DUAL: Reliability-Aware Power Management in Data Centers

Xin Xu
George Washington University

Kayo Teramoto*
Yale University

Allan Morales
George Washington University

H. Howie Huang
George Washington University

***Abstract*—A virtualized data center hosts users and applications within a large number of virtual machines (VM) to achieve easy provisioning and high utilization of physical resources. Energy efficiency and reliability are two primary concerns for operating a data center. Power saving techniques, such as dynamic voltage and frequency scaling (DVFS), are often employed to reduce the supply voltages of the CPUs in runtime when the computer system utilization is low. However, DVFS can potentially decrease the system reliability - the processors at low voltages are more likely to encounter soft errors that may result in VM or system crashes. In this work, we propose a data center management framework, DUAL, which consists of the new virtual machine power and reliability analysis tools. The framework is designed to balance the *dual* needs of a data center: reducing energy consumption and providing high reliability. The evaluations show that DUAL can help maintain the desired reliability and significantly reduce power consumption, which in turn will lower the overall operational cost of a data center.**

I. Introduction

A data center can have a large number of servers each hosting dozens of virtual machines (VM) and various applications, such as database, video streaming, and web servers. The goal of data center operators is to design effective management policies to ensure the quality of services at a competitive cost. Among many parameters of interest, power efficiency and reliability are two major factors that affect the overall cost of a data center. It is known that power consumption makes up a large percentage of the operational budget of a data center [1]. Power saving techniques such as dynamic voltage and frequency scaling, or DVFS (e.g., Intel SpeedStep) can dynamically adjust the CPU voltage when the utilization changes. As the workload in a data center fluctuates over time, a common management approach is to use the DVFS technique to optimize power consumption with application performance used as a constraint.

Unfortunately, the CPU voltage, which is controlled by DVFS, has a significant impact on the chip reliability, because the CMOS chips under low voltages become more prone to the soft errors [2], [3], [4]. The soft errors in CPUs could potentially result in silent data corruptions (i.e., data are corrupted but users do not notice), as well as system and software crashes. Impacts of soft errors have been well documented in [5], [6], and in one incidence [7], the soft errors that occurred on L2 cache led to the server crashes and affected several internet companies. Our evaluation also shows that a virtualized data center could experience seven times more failures when the DVFS is enabled in processors. More importantly, as the scale of a data center increases, the probability of failures and the cost for each visible failure rise significantly. A single downtime event with a 90-minute outage can cost half a million dollars on average, including recovery costs, loss of end-user productivity, and business disruption [8].

*Kayo Teramoto did this work as a Summer 2012 undergraduate research assistant at GWU.

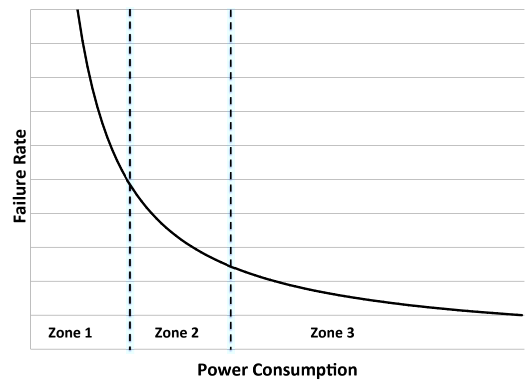

Figure 1. Tradeoffs on power consumption and failure rate

With power and reliability as two major factors, an ideal data center management policy shall aim to reduce power consumption while maintaining a low failure rate. Figure 1 illustrates three operational zones each representing a tradeoff between power consumption and failure rate. Generally speaking, as the power saving policy becomes more aggressive, the data center will consume less energy but run the risk of increased failure rates. Without considering the failure cost, one would develop power reduction algorithms as aggressively as possible and target Zone 1 as the optimal region. On the other end, if high reliability is desired, one would aim to reach Zone 3 for the minimal failure rate, although the power consumption would likely be the largest. Clearly, Zone 2 offers a potentially good balance of improving energy efficiency of data centers while minimizing the system failures.

In this work, we propose a new data center management framework, DUAL, which leverages the new VM power and reliability analysis tools to satisfy the *dual* needs of operating a data center - both reducing energy consumption and providing high reliability for VMs. We believe that an accurate VM power model shall take into account the frequencies of the processors that are controlled by power management policies. Similar approach can also be extended to study a reliability model that estimates VM failure rate. With the help of both models, system administrators can employ smart management policies that fine tune power saving techniques and control the impacts on the reliability of VMs. Prior research investigated power [9], [10] or reliability [4], [11] issues, but mostly in an isolated fashion. To the best of our knowledge, this work is the first attempt to address both challenges and makes the the following three contributions:

- We build a DVFS-aware VM power model that takes inputs from power measurement tools readily available in data centers to estimate server power consumptions. Experimental results show that the model can accurately estimate the power consumption of the server while CPU voltage is dynamically changed. Compared to a power model that does not consider DVFS, our model achieves a much better accuracy, reducing the estimation error from 18.5% to 6.7%.
- We establish a reliability model for VMs by using a soft error rate model and the VM failure rate based on a large-scale experiment of simulated fault injections.
- We propose a new DUAL framework that minimizes server power consumption with a guaranteed failure rate that can be specified by system administrators. The evaluation based on google data center trace shows that by using this framework, the power consumption is decreased by 10% while the failure rate of a data center is not affected.

The rest of the paper is organized as follows. Section II describes the proposed framework and VM power and reliability models. Section III presents the evaluation setup and results. Section IV discusses the related work and finally Section V concludes.

II. DUAL - Power and Reliability Analysis Framework

Our goal is to understand the tradeoff between power consumption and reliability in a data center. In such an environment, the infrastructure has a set of equipments (e.g., power distributions unit or PDU) and tools (e.g., Linux *top* and *iostat* commands) that can be used to collect a wide variety of system parameters, including CPU and memory utilization, CPU voltage/frequency, and disk I/O performance, as well as the failure events in CPU, memory, devices, and software (e.g., hypervisor and OS). The proposed DUAL framework utilizes these data to build the VM power and reliability models, which will assist in the selection of data center management policy. Figure 2 shows the architecture of the DUAL framework.

DUAL needs to address three challenges. First, while the impact of DVFS on performance and energy consumption has been extensively studied, its impact on the overall reliability in a data center environment is unknown. We intend to answer this question by providing new tools on power and reliability analysis. The second challenge is that data centers become increasingly virtualized. Cloud service providers usually provide the customers with VMs with various configurations and prices. As such, we need to quantify power consumption and reliability at the VM level, extending prior work on physical machines [9], [10], [4]. Third, VMs may have different reliability and power requirements. The DUAL framework shall provide the capability of fine-grained management. With DUAL framework, one can dynamically control power saving of physical resources based on specific requirements.

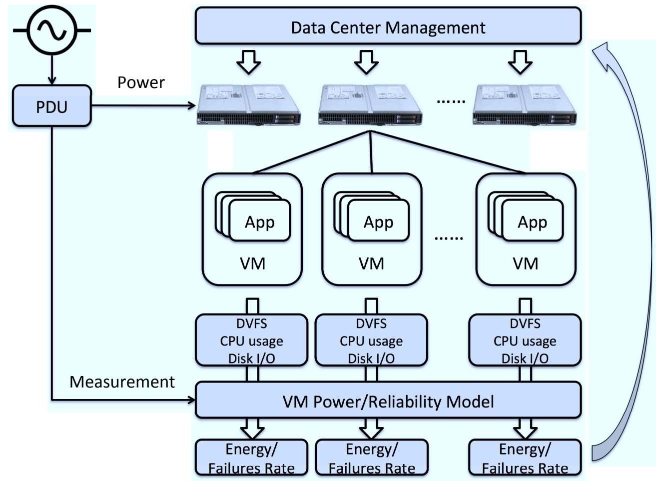

Figure 2. Proposed *DUAL* Framework

In the following, we explain the three key components: the VM power model, VM reliability model, and adaptive reliability and power management.

A. VM Power Model

VM power consumption cannot be measured with a power meter. Alternatively, we choose to model it based on system utilizations. We explicitly include the CPU frequency to model DVFS effects, which was not addressed in [12]. When system administrators adjust CPU frequencies to achieve the preferred power consumption, the model, without including the CPU frequency, may not be able to achieve high accuracy, as we will show shortly. To this end, we build a new VM power model by using the measurements on CPU frequencies, CPU utilizations, and disk utilizations. All of these data are readily accessible from operating systems, hypervisors, or data center management tools. We decide not to collect the DRAM statistics, as DRAM energy usage can be approximated as a constant. Because CPU frequencies are paired with voltages, here we use the frequency to represent the CPU voltage and frequency pair in the VM power model.

The power model for a server can be estimated as the sum of the CPU and disk power consumption. While other components such as network cards consume power as well, CPUs and disks are major components that reflect system activities.

$$power_{total} = \sum_{i=1}^{n}(power_{cpu,i}) + \sum_{i=1}^{n}(power_{disk,i}) + power_{idle} \tag{1}$$

where $power_{cpu,i}$ and $power_{disk,i}$ represent the power consumption by CPU and disk for VM_i, and $power_{idle}$ is the idle power of the physical machine - no application is running in the VMs. For each VM, the power consumption is calculated by aggregating its dynamic power portion, which is proportionally divided according to its CPU and disk usage, and an evenly divided portion of the server idle power. The accountable power of VM_i in a server with n VMs in total can be calculated as below:

$$power_i = power_{cpu,i} + power_{disk,i} + \frac{power_{idle}}{n} \tag{2}$$

The CPU power consumption of VM_i is estimated as:

$$power_{cpu,i} = a_i \times util_{cpu,i} + b_i \times util_{cpu,i} \times freq_i + d_i \tag{3}$$

where $util_{cpu,i}$ is the utilization for VM_i, and $freq_i$ is defined by the relative scaling of frequency between their minimum and maximum values. If a VM uses multiple cores with different DVFS states, the average value is used in this case. In the model, a_i and b_i are coefficients for CPU utilization and frequency, and d_i is a constant. Note that VMs running with different applications may have different power characteristics, even if their CPU/disk utilizations and voltages are the same. Therefore, each VM will have its own coefficients. In this equation, when the CPU utilization is zero, the power consumed by this CPU is nearly constant no matter which DVFS state it is currently in. When CPU utilization is non-zero, the DVFS state affects the CPU dynamic power proportionally. We find that this first order model achieves a relatively good accuracy, and plan to explore high-order models in future work.

The disk power is characterized as below:

$$power_{disk,i} = c_i \times util_{disk,i} + e_i \tag{4}$$

Here $util_{disk,i}$ is the normalized number of I/O accesses with respect to the maximum number of I/O accesses observed, c_i is the coefficient for disk utilization, and e_i is the idle power consumed by disks.

The total power consumption of a server can be derived by aggregating Equation 3, Equation 4 and constant idle power for other components of a system. Hence, the total power, $power_{total}$, can be represented as Equation 5.

Note that our model can be trained in runtime when the system experiences significant changes, e.g., many new applications come in, and a large number of new VMs are added.

$$\begin{aligned} power_{total} = &\sum_{i=1}^{n}(a_i \times util_{cpu,i}) + \\ &\sum_{i=1}^{n}(b_i \times util_{cpu,i} \times freq_i) + \\ &\sum_{i=1}^{n}(c_i \times util_{disk,i}) + power_{idle} \end{aligned} \tag{5}$$

Power Model Validation: We train and evaluate the proposed power model on a Dell PowerEdge R410 rack-mount server that has two Intel Xeon X5650 six-core processors, 24GB DRAM, a 250GB 7200RPM hard disk for operating systems, and a 1TB 7200RPM hard disk for data storage. We install Debian 6.0.6 64-bit operating system and Xen 4.0.1 hypervisor, and use a Watts up Pro power meter to measure the power consumption of the server. Four benchmarks from SPEC2006 [13] and PARSEC [14], are used to exercise CPU, memory, and disk I/O for model training. In particular, mcf, $bzip2$ and $canneal$ are chosen to represent user tasks, and $freqmine$ are chosen to represent the tasks run by cloud computing providers. One benchmark runs in each VM. For performance isolation, we pin the virtual CPUs of each VM to separate physical cores, and thus each server may run up to 12 VMs simultaneously.

While VMs are running, we collect the total power consumption and monitor CPU and disk utilizations and DVFS states at every second. We use different DVFS policies while benchmarks are running to make sure that a number of samples can be collected at each DVFS state. The timestamp is used to synchronize the sample data. The idle power is measured when all VMs are idle and only the monitoring programs are running in the host operating system. The model training can be formulated as an optimization problem, which is shown as below and solved by the least squares quadratic method:

$$\begin{aligned} &min\frac{1}{2}\left\|D \times x + power_{idle} - power_{total})\right\|^2 \\ &\qquad\qquad Subject\ to: \ x > 0 \end{aligned} \tag{6}$$

where D is the matrix of the CPU utilization, the disk utilization and the scaled frequency for each VM (the first three items in Eq. 5). For example, in the case of four VMs after n sample intervals, D is a n by 12 matrix. x is the coefficients that need to be trained. Since power is directly proportional to CPU and disk usages, all coefficients should be positive.

Figure 3 shows the measured and estimated power consumptions. One can see that the predicted power consumption closely follows the changes in measurement, with a mean error of less than 7% and a standard deviation of 4.5%. Further, we compare the accuracy of our DVFS-aware model with the model that does not include DVFS states (or the

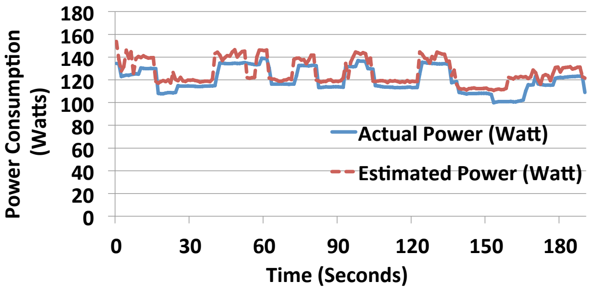

Figure 3. The VM Power Model Estimation

DVFS-unaware model, similar to the model used in [12]), and the comparison is shown in the Table I. The DVFS-unaware model yields a large absolute mean error, 18.7%, which is three times more than that of DVFS-aware model. Furthermore, the standard deviation of the DVFS-aware model is less than 5%, half of that of the DVFS-unaware model, which is another strong indicator that our model has a better accuracy. The reason is that the CPU P-state has a strong correlation with the power consumption. By including the CPU P-state, our model can quickly adapt to the changes in CPU P-state and adjust the power estimation accordingly.

Table I
MODELS ERRORS WITH AND WITHOUT CONSIDERING DVFS

	Mean Error	Std Dev	Max Error
DVFS-aware	6.7%	4.5 %	24.7%
DVFS-unaware	18.5%	9.7%	65.4%

B. VM Reliability Analysis

While a hard error can be detected by running tests in the factories, a soft error, which is the focus of our work, needs to be handled during the lifetime of the chip. Soft error rate (SER) is used to quantitatively measure the severity of soft errors. For 45nm technology, SER is estimated at one user-visible failure per month per 100 chips, which is expected to increase exponentially as technology scaling [15], [16]. In this work, we aim to understand the impact of soft errors on VM failure rate. Although fault injection to physical processor is feasible by introducing neutron beams, this practice is cost prohibitive and requires special equipments [17]. Alternatively, we conduct a large number of simulation based fault injections and construct the VM failure model based on the results.

On the high level, the VM failure model consists of three key parameters, SER, the masked rate of soft errors, and the failure rate for VM and host crashes. As soft errors may be masked and do not result in visible failures, the latter two parameters are used to model the failure rate from visible errors. Formally, the failure rate for one VM in a server with n VMs can be calculated as below:

$$\begin{aligned} R_{failure} = & 10^{\frac{d(1-F)}{1-F_{min}}} \times SER_{dvfs} \\ & \times (1 - R_{masked}) \times R_{crash} \times \frac{2}{n+1} \end{aligned} \tag{7}$$

Where R_{masked} is the percentage of masked soft errors, and R_{crash} is the percentage of errors that lead to VM crash and host crash, which is obtained from our fault injection experiments (to be described shortly). In a server with n VMs, the failure rate for each VM is $1/(n+1)$, assuming that the VM manager (VMM) could be treated as another VM, and the probability of one VM affected by a non-masked soft error follows uniform distribution. Note that unlike a failure in a VM that only results in crash within its own domain, a VMM failure would affect all VMs running on top of it. Therefore the actual failure rate for a VM is $2/(n+1)$.

SER_{dvfs} represents the SER with DVFS enabled that can be calculated using the base SER and the current frequency and voltage, formally written as [4]:

$$\begin{aligned} SER_{dvfs} &= r \times SER_{no\_dvfs} \\ &= 10^{\frac{d(1-F)}{1-f_{min}}} \times SER_{no\_dvfs} \end{aligned} \tag{8}$$

where r is the ratio between SER_{dvfs}, which is the new SER with DVFS enabled, and the SER_{no_dvfs}, which is the original SER without DVFS. Also, f_{min} corresponds to a minimum-energy frequency, and the variable F is defined by the relative scaling of frequency between minimum and maximum values. Note that the values are normalized frequencies with respect to the maximum frequency. The parameter d is a manufacturing technology dependent constant - a larger d means that the electrical circuits are more vulnerable to soft errors, and the SER will be changed more dramatically with the voltage. In our experiment, we use $d = 1$ to represent the potential tenfold increase in SER when the frequency drops to the minimum frequency [18].

In this work, we estimate the actual failure rate from non-masking errors by conducting fault injection experiments. Specifically, we use Simics [19] to simulate a full system with 4 processors and 2GB memory. Debian 6.01 and Xen Hypervisor 4.1.2 are installed within the simulated machine. Two VMs are running with the same benchmarks which are selected from SPEC2006, PARSEC and Filebench [20]. Each VM is assigned with one virtual CPU and is pinned to one physical processor. We conduct 2,000 injections for each benchmark, resulting in 8,000 injections total. The single bit flip soft errors are injected randomly into source registers. For each injection, only one error is injected, and the simulation continues to run. The outcomes are classified as 1) masked: the error does not trigger any visible faults at the user level; 2) host crash: the error results in a crash in the Dom0 or hypervisor, and reboot is required. In this case, all VMs are affected; 3) VM crash: the error results in a crash in one VM. Other VMs, Dom0 and hypervisor are running

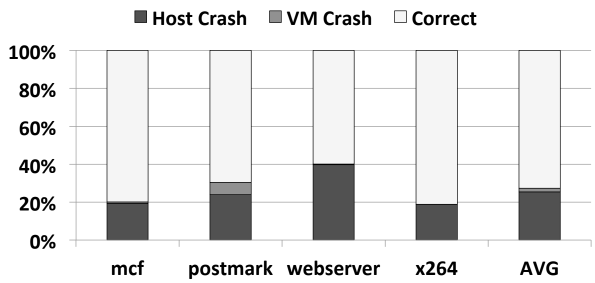

Figure 4. Fault Injection Results

correctly. The results are shown in Figure 4. On average, 27.4% of injections result in VM or system crashes. 2% of injections lead to VM crashes and 25.4% of injections lead to host system crashes.

C. Adaptive Reliability and Power Manager

In this paper, we mainly focus on the management policy that utilizes power saving techniques in a balanced manner to achieve both goals of reducing energy consumption and maintaining high reliability. Currently, there are several DVFS algorithms that have been implemented in modern operating systems. For example, Linux kernel has the *ondemand* and *conservative* governors. The ondemand governor dynamically adjusts the CPU voltage and frequency based on the current CPU utilization. If the utilization is higher than the specified *ondemand threshold*, which is often predetermined for all servers, the current voltage will be increased to the highest level. On the other hand, if the utilization is lower than this threshold, the current voltage will be decreased to the next lower level until it reaches the lowest level. The conservative governor is different in that the voltage is increased gradually when the utilization is higher than the *up threshold* instead of jumping to the highest level. The voltage and frequency pair decreases step by step when the utilization is lower than the *down threshold*.

Unfortunately, the default DVFS algorithms such as $ondemand$ do not consider the impact of failure rate of microprocessors, where higher failure rates may lead to higher occurrences of failures in the system. With our VM power and reliability models, the DUAL framework can now estimate VM-level power consumption and reliability and allow system administrators to balance the tradeoffs between them. Toward this goal, DUAL includes an adaptive power and reliability manager that consists of a feedback control algorithm that automatically adjusts the target reliability based on the actual failure rate, and a new DVFS algorithm that minimizes the power consumption at the failure rate which is dynamically determined by the control algorithm. Specifically, the control algorithm monitors the current failure rate and sets the desired reliability to increase the system reliability or to reduce power consumption. When the servers are experiencing a high failure rate, the controlled reliability should be increased to avoid potential failures in the future. When the actual failure rate is low, the system reliability should be adjusted to a moderate level to reduce power consumption. The feedback control theory is used to automatically achieve this goal. The input of this algorithm is the difference between the actual reliability and the target reliability. The output is the new controlled reliability. The control strategy can be formulated as:

$$R_k = R_{k-1} - a \times (R_{target} - R_{actual}) \tag{9}$$

where R_{actual} is the actual failure rate monitored by the algorithm; R_{target} is the target failure rate determined by system administrators and can be considered as constant in our experiment; R_{k-1} is the controlled failure rate which is set by the control algorithm at the time $k-1$; R_k is the controlled failure rate which is set by the control algorithm at the time k; and constant a quantifies the impact of the difference between the actual and target failure rates. The bigger value of a means a stronger control effect and a quicker response, but the system is less stable. Note that a can be fine tuned when systems are deployed.

After the control algorithm specifies system reliability, a new DVFS algorithm is proposed to achieve minimum power consumptions by utilizing our power and reliability models. The key observation is that the impact of DVFS on VM power consumption is not always same for all VMs. The applications can have their own power characteristics - for some applications power is sensitive to the CPU frequency, while for others increasing the CPU frequency does not significantly increase its power consumption. This can be confirmed from the coefficient of the frequency item in the VM power model. On the other hand, the processor reliability is relatively stable across different benchmarks, as shown in Figure 4. This provides a good opportunity to achieve the same reliability at the lower power consumption level.

The algorithm is listed in Algorithm 1. The current CPU utilization and disk I/O are measured and sent to VM power and reliability models. In this case, we not only use the models to estimate the current power consumption and reliability, but more importantly, we use them to predict the power consumption and failure rate for all possible combinations of CPU frequencies and then to find the one with lowest power consumption that satisfies the failure rate as specified by system administrators. This combination of CPU frequencies is then used to set the CPU frequency/voltage state. In this way, the system reliability is guaranteed with the minimum power consumption.

III. Evaluation

In this section, we conduct several case studies to demonstrate the benefits of the DUAL framework on analyzing and managing tradeoffs between power and reliability of a data center. The first case study consists of a simulation based on

Algorithm 1: Reliability-Aware Power Management

Data: cpu_i, $disk_i$ and $freq_i$ are CPU utilization, disk utilization and CPU frequency of VM_i. F_i is the list of all available frequencies of VM_i. n is total number of VMs; threshold is the targeted failure rate

Result: $freq_1$ ... $freq_n$

```
forall the freq_1 in F_1 do
    ...
    forall the freq_n in F_n do
        PredictedPower = PowerModel(cpu_1 , disk_1 , freq_1 ,..., cpu_n , disk_n , freq_n);
        PredictedFailure = FailureModel(freq_1 ,...,freq_n);
        if PredictedFailure ≤ threshold and PredictedPower ≤ power_min then
            power_min = PredictedPower;
            freq_min,1 = freq_1;
            ...;
            freq_min,n = freq_n;
        end
    end
end
return freq_min,1 .. freq_min,n;
```

the Google data center trace to show power reduction and reliability improvement when utilizing the DUAL framework. The second case study analyzes the annual cost of a data center when failure rate and power cost vary. In both cases, we use public Google trace to simulate the real data center operations [21]. The Google trace contains very detailed statistics of 12,580 machines running millions of tasks over one month, including the information for our models inputs, such as the duration of each task, the machine IDs that tasks are assigned to, and the CPU/disk utilization of each task. We assume that each machine has the same configuration as mentioned in Section II-A. We use the average disk times provided in this trace to represent disk utilizations.

A. Reliability Aware Power Management

In this experiment, we simulate the data center task allocations and then collect the power consumptions and failure rates. Here we assume that each server hosts two VMs. The jobs are assigned to VMs by hashing the job ID and VM ID. To match the interval in the trace, we use five minutes as the interval for DVFS sampling and changing. The power consumptions are normalized with the ondemand governor with up threshold set to 0.8. The failure rates are normalized to the baseline that is obtained by setting all CPUs to the lowest voltage/frequency. Figure 5 shows the normalized power consumption for the DUAL framework. While DUAL and *ondemand* have similar normalized failure rates, DUAL achieves much better energy efficiency, the maximum power saving is 10% and 5% on average. This result shows the effectiveness of the proposed DUAL on reducing power consumptions and maintaining failure rates.

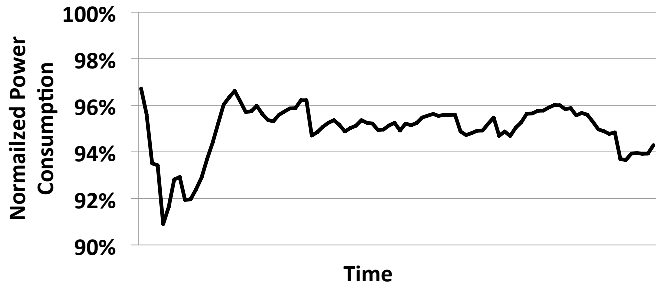

Figure 5. Power consumption of DUAL, normalized to that of *ondemand* policy. The average failure rate of DUAL is slightly lower than *ondemand*

With the capability of stabilizing failure rates, system administrators can set the controlled failure rates to meet the reliability requirements in real time, in order to balance the power consumption and failure rates. We conduct several experiments to demonstrate such application. Figure 6 shows the power consumption and system reliability adjusted by algorithms as the actual failure rate changes. In this experiment, the target failure rate of a data center is 30 failures per year. The actual failure rate is calculated by $failures/hours$. The system reliability is defined by Mean-Time-Between-Failure (MTBF), which is the inverse of the failure rate with the unit in hours. In the beginning (Phase 1), the actual failure rate is 0 indicating that system reliability is highest. In this case, the tradeoff can be made by reducing controlled reliability gradually for lower energy consumption. As the actual failure rate increases (Phase 2), DUAL is able to increase the controlled system reliability to avoid potential failures. As the failure rate decreases (Phase 3), DUAL gradually decreases the reliability again for power saving. These results show that DUAL is able to automatically track and adjust system reliability when observing the changes in the actual failure rate. Such capability is not possible with current *ondemand* policy.

B. Cost Analysis

In this case study, we use the VM power and reliability models to analyze the annual operation costs of a data center. For simplification, we define the annual cost to consist of three components: energy cost, failure cost, and amortized cost. The first two, the focus of this study, can be considered the dynamic portion of operational costs that vary on the amount of energy used and failures encountered. The amortized cost includes the expenses on personnel and equipments (e.g., physical infrastructure, network, power distribution and cooling systems). Formally, the cost func-

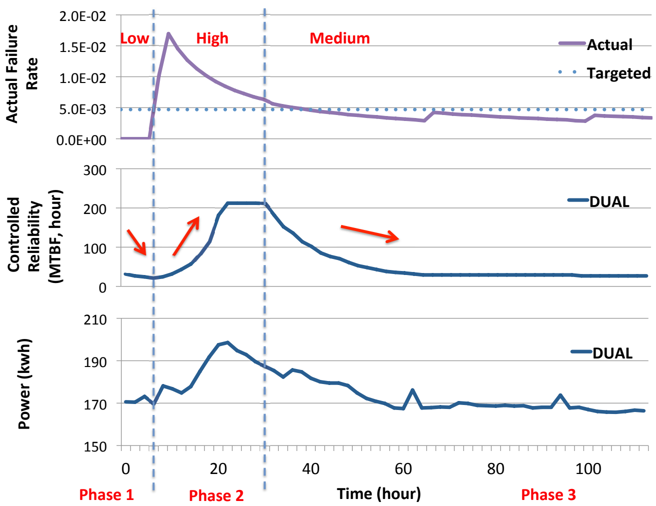

Figure 6. Power consumption and reliability controlled by DUAL as actual failure rate varies. In Phase 1, the actual failure rate is 0, and DUAL decreases the controlled reliability deliberately to reduce the power consumption; in Phase 2, the actual failure rate increases significantly, and DUAL will increase the controlled reliability accordingly; and in Phase 3, as the actual failure rate decreases slowly, DUAL gradually decreases the reliability again for power saving.

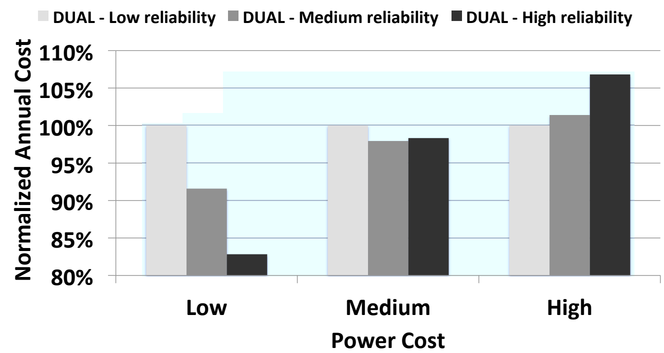

Figure 7. Annual operational cost of a data center. Low, medium and high electricity rates are defined as \$0.025/kwh, \$0.07/kwh,\$0.25/kwh respectively. Low, medium and high reliability are defined as 80%, 50%, 20% of the baseline failure rate that is calculated without DVFS.

tion can be written as follows:

$$Cost = \sum_{i=1}^{n}(T_i \times Cost_{failure}) + Cost_{power} + Cost_{amortized} \quad (10)$$

where $Cost$ is the total annual cost, n is the total number of failures, T_i is the duration of the i-th failure in hours, and $Cost_{failure}$ is the cost per hour for a failure. Note that the cost of failures depends on many factors and usually differs from data centers. $Cost_{power}$ is the cost of power consumption in a year, and $Cost_{amortized}$ is the cost amortized for that period of time.

In this study, we vary the electricity rate from \$0.025/kwh, \$0.07/kwh, and \$0.25/kwh, labeled as low, medium, and high rates [22]. The VM power model is used to calculate the power consumption of over 12,000 machines in the Google trace. Estimating the cost of a failure is more complicated and involves the failure rate and recovery cost. The base error rate is defined as the MTBF of 500 years [23]. Recall from the previous section, our fault injections reveal that 27.4% of faults result in VM or host crashes. As a result, seven failures in a year are expected in the data center of interest. For each failure, the affected system is expected to recover by rebooting and restoring applications to correct states. Therefore, the downtimes are relatively short - we approximate with an exponential distribution with a mean of 30 minutes. This estimation gives the total downtime ranging from 1 to 170 hours, consistent with the report [24], which serves as the baseline for the ondemand policy. The number of failures with the new DUAL framework can be calculated by multiplying this baseline and the normalized failure ratio that we obtain from the experiments. Because recovery cost is estimated at 4% of the overall failure cost [8], we take the estimated failure cost for Amazon in [24] and calculate the average recovery cost, \$7,200 for each failure.

Figure 7 shows the annual cost of the data center. The results are shown by different targeted reliability and electricity rates. Low, medium and high reliability are defined as 80%, 50% and 20% of the baseline failure rate when DVFS is disabled (highest reliability). For the same electricity rate, the results are normalized with respect to the one with low reliability. One can observe some interesting trends in the figure. If the power cost is high, the failure cost becomes less important, and can be compensated by power savings. In this case, the lowest annual cost can be achieved with the lowest reliability. On the other hand, if the power cost is low, the reliability cost dominates the overall cost. In this case, the lower annual cost can be achieved by ensuring the maximum reliability. However, in the case of medium power cost, the lowest annual cost can be achieved at medium reliability, indicating a balance of power saving and system reliability in the data center. This quantitative analysis suggests that in order to minimize the operational cost of a data center, a smart operator should study the historical failure data and carefully select an optimal power management policy by taking into consideration both reliability and power requirements.

IV. Related Work

Power Modeling: Previous work investigated power models for computer systems [25], [26], [27], but usually in non-virtualized environments. The work closely related to ours is [12], which uses CPU/disk/memory utilizations to model the server power consumption. However, it does not consider CPU DVFS states, making it less accurate than the proposed VM model in this paper.

Soft Errors: The impact of DVFS on soft error rate in processors has been studied, for example, [4] designs a dual modular redundancy (DMR) technique to stabilize the SER in multicore processors, but DMR induces hardware and performance overheads and is not available in commodity processors. In this paper, we extend an existing analytical SER model with a new set of fault injection experiments to estimate the actual failure rate. In this work, we focus on the soft errors in processors and plan to explore other components of a computer system in the future. In [28], fault injection experiments are conducted to understand the soft error propagation behaviors. In this work, we conduct fault injections for applications running in virtualized environments.

DVFS has been explored for power management in data centers [29], [30], [10]. However, they rarely consider the reliability as a constraint. As we point out in Section I, this may increase the overall operational cost rather than reducing it. We quantify the relationship between the reliability and power consumption in our DUAL framework, providing system administrators with the new capability of optimizing the overall operational cost.

V. Conclusions

In this work, we propose the DUAL framework that helps system administrators optimize both power consumption and reliability in data centers. In the DUAL framework, we build a VM power model that explicitly includes the impact of DVFS algorithms. Experiments show that the new power model can estimate power consumptions with higher accuracy. We also propose an adaptive power manager that minimizes server power consumption while maintaining high reliability. The evaluations demonstrate how to use the DUAL framework to quantitatively analyze the tradeoff between power consumption and reliability, where several case studies are conducted based on real data center traces. The results show that DUAL can help data centers to reduce up to 10% of power consumption, and the optimal cost can be achieved when considering different energy and failure costs, as well as various reliability requirements.

In future work, we plan to study more complicated cases such as more than one VM sharing one core, investigate new ways to further reduce the overheads and improve the effectiveness of the DUAL framework, and explore opportunities to deploy and evaluate DUAL in a production environment.

Acknowledgments

This work is supported in part by NSF grant OCI-0937875.

References

[1] C. Belady, "In the data center, power and cooling costs more than the it equipment it supports," *Electronics cooling*, vol. 13, no. 1, p. 24, 2007.

[2] S. Borkar, "Microarchitecture and Design Challenges for Gigascale Integration," in *MICRO*, 2005, keynote Address.

[3] B. Zhao, H. Aydin, and D. Zhu, "Reliability-aware dynamic voltage scaling for energy-constrained real-time embedded systems," in *ICCD*, oct 2008.

[4] R. Vadlamani, J. Zhao, W. P. Burleson, and R. Tessier, "Multicore soft error rate stabilization using adaptive dual modular redundancy," in *DATE'10*, 2010, pp. 27–32.

[5] E. Normand, "Single event upset at ground level," *Nuclear Science, IEEE Transactions on*, vol. 43, no. 6, pp. 2742 – 2750, dec 1996.

[6] J. Ziegler and H. Puchner, *SER - History, Trends and Challenges: A guide for designing with Memory ICs*. Cypress, 2004.

[7] R. Baumann, "Soft error rate overview and technology trends," in *Reliability Physics Tutorial Notes, Reliability Fundamentals*, April 2002.

[8] E. N. Power, "Understanding the cost of data center downtime: an analysis of the financial impact on infrastructure vulnerability," in *white paper*, 2011.

[9] Lim et al., "Power budgeting for virtualized data centers," ser. USENIX ATC'11.

[10] G. von Laszewski, L. Wang, A. Younge, and X. He, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in *Cluster*, Sep. 2009, pp. 1 –10.

[11] S. Yin, X. Ruan, A. Manzanares, Z. Ding, J. Xie, J. Majors, and X. Qin, "Improving reliability of energy-efficient parallel storage systems by disk swapping," in *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, Dec., pp. 87–94.

[12] K. et al., "Virtual machine power metering and provisioning," ser. SoCC '10.

[13] J. L. Henning, "Spec cpu2006 benchmark descriptions," *SIGARCH Comput. Archit. News*, vol. 34, no. 4, pp. 1–17, Sep. 2006.

[14] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.

[15] Shivakumar et al., "Modeling the effect of technology trends on the soft error rate of combinational logic," in *DSN*, 2002, pp. 389–398.

[16] Feng et al., "Shoestring: probabilistic soft error reliability on the cheap," in *ASPLOS'10*.

[17] Biswas et al., "Explaining cache ser anomaly using due avf measurement," in *HPCA*, 2010, pp. 1–12.

[18] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in *IRPS*, april 2011, pp. 5B.4.1 –5B.4.7.

[19] Virtutech. (2006) Simics Full System Simulator. [Online]. Available: http://www.simics.net

[20] filebench, "file bench," Online, 2012, http://filebench.sourceforge.net/.

[21] Google, "Goolge cluster data version 2," Online, 2011, http://code.google.com/p/googleclusterdata.

[22] J. Hamilton, "Overall data center costs," http://perspectives.mvdirona.com/2010/09/18/OverallDataCenterCosts.aspx.

[23] D. C. Bossen, "CMOS Soft Errors and Server Design," *IEEE 2002 Reliability Physics Tutorial Notes, Reliability Fundamentals*, vol. 121, pp. 07–1, 2002.

[24] International Working Group on Cloud Computing Resiliency, "Downtime statistics of current cloud solutions," Online, 2012, http://iwgcr.org/wp-content/uploads/2012/06/IWGCR-Paris.Ranking-002-en.pdf.

[25] Tiwari et al., "Instruction level power analysis and optimization of software," *Journal of VLSI Signal Processing*, vol. 13, pp. 1–18, 1996.

[26] F. Bellosa, "The benefits of event: driven energy accounting in power-sensitive systems," in *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, 2000.

[27] Brooks et al., "Wattch: a framework for architectural-level power analysis and optimizations," *SIGARCH Comput. Archit. News*, vol. 28, no. 2, pp. 83–94, May 2000.

[28] Xu et al., "Understanding soft error propagations using efficient vulnerability-driven fault injection," in *DSN'12*.

[29] Kolpe et al., "Enabling improved power management in multicore processors through clustered dvfs." in *DATE'11*.

[30] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," in *ISLPED*, aug 2007, pp. 38 –43.