Power Control for GPU Clusters in Processing Large-scale Streams

Qingkui Chen[12] Haifeng Wang[2,3] Bocheng Liu [2]
1. School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China
2.University of ShangHai for Science and Technology
Shanghai, School of Management, Shanghai, China
3.Lin Yi University, Linyi, China

***Abstract*—Many emerging online data analysis applications require Large-scale streams data processing. GPU cluster is becoming a significantly parallel computing scheme to handling large-scale streams data tasks. However power optimization is a challenging issue. In this paper, we present a novel power consumption control model to shift power budge among nodes in the cluster based on their real workload needs, while capping redundancy energy and controlling the total power budge of the cluster to keep or below a constraint imposed by its power supplies. Our controller is very suitable to the dynamic workloads task model and designed based on an Multi-Input_Multi-Output control theory. We analyze the power consumption behaviors of GPU cluster and the variation of workload. The detailed control problem formulation is presented and analyzed in theory. We finally conduct simulation experiments on a physical cluster to compare our controller with two state-of-the-art controllers. The experimental results demonstrate that our proposed controller outperforms the other controllers by having more accurate control and more stability.**

***Index Terms*—Power Consumption Control; Power Consumption Management; GPU Clusters; Model Prediction Control**

I.INTRODUCTION

Power management is one of the critical issues in High-performance field. The energy cost has become a major factor in the total cost of ownership of large-scale server clusters. Espeically there is a wide spectrum of applications that require real time processing of data streams, e.g. financial data analysis, processing the large-scale sensor networks output information, social computing, and network security, etc[1]. Due to the fact that Graphics processing units have significantly developed in the general-purpose computing field. GPUs cluster has become an important parallel computing scheme for large-scale data streams applications. The GPU clusters are working all day and night. So computing and cooling both consume a large amount of energy. For example, running a single 300W GPU for one year could consume 2628KWh of energy, with an additional 748kWh for cooling[2]. So it is important to energy cost optimization and energy capping for GPU clusters aiming at the large-scale data streams tasks. The energy optimization of GPU clusters include analysis the energy consumption characters of GPU clusters in real computing and adjusting power of each GPU nodes to reduce the redundancy energy consumption. Note that energy optimization should not degrade the performance of the GPU clusters.

The GPU is a multicore processor optimized for graphics workloads. For the general-purpose computing it reconfigure and reduce special logic units to augment general-purpose cores. Compared to the CPUs, GPUs have different architectures by introducing new forms of technology heterogeneity. Different technologies have intrinsically different performance delay and energy consumption. But the researchs about the power controlling and optimization of CPUs clusters achieve many research results that are beneficial to GPUs clusters. For example, power optimization of the heterogeneous web server clusters that dynamically determining the node's power state according to the workload to achieve global optimal solution. This paper aims at studying dynamic power controlling of GPUs clusters based on the feedback control theory. Traditionally, adaptive power management solutions mainly rely on heuristics. Recently, T.Horvath applied feedback control theory to power optimization for multi-layer web server clusters[3]. Luciano combined feedback control theory with mixed integer programming to build the power controlling model for web server clusters[4]. X. Wang proposed an power controlling model based on an optimal Multi-Input-Multi-Output(MIMO) control theory for high-density servers in modern data center[5]. This power controller controls the power consumptions of multiple servers in an enclosure simultaneously by manipulating the performance setting of each node.

The power optimization of GPUs that just focus on the single computing node has been studied since 2010. Lin analyzed the computation warp parallelism and memory warp parallelism of the GPUs programs. Then dynamic voltage frequence strategy was proposed based on power consumption model[6]. Lin decomposed a task to many subtasks and analyzed the dependency relationship

doi:10.4304/jcp.8.10.2489-2496

among all of the subtaks. Reducing power consumption of subtasks in the critical path is the key technology to optimize the computing energy. Additionally, power consumption prediction models were built from different levels to aid programmers to learn power consumption profile of applications[7,8,9]. However there is few research related to GPU clusters[10], especially the multi-tasking on GPU clusters. In this paper, we aim at the large-scale data streams task model that has variational workload. The workload has three status: peak times, normal times and abnormal times. Reducing the low utilization nodes' power budget in normal times is an significantly way to optimize the total energy consumption. On the other hand, Capping redundancy energy in peak times or abnormal times play an important role in preventing the energy budget overload of the GPU clusters.

In this paper, we propose a novel power controlling model for GPU clusters by considering the unexpected workload in large-scale streams data real processing. A controlling model is built based on Model Predictive Control theory from the whole cluster perspective to cap redundancy energy. The rest of the paper is organized as follows: Section 2 introduces GPU power consumption features and power state adjusting technologies. Section 3 describes related models. Section 4 discusses the design schemes of CPU clusters that may apply into the GPU clusters. Then provides the implementation details and theory analysis of the controlling model. Section 5 presents the results of our experiments conducted in GPU cluster. Finally concludes the paper..

II. PRELIMINARIES

A. Power Consumption

The architecture of GPUs are different from CPUs. So the power consumption characters of GPU clusters are not the same as CPU clusters[6]. Each computing node is heterogeneous processing unit that contains CPU controlling the communication and GPU handling the computing tasks. In addtion, the task model has an important effect on power consumption. There are two factors of task model for the computing energy. The first is communication density that is the ratio between the computing and the communication time. The other is arithmetic density of GPU computing[5]. The different task models have different communication and arithmetic density. So it is significantly to consider the task model effect to the power consumption.

B. Power Consumption Adjusting

Adjusting the power consumption state is the primarily technology to eliminate CPU and servers idle power. Traditionally, dynamic voltage and frequency scaling is to adjust the processor's power consumption state[7]. Currently the GPU's hardware doesn't support the dynamic voltage and frequency scaling mechanism. But there are three ways to adjust the GPU power consumption state. The first one that can provide realistic data for experiments is to adjust the hardware configuration. GPU companies allow to adjust the power consumption state by reconfigure the motherboard. The second way is that the operation systems allow to transform the power consumption state through the power configration interface[11]. For example, OpenSUSE 10.3 provides the Advanced Configuration and Power Interface (ACPI) to change the core frequency of GPUs[12]. Thirdly,some applications may adjust the GPU power consumption state. Such as AfterBurner and Dynamic Over-Clocking Technology of GTX280[13].

C. Data Streams Task Model

The large-scale data streams task is a data streams set that contains continuous streams. Namely $\Gamma = \{S_1, S_2, ..., S_n\}$, S_i denotes as stream. There is strictly partial ordering relation between S_i and S_{i+1}. S_i can be defined as (A_i, C_i, T_i, P_i). A_i is arithmetic density of the stream[14]. C_i is communication denstiy that is the ratio of communication time and computing time. T_i represents the computing time on GPUs. P_i is the status of stream that are normal, abnormal and peak status. The workloads of streams in peak status generally are more than that in normal status.

III GPU CLUSTER CONTROL DESIGN AND ANALYSIS

A. Exsiting Control Model

Power consumption is one of the most important design constraints for CPU clusters. Much of the prior works have attempted to reduce power consumption by imporving the energy efficiency. These research results are also beneficial to the GPU clusters. In this section, we will discuss three typical power controlling approaches that can be apply to the GPU clusters' power consumption control.

We use three state-of-the-art controllers, referred to as dynamic power controller, SISO, and Improved SISO, as baselines to compare our proposed controlling scheme. The dynamic power controller is a typical industry solution to power consumption control and briefly summarized as follows: Start with the GPUs in the cluster throttled to the lowest frequency level. Each group GG_i is set a specific set point θ_i. In each control period, if the total power consumption of the group is lower than θ_i, choose the node with the highest GPU utilization to increase its frequency level. And if the power consumption is greater than θ_i, So choose the lowest GPU utilization to reduce its frequency level by one group. Note that when selecting the GPU node to adjust its frequcency level, if there are the same GPU untilization, choose a node in a round robin fashion. This controller is denoted as Ad Hoc.

The second power control strategy is Single-Input-Single-Output (SISO) that is a control-theoretic controller designed to control a signle node's power consumption[15]. Due to the fact that SISO is a separate controller used on each node to control its power

independently from other nodes in the cluster. The power budget of each node is calculated by evenly dividing the total budget of the cluster by the number of nodes rather than considering the power budget allocation from the whole cluster perspective. So SISO is impossible to predict which node would have more workload and need more power budget at runtime.

The third baseline is Improved SISO(ISISO) that is SISO with an group-level supervisor to periodically shift power among different nodes accordingly to their GPU untilizations. In each control period, the supervisor manages the power allocation in each group. The current power consumption of the group is lower than the set point, the supervisor chooses the node with the highest GPU utilization to increase the power budget. To enforce the group-level power set point, the supervisor chooses the lowest GPU utilization node to decrease its budget. When the group power is greater than the set point, the supervisor reduces the highest utilization node' power budget, and increases the lowest utilization node's budget to reduce redunancy power and enforce the set point. Compared with SISO, ISISO is a two-layer control model based on the group model. Therefore, ISISO guarantees the stability of the power control system than the SISO controller. SISO and ISISO need the information of the GPU utilization. We use the software GPU-Z to get the GPU utiliztion.

B. Model Predictive Control

In this section, we propse a novel MIMO control model based on the Model Predictive Control (MPC) theory to manage the power of GPU cluters for real processing the large-scale data streams tasks. Because the nodes in GPU clusters are generally coupled together and the power consumption cannot be considered and controlled independently from each other. Our proposed control model is a MIMO power controller that is much better suited for power management in an cluster than SISO control model.

The kernel control strategy is Model Predictive Control theory that can deal with coupled MIMO control problems with constraints on the plant and the actuators. This feature makes MPC well suited for power control of cluster enclosures. And our proposed control system is denoted as MPC. Since Model Predictive Control is a control strategy growing up from the practice industry process and includes the predictive models, rolling optimization and feedback compensation. It is takes on strong robustness[16].

1.Controller design

This controlling system includes main controller, power consumption monitor, model predictive module, utilization monitor and actuators. The controlling system works as follows: The power monitor measures the power of each node in every control period and sends that to the main controller. Each node's utilization monitor gathers the GPU utilization and sends to the main controller. The main controller collects the power value and utiliztion value and sends these data to the model predictive module. The model predictive module optimizes the prediction model by the feedback information. The main controller updates the control parameters and then sends the control command to each node's actuator that change the node's power state.

2. System Modeling

In this section, we build the model of the dynamics of the power controlled system with the well-established approach called system identification. The goal of the control system is to guarantee the total energy consumption value converge to ideal value P_s, namely the controlled variable $CP_{(k)}$ converges to P_s. In order to design an effective controller, it is important to model the dynamics of the controlled system that is the relation model between the controlled variable and the manipulated variables. Here the manipulated variables are the frequency level of GPU and the number of active SMs. The total energy consumption value is the controlled variable. The GPU cluster's energy consumption vary with the change of each node's power consumption state. Then we introduce the notation as follow table 1.

TABLE 1
CONTROL SYSTEM NOTATION

Symbol	Describe
T	Control period
$P_i(k)^f$	power consumption of node i (frequency part)
$f_i(k)$	The frequency level of the node I
$P_i(k)^{sm}$	power consumption of node i (SM part)
$P_i(k)$	The total power consumption of node i
CP(k)	The total energy consumption of cluster
$\Delta f(k)$	The difference frequency
$\Delta SM(k)$	The difference number of SM
P_S	The power set point

We use two approaches to change the GPU power state. The first way is to adjust the frequency level from the GPU hardware. And the other way is to reconfigure the number of threads to control the active SM number[17,18]. So the frequency level of GPU and the number of SM are controlled variables. We infer the relationship by collecting data from experiments and establish a statistical model between the power consumption and the frequency, the active SMs number.

A GPU has several discrete frequencies in specific frequency range, $[f_{\min} < f_i < f_{\max}]$. So the computing frequency of GPU is a frequency set, namely $F = \{ f_i \mid f_{\min} < f_i < f_{\max}, i = 1, 2, ..., t\}$. Where $f_{\min}$ and $f_{\max}$ denote the minimum and the maximum frequncy in normal working condition. Although the relation model between the processor frequency and the power consumption is not perfectly linear. For the simplicity, we linearize the model as Equ. 1.

$$P_i(k)^f = a_i f_i(k) + b_i \quad (1)$$

Where a_i is a generalized parameter varying for different GPUs. k is the kth control period and b_i is linear parameter.

Although the frequency determines the power consumption. The number of active SMs has also effect on the power consumption. So the power consumption of GPU include two parts as Equ. 2.

$$P(k) = P(k)^f + P(k)^{sm} \quad (2)$$

Where $P(k)^f$ is main term of the power consumption determined by the frequency. $P(k)^{sm}$ is the second term determined by the active SMs number. Then we should consider the effect of the active SMs on the power consumption. Suppose that the active SMs number is $SM_i(k)$. The relationship model between $P(k)^f$ and the active SMs is as Equ. 3[8].

$$P_i(k)^{sm} = c_i SM_i(k) + d_i \quad (3)$$

The power consumption model in k control period is as follows combined the Equ.2 and Equ.3.

$$P_i(k) = a_i f_i(k) + b_i + c_i SM_i(k) + d_i \quad (4)$$

So the power consumption model of ith node in k+1th control period is

$$P_i(k+1) = P_i^f(k) + a_i \Delta f_i(k) + c_i \Delta SM_i(k) \quad (5)$$

Where $\Delta f_i(k) = f_i(k+1) - f_i(k)$ and $\Delta SM_i(k) = \Delta SM_i(k+1) - \Delta SM_i(k)$. Based on Equ.5, we consider the total power consumption of GPU cluster in the k+1 th control period. Its power consumption can be modeled in the matrix form:

$$P(k+1) = P(k) + A\Delta f(k) + C\Delta SM(k) \quad (6)$$

Where $P(K) = [p_1(k), p_2(k), ..., p_n(k)]^T$,

$A = diag[a_1, a_2, ..., a_n], C = diag[c_1, c_2, ..., c_n]$,

$\Delta f(k) = [\Delta f_1(k), \Delta f_2(k), ..., \Delta f_n(k)]^T$,

$\Delta SM(k) = [\Delta SM_1(k), \Delta SM_2(k), ..., \Delta SM_n(k)]^T$.

Assume the GPU cluster has N nodes. Then the total power consumption of GPU cluster is the sum of all the node's power in the cluster.

$$CP(K) = \sum_{i=1}^{N} P_i(k) \quad (7)$$

The time difference between the k+1 th and the k th control periods is Δt. The energy consumption of the GPU cluster in the k th control period is

$$E(K) = \sum_{i=1}^{N} (a_i f_i(k) + b_i + c_i SM_i(k) + d_i) \times \Delta t \quad (8)$$

Note that we use the total power consumption $CP_{(K)}$ as the controlled variable to simplify the controller design rather than the energy consumption $E_{(K)}$.

3. Control system analysis

The key design of the controlling system is the controller. The model predictive controller includes the system model, cost function, reference trajectory and a least-squares solver. The controller uses the system model to predict the control behavior over control period. The system model defined in Equ.6 has been built in system identification. The cost function is defined as

$$V(k) = \sum_{i=1}^{p} \left\| cp(k+i|k) - ref(k+i|k) \right\|_{w(i)}^{2} + \sum_{i=0}^{M} \left\| \Delta f(k+1|k) + f(k+i|k) + \Delta sm(k+1|k) + sm(k+i|k) - F_{\max} - SM_{\max} \right\|_{R(i)}^{2} \quad (9)$$

Where P denotes the prediction horizon and M denotes the control horizon. $W_{(i)}$ is tracking error weight matrix and $R_{(i)}$ is the control penalty weight matrix. The first term of the Equ.9 represents the difference value between the prediction value and the future reference value. The future reference value is generated by a reference trajectory $ref(k+i|k)$ that is defined as a ideal trajectory along which the total power consumption of cluster should change from the current value to the set point. Our reference trajectory is an exponential reference trajectory to rectify the control output.

$$ref(k+i|k) = P_s - e^{-\frac{\alpha T}{T_{ref}} i} (P_s - cp(k)) \quad (10)$$

Where T_{ref} is the time constant that specifies the system response speed. The smaller value of T_{ref} is, the faster converge speed to the set point is. The second term in the equation 9 represents the control penalty that is to minimize the oscillation of the manipulated variable, such as the frequency level of GPU. Finally, our power control problem can be transform to a least-squares problem. So the controller uses a least-squares slover to solve the control model's parameters.

The power control system works as follows: the controller predicts the control behaviors based on the system model in the prediction horizon. The control input will be selected that has a minimized cost function under the constraints. Here the control input trajectory includes the control inputs in the following M control period, $\Delta f(k)$, $\Delta f(k+1|k)$, $\cdots$, $\Delta f(k+M-1|k)$; $\Delta SM(k)$, $\Delta SM(k+1|k)$ $\cdots$, $\Delta SM(k+M-1|k)$. Once the control input trajectory is calculated, only the first elements $\Delta f(k)$ and $\Delta SM(k)$ are applied as the control input to this system. The prediction horizon slides one control period and the control input is computed agained based on the feedback $CP_{(k)}$ from the power monitor. It is important to recomput the input trajectory due to the fact that the original prediction may be incorrect to the uncertainties of the workload of large-scale data stream task.

IV EXPERIMENTS

In this section, we present the experimental results conducted in the real GPU cluster. Our experimental data

includes the stimulation data and the real data generated from the engineering projects. We compared MPC against the Adhoc and ISISO, in term of control accuracy and application performance, using the artificial and real experimental data. Here the power measurement card was designed to capture the power consumption value in real-time computing. It converts the current signal to voltage signal and record into the computer.

A. Baselines and Setup

The main idea is to measure the power consumption of the GPU cluster and compare the difference between the measured power value and the set point denoted as δ. The smaller value of the difference is, the higher accuracy of control system is.

Our GPU cluster is composed of 25 nodes. A Ubuntu desktop machine is running as the controller and the others nodes running Ubuntu server are responsible for computing. The controller machine is a Dell Latitude Dell 610 with 1.7 GHz Intel Pentium processor and 2GB RAM. The twenty-four nodes are equipped with 2.33GHz Intel Corel processors with NVIDIA GeForce GTX 280 or NVIDIA GeForce GTX 480. All the machines are connected via an internal Ehternet Gigabit switch.

B. Stimulation Experiments

We introduce the stimulation details of the artificial data. The program set R is firstly selected from the CUDA SDK and the benchmark provided in literature[18]. R denotes as R = {R_1, R_2, ⋯R_k}. We then randomly select different program R_i from R with normal distribution and combined these programs into a program R_t. R_t is a sequence of R_i that has different arithmetic denstiy, such as, R_t ={ R_1, R_1, R_3, R_m⋯R_k, R_5…..}. From now on, in all experiments, for the sake of comparison, the execution time should be greater than 3600s. The set point is set to 4150W. As shown in Figure 1, the dotted line represents the set point.

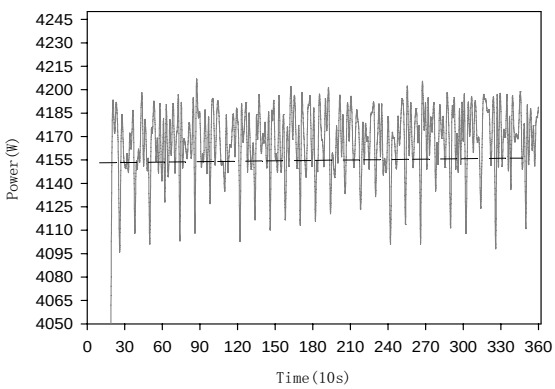


Fig1 Variation of power consumption with Ad Hoc controller

Compare to Ad Hoc controller, MPC has more control accuracy. Since Ad Hoc just simply raises or lowers the power consumption state by one step when the measured power is lower or higher than the set point. Figure 1 shows δ is largest among the three controllers. And the variation range of power is between the +60W and -55W. The start-up time that is from the begin to the set point is longest and in this experiment is about 280s. Ad Hoc has control delay since when the measured power is lower or higher than the set point, the controller needs more time to adjust the control input. The control delay is about between 2 and 3 control periods. Additionally, due to the randomness of the Ad Hoc control model in the group level, this controller should take a relatively long time to converge the set point.

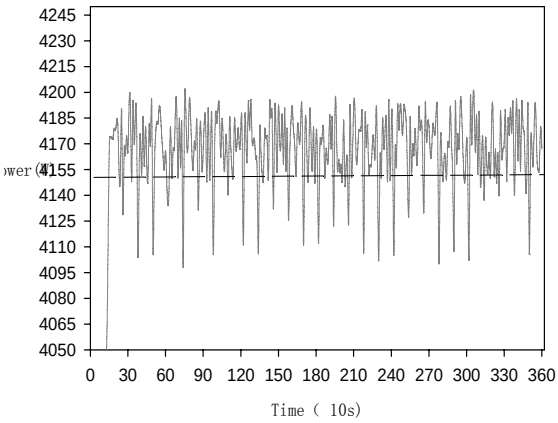


Fig2 Variation of power consumption with ISISO controller

Compared to Ad Hoc, SISO is a proportional controller based on well-established control theory. So ISISO has higher control accuracy in contrast to Ad Hoc. Figure 2 shows a typical run of ISISO controller. δ is less than Ad Hoc and the variation range is about [-50W, +48W]. Because the power budget of each node in cluster is calculated by evenly dividing the total budge by the number of nodes in this cluster. In contrast to Ad Hoc, the control delay of ISISO becomes smaller. The variation range is smaller that also can prove this fact. In short, the experimental results indicate that the control accuracy of ISISO is higher than Ad Hoc. Due to the fact that ISISO is a separate controller used on each node to control its power independently from other nodes in the cluster. This measure makes the controlling system with locality randomness that is different from the global randomness in the Ad Hoc system. So as shown in Figure 2, the start-up time of ISISO is significantly less than Ad Hoc.

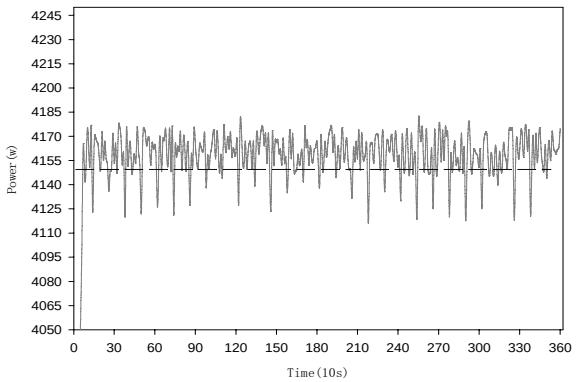


Fig3 Variation of power consumption with MPC controller

The proposed control system based on MPC explicitly incorporates the interprocessor coupling in an cluster into the MIMO model and control the power consumption from the cluster perspective. So As shown in Figure 3 δ is smallest in the three control schemes. And the variation range is about in [-30W, +33W]. This results prove that the proposed controller has the best control accuracy. Since the MPC can predict the future control behaviors based on the historical data, the control delay of MPC controller is the smallest and it has the fastest converge speed. Note that due to regardless of extra energy loss in computing phase, most of the measure power is greater than the set point.

C. Engineering Data Experiments

In this section we will show some power optimization results and compare MPC with the other two control schemes using the real data from engineering project. The real data are from a 3G mobile video quality monitor system and the Intrusion detection system of the Lincoln lab. The real data from the engineering projects has three states: normal, peak and abnormal. It is difficult to simulate the three states for the artificial data. So this experiments is very meaningful for the power control system. The large-scale video streams data is collected in the RTP packages and combined into TS format[19]. We use three machines to send the video streams data and the GPU nodes are mainly responsible for the H.264 decompression or the video quality feature extraction[20]. The data of intrusion detection system is large-scale packages in Tcpdump format[21]. The GPU nodes handle with the string filtering tasks. Note that we focused on the power controlling of GPU cluster rather than the workload balance. For the sake of simplicity, the assignment of workload in the cluster has been done with manual intervention.

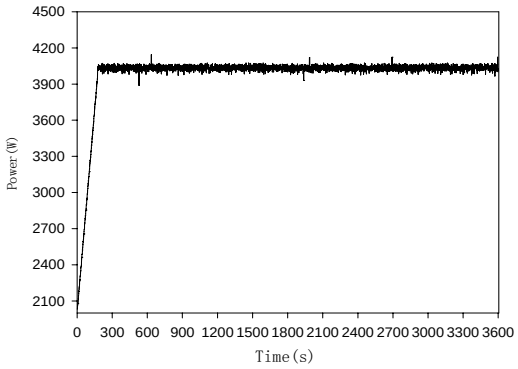


Fig4 Video Quality Monitoring System under the proposed controller

Figure 4 shows the result of running the proposed controller to handling the video quality monitor task with the set point set to 4000W. The start-up time is about 300s and δ is about from -32W to +35W. The arithmetic density of video quality monitor tasks are stable and the loading time of data streams is mainly in normal distribution. Thus the power consumption is controlled steadily in the range of set point and just appears seven deviations. Additionally, the start-up time is longer than the simulation experiments due to the longer loading and assignment time of the large-scale data streams.

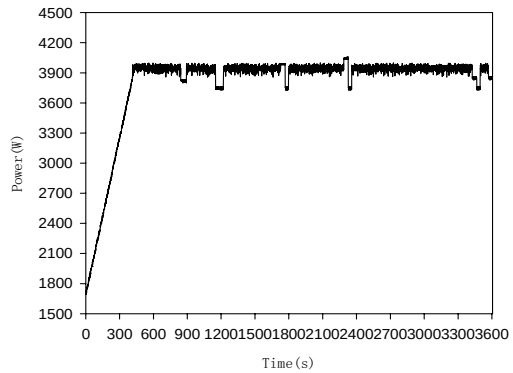


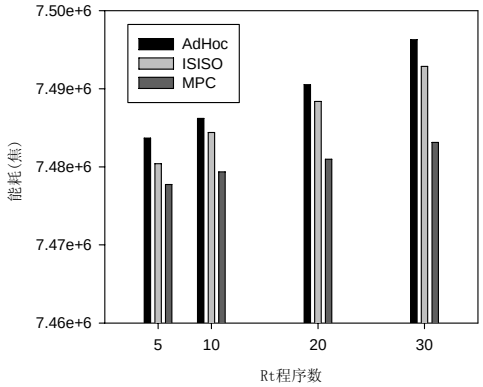
Fig5 Intrusion Detection System under the proposed controller

Plot in Firuge 5 shows a typical running of our propose controlling system to the instrusion detection with the Lincoln Lab data. The start-up time is about 500s and δ keeps in the range from -46W to +57W. Since the variation of the packages in IDS is very complex and the loading time of streams data is unexpected. It leads to appear six serious deviations that are longer duration. This results demonstrate that we should pay much more attention the stability of the proposed control model to the irregular size stream data.

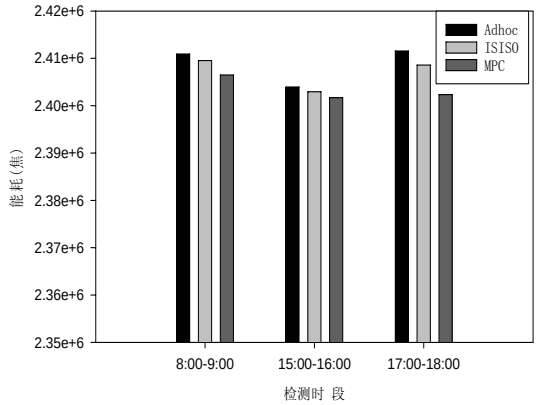
D. The Variation of Workload Analysis

In this section, we investigate the impact of the dynamic workload on the power control models. We use a simply way to simulate the uncertainty of the large-scale streams data workload. This detailed simulation is described as follows: the program R_t is to generate a simulation benchmark with specific randomness

algorithm. And t is the number of programs in the R_t. It can suppose that the number of programs determines the simulation benchmark's uncertainty. For example, if one benchmark program B_1 generated by R_{t1} and $t_1 = 5$, the other B_2 generated by R_{t2} and $t_2 = 5$. So the uncertainty of B_2 is greater than B_1.



(a) simulation data



(b) video quality detection data

Figure 6 Comparison of power under different workload

Figure 6(a) plots the power variation of the three control systems. We can see that the power consumption raises with the increasing uncertainty of the simulation benchmarks. However MPC has minimize increase of power consumption. This results indicate that MPC is very suitable for processing the dynamic workload tasks and may cap the redundancy energy in the cluster level. Figure 6(b) compares the three control systems to process the video quality monitor data in different periods. It includes three periods that are 8:00-9:00, 15:00-16:00 and 17:00-18:00. 17:00-18:00 represents the abnormal period due to the sports game and 8:00-9:00 represents peak period since it is a rush-hour. The power consumption of Ad Hoc and ISISO in peak and abnormal periods are significantly greater than that in normal period. However, the power consumption of MPC in the abnormal period is similar to that in the normal period. This is because that the video streams change dramatically rather than the video stream flow increasing rapidly. So it can prove that MPC is adpated to the unexpected workloads in real system. Aiming at the unexpected workloads the control stability of MPC is better than others.

V CONCLUSIONS

In the processing large-scale streams data tasks field, the power consumption optimization and capping is becoming a challenge for effectively operation a GPU cluster. We consider unexpected workloads and the control system stability and presented an power controller from the cluster perspective to optimize and cap the power consumption. Our controller uses the model predictive control theory. It can adjust and moving optimize the future control input based on the feedback information that is the total energy consumption in the previous control period. Experimental results using simulation data and the engineering data demonstrate that our controller outperforms two state-of-art controllers, Ad Hoc and ISISO, by having better control stability and more accurate power control. Our results also show that our controller is very suitable to the large-scale data streams real processing. In our future work, we plan to analyze setting of the set point and find a more accurately way to determine it. Additionally, we also plan to design the control system to reduce the power consumption of cooperative computing between the CPUs and the GPUs.

ACKNOWLEDGEMENT

We would like to thank the support of National Nature Science Foundation of China (No.60970012), the Innovation Program of Shanghai Science and Technology Commission (No. 09511501000, 09220502800), the program of The Ministry of Education of China(20113120110008), and Shanghai Leading Academic Discipline Project(XTKX2012).

REFERENCES

[1] Thomas Repantis, Xiaohui Gu, Vana Kalogeraki. Qos-Aware Shared Component Composition for Distributed Stream Processing System[J] IEEE Transaction on Parallel and Distributed System, Vol.20, No.7,2010,pp:968-982.

[2] Xiaorui Wang, Ming Chen, Xing Fu. MIMI Power Control for High-Density Servers in an Enclosure[J] IEEE Transaction on Parallel and Distributed System, Vol.21, No.10,2010,pp:1412-1426.

[3] T. Horvath, T. Abdelzaher, K. Shadron, X. Liu. Dynamic voltage scaling in multitier web servers with end-to-end delay control[J] IEEE Transaction on Computers,vol.56(4), 2007,pp:444-458.

[4] L. Bertini, Julius C.B. ,D. Mosse. Power optimization for dynamic configuration in heterogeneous web server clusters[J] The Journal of Sysems and Software, Vol.83(4),2010,pp:585-598.

[5] Xiaorui Wang. Coordinating Power Control and Performance Management for Virtualized Server Clusters[J] IEEE Transaction on Parallel and Distributed System, Vol.22, No.2,2011,pp:245-259.

[6] Lin Yi-Song, Yang Xue-Jun, Tang Tao, Wang Gui-Bin, Xu Xin-Hai. A GPU low-power optimization based on parallelism analysis model. Chinese Journal of Computers, 2011, Vol.34, No.4, pp:705-716.

[7] Zhu Er zhou , Haibing Guan,et al. A translation framework for executing the sequential binary code on

CPU/GPU based architectures[J] Journal of Software, Vol. 6, No. 12,p:2331-2340, 2011.

[8] Haifeng Wang, Qingkui Chen. An Energy Consumption Model for GPU Computing at Instruction Level[J] International Journal of Advancements in Computing Technology. Vol.4, No.2 ,p192-200,2012.

[9] Haifeng Wang, Qingkui Chen. Power Estimating model and Analysis of General Programming on G P U[J] Journal of Software, Vol. 7, No. 5, pp:1164-1170, 2012.

[10] Qinghui Tang, Sandeep Kumar S. Gupta, Georgios V. Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach[J] IEEE Transaction on Parallel and Distributed System, Vol.19, No.11,2008,pp:1458-1472.

[11] Guibin Wang, Xiaoguang Ren. Power-efficient Work Distribution Method for CPU-GPU Heterogeneous System[C] International Symposium on Parallel and Distributed Processing with Applications. 2012.

[12] AMD Display Library, http://developer.amd.com/GPU/ADLSDK/Pages/default.aspx.

[13] Sunpyo Hong, Hyesoon Kim. An Integrated GPU Power and Performance Model[C]. in ISCA'10. 2010.

[14] S. Collange, D.Defour, and A. Tisserand, "Power consumption of gpus from a software perspective," in Proceeding of the 9th International Conference on Computational Science. Berlin, Heidelberg: Springer-Verlag, pp.914-923,2009.

[15] Z. Wang, C. McCarthy, X, Zhu, P. Ranganathan, V. Talwar. Feedback Control Algorithms for Power Management of Servers[C] Proc. Third Int'l Workshop Feedback Control Implementation and Design in Computing Systems and Networks,2008.

[16] Stanimir Mollov, Robert B. etc al. Effective Optimization for Fuzzy Model Predictive Control[J] IEEE Transtractions on Fuzzy Systems. Vol.12, No.5, pp.661-674.2004.

[17] D. Brooks, M. Martonosi. Dynamic Thermal Management for High-Performance Microprocessors[C].In Proc. Seventh Int'l High-Performance Computer Architecture, 2001.

[18] S. Hong , H. Kim. An analytical model for a gpu architecture with memory-level and thread-level parallelism awareness. In ISCA,2009.

[19] Qi Qu,Yong Pei, Modestino,J.W. An Adaptive Motion-Based Unequal Error Protection Approach for Real-Time Video Transport Over Wireless IP Networks[J] IEEE Transaction on Multimedia.Vol.8(5),2006.pp:1033-1044.

[20] Zhangang Wang, Suping Peng ,Tao Liu. GPU accelerated 2-D staggered-grid finite difference seismic modelling [J] Journal of Software, Vol. 6, No. 8,p:1554-1561, 2011.

[21] MIT LINCOLN LAB.2000. DARPA 2000 intrusion detection evaluation datasets[OL] http://ideval.l1.mit.edu/IST/ideval/data/2000/2000_data_index.html.

Qingkui Chen Received the MS degree in computer science from the JILIN university, and PhD degree in University of Shanghai for Science and Technology. He is a full professor of computer science at the University of Shanghai for Science and Technology, China, where he is the head of the Network Computing Group and the Wireless Sensor Network Research Center. His research interests include GPGPU, Network Computing, and WSN. (chenqingkui@gmail.com)

Haifeng Wang He received his diploma in computer science from the Shandong University in 1995, China. He is currently a PHD candidate in the University of Shanghai for Science and Technology. His current research interests include GPU Computing, Network Computing. (gadfly7@126.com).

Bocheng Liu. He is currently a PHD candidate in the University of Shanghai for Science and Technology. His current research interests include GPU Computing(yoyo.liubocheng@gmail.com).