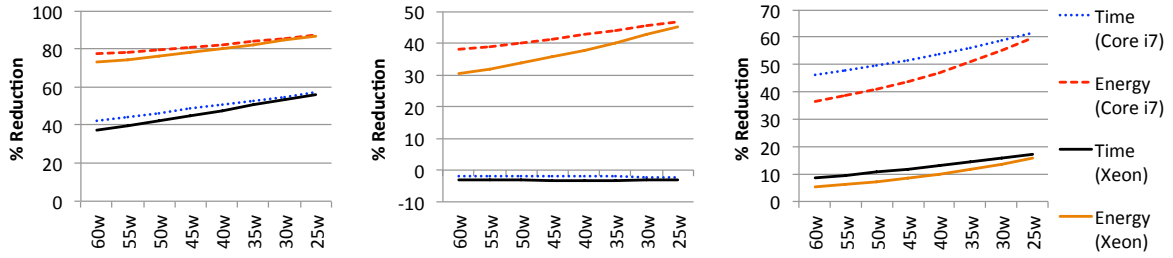
for each power cap in the figure. Notice that α_t^- are on same vertical line because power cap unaware OCI stays the same. On the other hand, as the power cap is reduced the curves shift upward due to increased execution time, and the curve shifts towards the right due to increasing MTBF. Therefore, α_t^+ increases as the power cap decreases due to change in MTBF. Prior models can not take this effect into account and lead to suboptimal OCI estimation. This explains why and how the power capping aware model outperforms the prior models in different situations.

Finding 7: *As the system scale increases, the benefit of power capping aware OCI model also increases significantly compared to the prior models.*

Next, we show that it is critical to choose the correct power cap level to achieve minimum execution time and energy consumption. Fig. 10(a) shows the best performance is achieved when power cap is 50 watts, and the lowest energy consumption is achieved when power cap is 45 watts. This illustrates that the optimal power capping level itself depends upon the metric of optimization (e.g., performance, and energy). We also point out that the corresponding OCI on these power capping levels would be different as well, this can be obtained via our model. Note that this result includes the failure events, checkpointing, and restart phase.

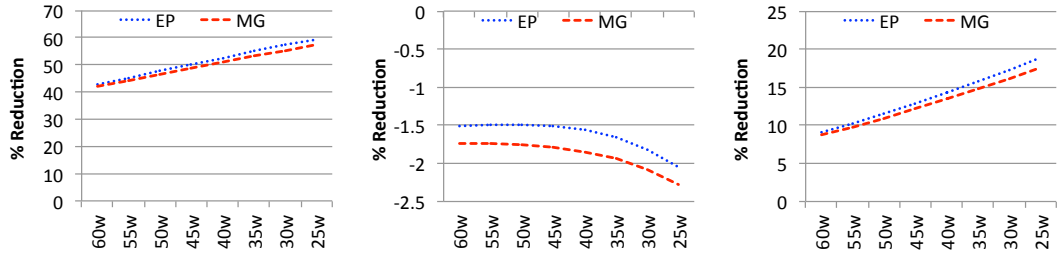
On the other hand, the results in Fig. 10(b) do not take failures, checkpoint, and restart into account. Interestingly, in this case the lowest execution time is achieved with power cap of 60 watt, and least energy consumption is observed for power cap of 35 watts when we do not consider failures, checkpoint, and restart. This is a critical finding that illustrates that optimal power cap levels can not alone be decided by how power capping affects the application in isolation, without taking failures, checkpointing, and restart phase into account. These shifts in optimal power caps are caused by the impact of power capping on MTBF. We find that using 45 watt power cap instead of 35 watt power cap leads to 20.2% savings in energy consumption. Note that this reduction in energy consumption is between power-unaware model based OCI at 35 watt and power-aware model based OCI at 45 watts. Similarly, using 50 watt power cap instead of 60 watt power cap leads to 12.6% reduction in execution time.

Finding 8: *The optimal power cap levels for minimizing execution time and energy consumption are different and so are their corresponding OCIs. The optimal power cap levels for minimizing execution time and energy consumption change once failures, checkpoint, and restart phases are taken into account. The corresponding difference in execution time and energy consumption is significant. Our results also show that power capping aware OCI model leads to significant improvements.*



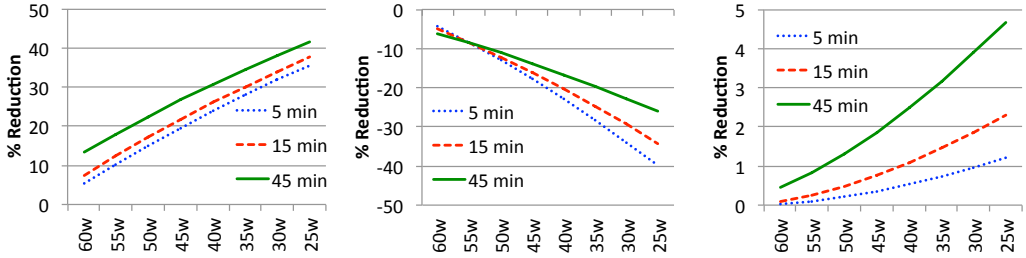
(a) Checkpoint time (b) Wasted time (c) Total execution time

Fig. 11: Percentage reduction in time and energy spent in checkpointing, wasted work, and the total time/energy under various power caps and for different cooling platform parameters.



(a) Checkpoint time (b) Wasted time (c) Total execution time

Fig. 12: Sensitivity study on application-specific coefficients (parameter A and B). Reduction in time spent in checkpointing, wasted work, and total execution under different power caps.



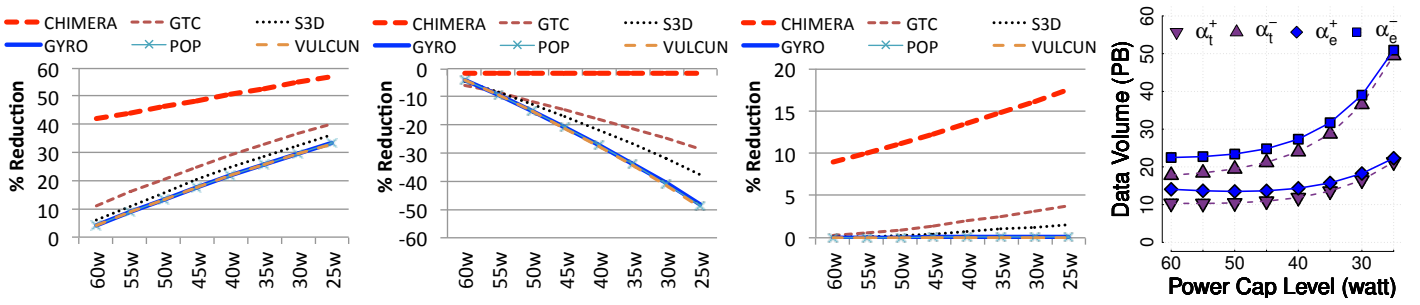
(a) Checkpoint time (b) Wasted time (c) Total execution time

Fig. 13: Sensitivity study on time to checkpoint (β). Reduction in time spent in checkpointing, wasted work, and total execution under different power caps.

Next, we study how improvement obtained by the power capping aware OCI model changes when the application specific parameters (A and B), and platform specific parameters (C and D) change. We also study the impact of time to checkpoint (β) on the improvements. It is followed by evaluation for real scientific applications. We compare our power capping aware OCI model with prior OCI models, in terms of execution time, energy consumption, and overall checkpointing data volume.

Percentage reduction in execution time and energy consumption for checkpointing, wasted, and total between prior OCI models and our OCI model under various power caps are shown in Fig. 11. First, we focus on results for Xeon platform ($C = 0.26$, $D = 38.6$). Comparing with prior models, our model can achieve 9% to 17% reduction in total execution time, and 42% to 57% reduction in checkpointing time. There is a minor increase in waste work but it is offset by significant savings in checkpointing time. Percentage changes for total and checkpointing time increase as power cap decreases. Energy consumption also follows similar trend. The reason is that the difference between OCIs derived from prior OCI models and our power capping aware OCI model is increasing as power cap decreases, as shown in Fig. 9. We chose Core i7 platform because it has higher sensitivity to temperature with respect to power consumption (higher value of C parameter, $C = 0.75$, $D = 29.1$). We find that platforms with higher temperature gradient w.r.t. power capping benefit significantly more by applying power capping aware OCI model (Fig. 11), and their corresponding OCI is also significantly different that can be obtained from our model. Overall, the power capping aware OCI also results in reduction in the checkpoint time.

We point out that reducing the number of checkpoints can relieve the burden on the storage system of an HPC system which is a shared and constraint resource. Therefore, power



(a) Checkpoint time (b) Wasted time (c) Total execution time (d) CHIMERA Checkpoint Volume

Fig. 14: Percentage reduction in time spent in checkpointing, wasted work, and total execution under different power caps for several leadership applications.

capping aware OCI may in return improve the overall I/O performance of the whole system and other applications.

Next, we study the sensitivity toward application specific parameters (A and B) that represent the impact of power capping on application performance (Section IV). We perform experiments with the applications that have regression functions in Fig. 1(b) at both extremes (i.e., MG and EP). Reduction in time spent on checkpointing, wasted, and total execution is shown in Fig. 12. From the figure we can observe that the application specific parameters do not have a significant impact on the improvements of power capping aware OCI. This is expected because parameter A and B do not directly impact the OCI estimation α_t^+ and α_e^+ (as noted earlier in the modeling section). Also, we note that the OCI is same for MG and EP. Note that slight difference in the percentage reduction is because the execution time still depends on the parameter A and B.

Finding 9: *When comparing our power capping aware OCI model with prior OCI models, percentage changes on checkpointing, wasted, and total execution time are not highly sensitive to the application-specific coefficients. Also, we show that the platforms with higher temperature gradient w.r.t. power capping benefit significantly more by applying power capping aware OCI model.*

Next, we also perform a sensitivity study on the time-to-checkpoint. We present experimental results for β equals to 5, 15, and 45 minutes. Fig. 13 shows that reduction in checkpointing, wasted, and total execution time increases when time to checkpoint increases. This reduction is even more pronounced for lower power caps and shows significant reduction in checkpoint time. This indicates that applying our model can reduce the checkpointing, and total execution time more significantly when time to checkpoint is larger.

Finding 10: *Power capping aware OCI model has increasing gains over prior OCI models as the time to checkpoint increases. With increasing system/problem scales and relatively slow growing I/O bandwidth, our model can obtain increasing benefits in I/O bandwidth constrained systems.*

Finally, we perform the evaluations based on the checkpoint data size and execution time of leadership applications run on OLCF machines [1], [2], as shown in Table I. Checkpointing time is the quotient of checkpoint data size divided by average PFS bandwidth. These leadership applications utilize application level checkpointing instead of system level checkpointing. Their checkpointing time does not necessarily scale up with problem size, and is user-specific.

We keep the assumptions in this section except the checkpointing time obtained from BLCR. We use average 10GB/s bandwidth as obtained from Spider parallel filesystem (PFS) attached to the the Titan supercomputer [32] to calculate checkpointing time. Percentage changes for checkpointing, wasted, and total execution time between prior OCI models and our power-aware OCI model under different power caps are shown in Fig. 14. We do not show results for reduction in energy consumption due to space constraint, but energy consumption follows similar trends as execution time, as observed in Fig. 11.

As shown in Fig. 13, the checkpoint time has significant impact on the savings achieved by power capping aware OCI. Similarly, we see that applying our power-aware OCI model to CHIMERA reduces the total execution time by 9% to 18% compared to prior OCI models because it has large checkpoint data size. Applications such as GTC and S3D have moderate checkpoint data sizes. Total execution time decreases by 4% and 2% respectively, when applying our power-aware model to GTC and S3D. For applications with small checkpoint data sizes, i.e., GYRO, POP, and VULCUN/2D, our power-aware model has about the same total execution time as prior models.

Finding 11: *Using the power capping aware OCI model, applications with large checkpoint data size can achieve substantial reduction on checkpointing time and total execution time over prior OCI models.*

Although our OCI model has limited benefits in terms of total execution time when application checkpoint data size is small, it can still significantly reduce the total volume of checkpoint data being written to storage systems. For GTC and S3D, our model can reduce the checkpoint data volume by up to 40% and 36% respectively. Even for applications such as GYRO, POP, and VULCUN, our model can still reduce the checkpoint data volume by up to 33%. This means that less checkpoints are written to the storage system, which helps resolving the PFS bottleneck problem, and improving application and checkpointing I/O performance. We show checkpointing data volume of CHIMERA for all four OCI schemes in Fig. 14(d). The power capping aware OCI reduces the checkpoint volume by 57% for 25 watts power cap level, compared to the prior models. We also notice that the gap

between our power capping aware OCI model and the first order model increases when the power cap decreases.

VIII. Conclusion and Future Work

In this paper, we investigate the effects of power capping on the optimal checkpointing interval. We study the effect of power capping on compute and checkpointing phase for a variety of scientific applications. We also demonstrate and quantify how power capping affects the system reliability due to change in temperature. We propose an power-aware OCI model and validation shows that our model can accurately predict the OCI under power capping. Our evaluation shows that applying our model to a set of large-scale applications can save up to 18% energy and execution time. Moreover, our model reduces the volume of data movement by up to 57% for these large-scale applications. In the future, we plan to extend our model to support heterogeneous platforms and to consider the impacts of manufacturing variations.

References

[1] D. Kothe and R. Kendall, "Computational Science Requirements for Leadership Computing," Oak Ridge National Laboratory, Tech. Rep., 2007.

[2] W. Joubert, D. Kothe, and H. A. Nam, "Preparing for Exascale: ORNL Leadership Computing Facility Application Requirements and Strategy," Oak Ridge National Laboratory, Tech. Rep., 2009.

[3] J. Duell, P. Hargrove, and E. Roman, "The Design and Implementation of Berkeley Lab's Linux Checkpoint/Restart," Berkeley Lab, Tech. Rep., 2002.

[4] A. Moody, G. Bronevetsky, K. Mohror, and B. R. d. Supinski, "Design, Modeling, and Evaluation of a Scalable Multi-level Checkpointing System," in *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, 2010, pp. 1–11.

[5] L. Bautista-Gomez, S. Tsuboi, D. Komatitsch, F. Cappello, N. Maruyama, and S. Matsuoka, "FTI: high performance fault tolerance interface for hybrid systems," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011, pp. 32:1–32:12.

[6] J. Bent, G. Grider, B. Kettering, A. Manzanares, M. McClelland, A. Torres, and A. Torrez, "Storage challenges at Los Alamos National Lab," in *IEEE MSST*, 2012, pp. 1–5.

[7] F. Cappello, "Fault Tolerance in Petascale/ Exascale Systems: Current Knowledge, Challenges and Research Opportunities," *International Journal of High Performance Computing Applications*, vol. 23, no. 3, pp. 212–226, 2009.

[8] D. Tiwari, S. Gupta, and S. S. Vazhkudai, "Lazy Checkpointing: Exploiting Temporal Locality in Failures to Mitigate Checkpointing Overheads on Extreme-Scale Systems," in *44th Annual IEEE/IFIP Int'l Conference on Dependable Systems and Networks*, 2014, pp. 25 – 36.

[9] L. Bautista-Gomez, A. Gainaru, S. Perarnau, D. Tiwari, S. Gupta, C. Engelmann, F. Cappello, and M. Snir, "Reducing waste in extreme scale systems through introspective analysis," 2016.

[10] J. W. Young, "A First Order Approximation to the Optimum Checkpoint Interval," *Communications of the ACM*, vol. 17, no. 9, pp. 530–531, 1974.

[11] J. Daly, "A Model for Predicting the Optimum Checkpoint Interval for Restart Dumps," in *Proceedings of the International Conference on Computational Science*, 2003, pp. 3–12.

[12] J. Daly, "A higher order estimate of the optimum checkpoint interval for restart dumps," *Future Generation Computer Systems*, vol. 22, no. 2006, pp. 303–312, 2004.

[13] C. Lefurgy, X. Wang, and M. Ware, "Power capping: a prelude to power shifting," *Cluster Computing*, vol. 11, no. 2, pp. 183–195, 2008.

[14] A. Gandhi, M. Harchol-Balter, R. Das, J. O. Kephart, and C. Lefurgy, "Power Capping Via Forced Idleness," in *Proceedings of Workshop on Energy-Efficient Design*, 2009.

[15] M. Dimitrov, C. Strickland, S.-W. Kim, K. Kumar, and K. Doshi, "Intel Power Governor," https://software.intel.com/en-us/articles/intel-power-governor, July 2012.

[16] Intel, *Intel 64 and IA-32 Architectures Software Developer's Manual*. Intel Corporation, 2015, vol. 3B, no. 2.

[17] K. Ma and X. Wang, "PGCapping: Exploiting Power Gating for Power Capping and Core Lifetime Balancing in CMPs," in *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, 2012, pp. 13–22.

[18] A. Hussein, A. L. Hosking, M. Payer, and C. A. Vick, "Don't race the memory bus: taming the gc leadfoot," in *Proceedings of the 2015 ACM SIGPLAN International Symposium on Memory Management*. ACM, 2015, pp. 15–27.

[19] S. Agarwal, R. Garg, M. S. Gupta, and J. E. Moreira, "Adaptive Incremental Checkpointing for Massively Parallel Systems," in *Proceedings of the 18th Annual International Conference on Supercomputing*, 2004, pp. 277–286.

[20] K. Ferreira, J. Stearley, J. H. Laros, III, R. Oldfield, K. Pedretti, R. Brightwell, R. Riesen, P. G. Bridges, and D. Arnold, "Evaluating the Viability of Process Replication Reliability for Exascale Systems," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, 2011, pp. 44:1–44:12.

[21] M. Forshaw, A. S. McGough, and N. Thomas, "Energy-efficient checkpointing in high-throughput cycle-stealing distributed systems," *Electronic Notes in Theoretical Computer Science*, vol. 310, pp. 65–90, 2015.

[22] S. S. Shende and A. D. Malony, "The Tau Parallel Performance System," *International Journal of High Performance Computing Applications*, vol. 20, no. 2, pp. 287–311, 2006.

[23] P. J. Mucci, S. Browne, C. Deane, and G. Ho, "PAPI: A Portable Interface to Hardware Performance Counters," in *Proceedings of Department of Defense HPCMP Users Group Conference*, 1999.

[24] T. Patki, D. K. Lowenthal, B. Rountree, M. Schulz, and B. R. de Supinski, "Exploring Hardware Overprovisioning in Power-constrained, High Performance Computing," in *Proceedings of the 27th International ACM Conference on International Conference on Supercomputing*, 2013, pp. 173–182.

[25] D. Tiwari, S. Gupta, J. Rogers, D. Maxwell, P. Rech, S. Vazhkudai, D. Oliveira, D. Londo, N. DeBardeleben, P. Navaux *et al.*, "Understanding gpu errors on large-scale hpc systems and the implications for system design and operation," in *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*. IEEE, 2015, pp. 331–342.

[26] S. Gupta, D. Tiwari, C. Jantzi, J. Rogers, and D. Maxwell, "Understanding and exploiting spatial properties of system failures on extreme-scale hpc systems," in *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*. IEEE, 2015, pp. 37–44.

[27] B. Nie, D. Tiwari, S. Gupta, E. Smirni, and J. H. Rogers, "A large-scale study of soft-errors on gpus in the field," in *High Performance Computer Architecture (HPCA), 2016 IEEE 22nd International Symposium on*, 2016.

[28] D. Tiwari, S. Gupta, G. Gallarno, J. Rogers, and D. Maxwell, "Reliability lessons learned from gpu experience with the titan supercomputer at oak ridge leadership computing facility," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2015, p. 38.

[29] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, "Temperature management in data centers: why some (might) like it hot," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, pp. 163–174, 2012.

[30] "Arrhenius equation," https://en.wikipedia.org/wiki/Arrhenius_equation.

[31] P. Ellerman, "Calculating Reliability using FIT and MTTF: Arrhenius HTOL Model," microsemi, Tech. Rep., 2012.

[32] G. Shipman and et al., "A next-generation parallel file system environment for the OLCF," in *Proceedings of the Cray User Group Conference*, 2012.