

多指标自趋优的 GPU 集群能耗控制模型

王海峰¹ 陈庆奎²

¹(临沂大学信息学院 山东临沂 276002)

²(上海理工大学光电信息与计算机工程学院 上海 200093)

(wanghaifeng@lyu.edu.cn)

Multi-Indices Self-Approximate Optimal Power Consumption Control Model of GPU Clusters

Wang Haifeng¹ and Chen Qingkui²

¹(Information School, Linyi University, Linyi, Shandong 276002)

²(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093)

Abstract GPU clusters have become important high-performance parallel computing systems in the large-scale stream data field. In practice, the computing requires high computing speed, less power consumption and better reliability. So GPU clusters have three significantly performance indices restraining each others that are computing speed, power consumption and reliability. In real-time computing phase, it needs to dynamically search the optimal point that is the tradeoff among computing speed, power consumption optimization and reliability. So the multi-indices optimization in GPU clusters power consumption control process is a challenging issue. To consider the three indices simultaneously, a comprehensive index is generated by maximum entropy function that can combine them. Then an adaptable control model is built based on model prediction theory that can dynamically scale power consumption status with the workloads variation. This control model can cap the redundant energy consumption and control the power consumption of the GPU clusters under a specific ideal set point while guaranteeing computing speed and reliability. Compared with the control scheme without considering reliability, the results demonstrate that the proposed control scheme has better control stability and robustness and is very suitable to apply into GPU cluster power management projects to handle the real-time large-scale stream data.

Key words power consumption optimization; reliability; GPU clusters; model prediction; maximum entropy function

摘要 在大规模流数据实时处理领域中图形处理器(graphics processing unit, GPU)集群是一种重要的并行计算系统,对计算速度、能耗和可靠性3项指标都有较高要求.然而各指标互相约束,在实时计算中需要动态寻找最优均衡点,因此GPU集群中多项性能指标实时优化成为一个具有挑战性的问题.为综合考虑计算速度、能耗和可靠性3项指标,利用极大熵函数法把多项指标转化为一个综合性能评价指标,再以模型预测控制理论为基础构造一个自适应强的控制模型,该模型能够依据计算负载的变化动态调整集群内节点的能耗状态,在保证计算速度和可靠性的前提下消减冗余计算能耗.与未考虑可靠性的

收稿日期:2013-08-26;修回日期:2014-03-11

基金项目:国家自然科学基金项目(60970012);山东省自然科学基金联合专项项目(ZR2013FL005);山东省自主创新及成果转化专项项目(2014ZZCX02702)

基准控制模型进行对比实验,结果表明所提出的模型具有较好的控制稳定性和鲁棒性,适合应用到 GPU 集群节能管理中.

关键词 能耗优化;可靠性;GPU 集群;模型预测;极大熵函数

中图法分类号 TP302

随着图形处理器(graphics processing unit, GPU)通用计算的发展,GPU 集群逐渐发展为应用于各种行业的并行计算系统.由于 GPU 集群规模和节点性能的增加,对能源的需求不断上升.能耗的增加不仅需要更复杂的散热技术,而且会降低计算可靠性.当能源成为一种重要计算资源,如何有效地实现 GPU 集群节能成为高性能计算的研究热点^[1].从体系结构角度来看,GPU 非常适合处理计算密集型任务,在应用中 GPU 集群常用来处理大规模实时流数据.大规模流数据计算是一种广泛应用的任务模型,在计算过程中不需要存储数据,只要实时获得计算结果,具有极强的时序性^[1].因为 GPU 集群以 24 h×7 的方式进行持续计算,一款额定功率 300 W 的 GPU 年消耗 2 678 kWh 的运行电能和 758 kWh 冷却电能^[2],所以 GPU 集群实时处理大规模流数据的节能控制具有重要意义.

集群能耗控制可使用动态调整电压或频率(dynamic voltage frequency scaling, DVFS)来消减计算冗余能耗^[3-4].最近出现的节能方法是应用控制理论,通过开环或闭环反馈控制 CPU 集群或者 Web 服务器集群的能耗^[2,5-6].另外,对于计算任务的节能研究主要关注并行实时任务的节能调度策略和算法^[7-9].较少有学者注意集群节能控制与特定计算任务相结合的研究,并且几乎忽视了节能控制对计算可靠性的影响.当处理器降低电压或核心频率时会导致瞬间故障率提高,引起计算可靠性的降低^[10-11].大规模流数据实时计算是一种应用广泛、典型的持续计算任务,不仅要求在规定时限内完成任务,而且对结果的准确性有严格要求.因此大规模流数据实时计算对 GPU 集群的计算速度、可靠性和能耗都有严格限制,合理解决这些约束成为 GPU 集群功耗优化控制走向生产环境的关键.

针对大规模流数据计算的特点,本文提出一种考虑计算速度、能耗、可靠性 3 种指标的自适应优化控制模型,可依据计算负载的变化实现自动趋优控制.通过最大熵函数把计算速度、能耗和可靠性组合成一个综合性能指标,再用模型预测策略建立能耗控制模型来管理 GPU 集群节点的能量配给,将综

合性能指标控制在理想设定值范围内,在保证计算速度和可靠性前提下实时消减冗余能耗.

1 问题描述

在大规模流数据实时计算中,优化 GPU 集群的计算能耗要考虑以下两个难题:

1) 计算负载变化的不确定性.这将导致 GPU 集群出现 2 种现象:①某些节点提前完成任务后处于等待状态,因此产生冗余时间;②节点存在冗余能耗.因为节点始终以较高能耗状态运行,而不是处于最优能耗状态,所以在持续计算中一直存在冗余能耗.

在传统 CPU 集群能耗优化研究中,由于节点存在冗余时间,因此适度降低处理器的核心频率,用改变能耗状态的方式来消减冗余能耗,使能耗优化控制成为可能^[6].然而 CPU 集群能耗优化的研究成果很难直接应用到 GPU 集群中,因为 GPU 集群在能耗方面具有一定特殊性^[6],这种特殊性由 2 个因素造成:①GPU 集群节点内的异构性. GPU 集群节点由负责数据通信管理的 CPU 和数据计算的 GPU 构成,这种异构性增加了计算能耗的复杂性.②任务模型的复杂性.在 GPU 通用计算中 CPU 负责收集、整理并传输数据到 GPU 显存中,GPU 提取数据进行计算,CPU 再回收结果.大规模流数据计算中包含各种计算任务,每种任务具有不同的通信密度和计算密度^[2],而且通信和计算指令的能耗不同,因此任务模型的复杂性也影响 GPU 集群能耗.总之,为了克服计算负载变化不确定性对能耗优化的影响,必须以动态的、自适应的方式来控制计算能耗的优化.

2) 计算速度、能耗和可靠性的均衡.在执行时限内通过降低 GPU 核心频率来改变能耗状态,并达到节能目标.然而,DVFS 机制不仅牺牲计算速度,而且对计算可靠性还有一定的副作用,会引起暂时性故障率的升高.因此,在能耗优化中要综合考虑计算速度、能耗和可靠性之间的关系,寻找最优或次优的均衡解.上述难题可归纳为一个问题,为实现计算能耗的优化,在 GPU 集群持续计算过程中动态

寻找计算速度、能耗和可靠性的最优均衡解的控制问题。

2 性能指标

为考虑计算速度、能耗和可靠性之间的均衡关系,先用系统辨识法建立三者的量化模型.系统辨识法以实验数据为基础,依靠统计法来确定动态系统的模型结构或参数的一项成熟技术.本文只考虑 GPU 核心频率对计算速度、能耗和可靠性的影响,不考虑频率切换引起的时间和功耗损失。

2.1 计算速度指标

大规模流数据实时处理对计算速度有限制,在规定时间内必须完成计算任务.由于负载变化的复杂性,导致集群中某些节点提前完成计算,因此集群节点中存在一定的弹性计算时间。

在此用基准程序的执行时间来量化计算速度,研究显示计算任务的执行时间与处理器频率存在较强线性关系^[2,7-8],而外存访问和输入输出的性能损失是影响其线性关系的主要因素.为简化性能指标的复杂性,忽略 GPU 与主机内存通信时的性能损失(实验基准程序选择 CUDA SDK 中的 Demo 程序集),使用系统辨识法来确定计算速度与核心频率两者之间的关系为

$$S(f) = \alpha \left(\frac{f - f_{\min}}{f_{\max} - f_{\min}} \right), \quad (1)$$

式(1)中, S 表示 GPU 节点的计算速度, f 是当前 GPU 的核心频率, $\{f = f_i \mid f_{\min} < f_i < f_{\max}\}$, $f_{\max}$ 和 $f_{\min}$ 分别是 GPU 核心频率的最大和最小值, α 是与 GPU 体系相关的常量因子.下面重点讨论体系因子 α ,以 NVIDIA 公司的 GPU 为例,现存 GT80, GT200, Fermi 和 Kepler 4 种体系.由于硬件体系结构的优化,不同体系的 GPU 以相同频率运行时计算速度存在差别,越先进的体系结构计算速度越快.比如 Kepler 的节点以 1 000 MHz 运行时速度高于同频率的 Fermi 节点.体系因子 α 可解决体系差异造成的量化偏差,通过最小二乘参数辨识法确定 4 种体系结构的因子为 $\{\alpha_{\text{GT80}} = 0.75, \alpha_{\text{GT200}} = 0.92, \alpha_{\text{Fermi}} = 1.27, \alpha_{\text{Kepler}} = 1.56\}$ 。

该模型表明提高 GPU 核心频率可线性增加计算速度,计算速度 S 经归一化处理为无量纲变量,其取值范围为 $(0, 1)$,取值越大则计算速度越快.经拟合优度检验,该线性模型的可决系数 $R^2 > 96\%$ 。

2.2 能耗指标

当以不同频率运行时处理器的计算能耗不同,因此仍然考虑能耗与 GPU 核心频率之间的关系.计算能耗是处理器计算功率和执行时间的乘积,在已知任务执行时间的情况下功率可反映能耗变化,在此利用计算功率来量化能耗指标.研究表明处理器核心频率与计算功率呈近似的线性关系^[2],如式(2)所示:

$$P(f) = \frac{p_{\max} - (\beta f + b)}{p_{\max} - p_{\min}}. \quad (2)$$

以现存研究的能耗模型为基础,再用系统辨识法确定 GPU 的模型参数.式(2)中 P 表示计算功率, f 表示当前运行时 GPU 核心频率,参数 β 是 GPU 体系因子,不同于 2.1 节的体系因子,每代 GPU 体系改良时均考虑能耗优化,新体系能耗要优于旧体系,因此体系因子为 $\{\beta_{\text{GT80}} = 1.25, \beta_{\text{GT200}} = 1.12, \beta_{\text{Fermi}} = 0.95, \beta_{\text{Kepler}} = 0.86\}$; b 是调节参数,用来量化同一体系结构中不同类型 GPU 的功率差距^[12]; $p_{\max}$ 和 $p_{\min}$ 分别表示该 GPU 最大和最小计算功率。

能耗模型表明,随着 GPU 核心频率的提高计算功率呈线性增加.经过式(2)的归一化处理,能耗指标 P 为无量纲变量,取值范围为 $(0, 1)$,取值越大则能耗越小.经拟合优度检验,该线性模型的可决系数大于 93%,再用随机白噪声校验模型的准确性,发现模型输出值与实际测量值的相对误差为 0.97%。

2.3 可靠性指标

在实时计算中暂时性故障是影响 GPU 集群可靠性的主要因素,比如硬件访存故障、软件错误等.现有研究认为处理器的暂时性故障与关键电路的电流呈指数关系^[13],而关键电路的电流与电压有关,因此 DVFS 机制会影响 GPU 关键电路的电流,降低核心频率,导致暂时性故障率提高.在此重点考虑暂时性故障率对计算可靠性的影响,用任务失效率来度量计算可靠性,建立可靠性与 GPU 核心频率的关系式.先定义频率的规范算子:

$$R = f / f_{\max}, \quad (3)$$

式(3)中, R 表示运行时频率与最大频率的比值.为适应集群中的异构环境,用规范算子归一化处理各种 GPU 频率值.任务失效率 L 是失效任务数与总任务数的比值^[11],如式(4)所示:

$$L[R(f)] = L_0 \times 10^{\gamma(1-R)/1-R_{\min}}, \quad (4)$$

式(4)中, L_0 为当 GPU 以最大频率运行时的任务失效率, γ 为体系因子, $R_{\min}$ 为 $f_{\min}/f_{\max}$.虽然文献[11]

提供处理器任务失效率模型,但是没有介绍确定参数 L_0 和 γ 的方法.

采用统计法拟合任务失效率的参数. 首先选择 H. 264 视频解压缩作为基准程序. 因为 H. 264 解压缩中存在离散余弦变换、运动补偿等数据密集型算法,不仅包含适合 GPU 并行处理的典型运算,而且对准确性非常敏感^[14],即使一帧图像中出现一个像素的错误也会导致这帧解压缩失败. 另外用图像是否出现局部损伤来判断解压缩任务的失败,便于人工或软件作统计判断^[15]. 具体过程如下:1)选取 10 条 GPU 解压时间超过 3 000 s 的 H. 264 视频流,再解压缩成 yuv 格式的视频. 把每个视频流视为一个计算任务集,而截取其中一段视频流的解压缩为一个计算任务. 若某段视频流中出现损伤帧,则认为出现一个失效任务. 例如取解压缩时间为 3 000 s 视频流,以 60 s 为单元划分成 50 个任务. 若每个 60 s 内出现一个或多个损伤帧,则认为出现一个失效任务. 2)用软件 yuvPlayer 播放 yuv 视频,在播放中自动统计出损失帧数,以此计算失效任务数.

以 NVIDIA 公司两款典型的显卡 GTX280 和 GTX295 为例,用不同核心频率进行实验. 表 1 显示,随着核心频率的降低失效任务数逐渐增多. 拟合任务失效率的参数 L_0 和 γ ,两款 GPU 任务失效率分别为 L^1, L^2 ,如式(5)(6)所示:

$$L^1(f)=0.047\times10^{1.2(1-R)/1-R_{\min}},\tag{5}$$

$$L^2(f)=0.053\times10^{1.31(1-R)/1-R_{\min}}.\tag{6}$$

Table 1 Average Failure Task Number with Different Frequences

表 1 在不同计算频率下的平均失效任务数

GPU/MHz	GTX280	GTX295
600	37.1	43.2
700	21.3	23.6
800	12.3	12.9
900	7.1	7.1
1 000	4.1	3.9
1 100	2.2	2.1
Total Tasks	50	40

其他类型 GPU 也可用该方法确定参数,最后用式(7)规范化单个 GPU 节点的任务失效率,并作为量化计算可靠性的指标:

$$L(f)=\frac{L_0\times(1-10^{\gamma(1-R)/1-R_{\min}})}{L_0-L_{\min}}.\tag{7}$$

可靠性指标归一化到(0,1)内,值越高则计算可

靠性越好. 最后采用基于残差的非线性拟合优度检验任务失效率模型,其可决系数大于 87%^[16].

3 极大熵函数法

极大熵函数法可综合考虑 3 种性能指标,将 3 种指标转化成一个综合性能指标,具有良好的最优解辨识度.

3.1 多目标优化

由于计算速度、能耗和可靠性是相互约束的 3 种性能指标,因此可用多目标优化方法来处理三者的均衡关系. 多目标优化主要解决两个或两个以上存在相互制约关系的目标优化问题,基本形式如下^[17]:

$$\begin{cases} \min y=F(x)=(f_1(x),f_2(x),\cdots,f_m(x))^T \\ \text{s. t. } g_i(x)\leqslant 0, i=1,2,\cdots,q, \\ h_j(x)=0, j=1,2,\cdots,p. \end{cases}\tag{8}$$

式(8)描述了一个具有 n 个决策变量、 m 个目标变量的多目标优化问题, $\mathbf{x}=(x_1,x_2,\cdots,x_n)\in\mathbf{X}\subset\mathbf{R}^n$ 为 n 维的决策向量, $\mathbf{X}$ 为 n 维的决策空间, $\mathbf{y}=(y_1,y_2,\cdots,y_m)\in\mathbf{Y}\subset\mathbf{R}^m$ 为 m 维的目标向量, $\mathbf{Y}$ 为 m 维的目标空间. 目标函数 $F(\mathbf{x})$ 定义 m 个由决策空间向目标空间的映射函数. 先介绍两个基本概念.

定义 1. 可行解. 对于某 $\mathbf{x}\in\mathbf{X}$,若满足式(8)中的约束条件,则 $\mathbf{x}$ 为可行解. 由 $\mathbf{X}$ 中的所有可行解组成的集合称为可行解集合,记 $\mathbf{X}_h$,且 $\mathbf{X}_h\in\mathbf{X}$.

定义 2. Pareto 占优解. 若 $\mathbf{x}_a, \mathbf{x}_b\in\mathbf{X}_h$ 是式(8)中的 2 个可行解,且满足如下条件: $\forall i=1,2,\cdots,m, f_i(\mathbf{x}_a)\leqslant f_i(\mathbf{x}_b) \wedge \exists j=1,2,\cdots,m, f_j(\mathbf{x}_a)<f_j(\mathbf{x}_b)$, 则称 $\mathbf{x}_b$ 比 $\mathbf{x}_a$ Pareto 占优,记作 $\mathbf{x}_a>\mathbf{x}_b$.

3.2 多目标转化单目标

在处理多目标优化问题时,通常要把多个目标函数转化成单个目标函数来处理. 最典型、直观的方法是线性权重组合法. 为各目标函数设置权值,再用线性求和转成单一目标. 若多目标函数 $F(\mathbf{X})=(f_1(\mathbf{x}),f_2(\mathbf{x}),\cdots,f_n(\mathbf{x}))^T$, 为 $f_i(\mathbf{x})$ 设定权值 λ_i , 则转化后的单目标函数为 $\sum_{i=1}^n \lambda_i f_i(\mathbf{x})$.

但是该方法对 Pareto 占优解的辨识性较差. 例如,令 $n=2$,单目标函数为 $\sum_{i=1}^2 \lambda_i v_i, \lambda_1=0.8, \lambda_2=0.2$,且存在两个可行解 $\mathbf{v}_1=(2,100)^T, \mathbf{v}_2=(20,28)^T$,将 $\mathbf{v}_1$ 和 $\mathbf{v}_2$ 代入目标函数后结果均为 21.6,无法辨别 $\mathbf{v}_1$ 和 $\mathbf{v}_2$ 的优劣. 由于权重的分配 λ_1 远大于 λ_2 ,说明第 1 个目标函数重要,因此 $\mathbf{v}_2=(20,28)^T$

比 $v_1 = (2, 100)^T$ 更符合实际要求。

3.3 极大熵函数

为提高 Pareto 占优解的辨识精度,引入极大熵函数.极大熵函数法能取多个目标中最小的目标函数并进行最大化处理,以此保证每个目标函数均达到最大化^[18].

定义 3. 以下函数为在 $x \in \Omega \subset \mathbb{R}$ 上的极大熵函数:

$$Q_p(F(x)) = 1/p \ln \left(\sum_{i=1}^m e^{p f_i(x)} \right), \quad (9)$$

$$\begin{cases} \min Q_p(F(x)), \\ \text{s. t. } G(x) \leq 0, \end{cases} \quad (10)$$

其中,目标函数 $F(X) = (f_1(x), f_2(x), \dots, f_k(x))^T$, 约束函数 $G(X) = (g_1(x), g_2(x), \dots, g_m(x))^T$. 在式(9)(10)中 p 是一个重要的参数,因为当 p 趋向无穷时 $Q_p(x)$ 可近似代替目标函数 $F(X)$, 所以 p 的选择至关重要。

定理 1. 对于任何 $x \in \Omega \subset \mathbb{R}$,极大熵函数 $Q_p(x)$ 随着参数 p 单调递减,并且当 $p \rightarrow \infty$ 时以 $F(x)$ 为极限^[19],即

$$\begin{aligned} \forall p_1, p_2, p_1 \leq p_2, Q_{p_1}(x) &\geq Q_{p_2}(x), \\ \lim_{p \rightarrow \infty} Q_p(x) &= F(x). \end{aligned} \quad (11)$$

由定理 1 表明,若参数 p 选值较大时,目标函数 $F(X)$ 可由极大熵函数 $Q_p(x)$ 近似替代.虽然参数 p 取有限值时仅能得到问题的近似解,但是只要取较大数就能保证较高精度^[18].因此设 $p = 10^5$.再以 3.2 节的实例来验证辨识效果,设极大熵函数为 $Q(v) = 1/10^5 \ln \left(\sum_{i=1}^2 e^{10^5 \lambda_i v_i} \right)$,将可行解 v_1 和 v_2 代入后,可得出 $Q(v_1) > Q(v_2)$,说明可行解 v_2 优于 v_1 .

3.4 综合性能指标

在 GPU 集群能耗控制中,式(1)(2)(7)分别表示计算速度、能耗和可靠性 3 个性能指标,并作为 3 个目标函数,则 $F(X) = [S(x), P(x), L(x)]^T$,其中自变量 x 为 GPU 核心频率.可用以下极大熵函数来描述综合性能指标:

$$Q_p(F(x)) = 1/p \ln \left(\sum_{i=1}^3 e^{p f_i(x)} \right), \quad (12)$$

其中, $f_i(x)$ 是 $F(X)$ 的分量,该函数构造出一个综合性能指标,简记作 q .若在控制时刻 t_i 和 t_{i+1} ,某节点的频率分别为 f_1, f_2 ,代入式(12)得 $q_i > q_{i+1}$,可知该节点以频率 f_2 运行要优于以 f_1 运行,因此考虑将频率由 f_1 调整到 f_2 ,取得更好的综合性能.在第 4 节将用综合性能指标作为控制变量,并建立能耗控制模型。

4 多指标自趋优控制

在大规模流数据的实时处理中,计算负载的变化存在不确定性.在 GPU 集群节能管理中需要针对这种不确定性建立实时控制机制,实现计算速度、能耗优化、可靠性 3 项指标的自适应寻优。

为设计 3 项性能指标的自趋优控制模型(multi-indices self-approximate optimal control, MSOC),采用经典过程控制中的模型预测理论.模型预测控制(model predictive control, MPC)是构造系统模型来预测未来时域的控制输出,并将控制输出限定在理想范围内,面向不同应用场景演化出各种实时控制算法^[20].如图 1 所示, $y(k)$ 为控制时刻 k 的实际输出, $r(k+i)$ 为理想设定值, $u(k+i)$ 为控制输入

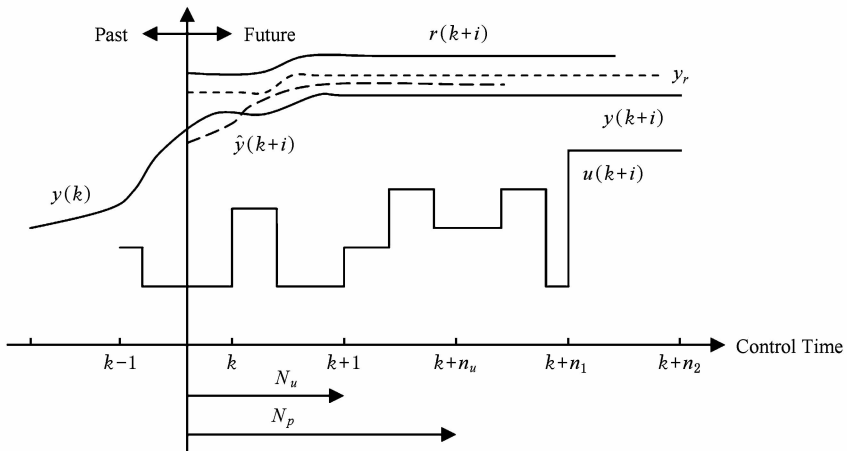


Fig. 1 Model prediction control diagram.

图 1 模型预测控制示意图

律, $\hat{y}(k+i)$ 为模型在时刻 $k+i$ 的预测输出, n_u 为控制时域, n_p 为预测时域, y_r 是参考轨迹用来引导控制输出, 防止受噪声干扰产生振荡. 在时刻 k 预测输出 $\hat{y}(k)$ 与实际输出 $y(k)$ 的误差作为反馈信号, 调整下一个预测周期的预测输出 $\hat{y}(k+1)$, 以此形成整个控制过程中的滚动优化^[21].

4.1 系统模型

系统模型是操作变量与被控变量的关系模型. 在 GPU 集群节能控制中, 通过调整核心频率能改变计算速度和能耗状态, 而且也影响计算可靠性, 因此核心频率为操作变量, 被控变量是综合性能指标变量, 如式(12)所示. 由于该控制系统是典型离散事件系统, 因此以节点为单位建立操作变量与被控变量的差分方程, 基本符号如下: T 表示一个控制周期的时间长度, k 表示第 k 控制周期, $f_i(k)$ 表示集群第 i 节点的核心频率, $q_i(k)$ 表示第 i 节点的综合性能指标.

由 3.4 节的式(12)可得集群第 i 节点在第 k 控制周期的模型函数:

$$q_i(k) = 1/p \ln\{p(S[f(k)] + P[f(k)] + L[f(k)])\}, \quad (13)$$

其中, S, P, L 分别表示计算速度、能耗和可靠性指标. 第 i 节点的计算频率调整后, 转入第 $k+1$ 个控制周期, 则在 $k+1$ 控制周期的差分方程为式(14)所示:

$$q_i(k+1) = q_i(k) + d_i(k) + a_i d_i(k) + L_0 \times (1 - 10^{d_i(k)}), \quad (14)$$

其中, $d_i(k) = f_i(k+1) - f_i(k)$. 设集群有 N 个节点, 则受控变量为集群各节点综合性能指标之和 TQ , 如式(15)所示:

$$TQ(k) = \sum_{i=1}^N q_i(k). \quad (15)$$

而系统模型中的操作变量为各节点核心频率 $F = [f_1, f_2, \dots, f_n]$.

4.2 控制系统

控制系统由节点状态监测、能耗监测和控制 3 个模块组成, 如图 2 所示. 状态监测模块(status monitor, SM)以进程形式分布在集群各节点中, 在软件 GPU-Z 的辅助下收集 GPU 节点的计算频率、GPU 利用率、风扇转速等状态信息, 在状态信息中只有计算频率为计算速度指标提供依据, 其他状态信息只了解节点健康状况; 能耗监测模块负责测量各节点计算功率; 控制模块中分为控制器和执行器, 控制器接受状态监测和能耗监测模块提供的综

合指标数据, 并形成受控变量, 对未来控制输出进行预测; 执行器根据控制输出调整下一个控制周期中各节点的核心计算频率. 图 2 中符号 S 表示节点状态信息, P 表示计算功率, $F = [f_1, f_2, \dots, f_n]$ 表示在新一轮控制周期中各节点计算频率.

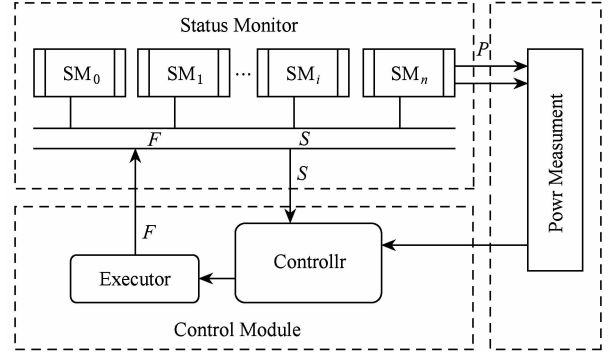


Fig. 2 Power consumption control system of GPU clusters.

图 2 GPU 集群能耗控制系统

控制模块的核心—控制器的设计: 控制器包括系统模型、代价函数、参考轨迹和最小二乘法运算器. 系统模型主要由辨识法建立, 如式(15)所示; 代价函数本属于经济学术语, 用代价的方式量化各种控制输出与设定值的距离, 作为判断控制输出的量化依据, 如式(16)所示:

$$V(k) = \sum_{i=1}^p \|CQ(k+i|k) - ref(k+i|k)\|_{w(i)}^2 + \sum_{i=0}^{M-1} \|d(k+1|k) + f(k+i|k) - F_{\max}\|_{R(i)}, \quad (16)$$

式(16)中, $V(k)$ 为第 k 个控制周期时的代价函数, P 为预测周期, M 为控制周期, $W(i)$ 为误差权矩阵, $R(i)$ 为惩罚权矩阵. 式(16)的第 1 部分是预测值与参考值的差. 目标是将输出控制在理想设定值范围内, 而参考轨迹用来引导预测控制方向, 防止控制器出现剧烈振荡. 在此由式(17)的参考轨迹函数产生未来输出的参考值^[2]:

$$ref(k+i|k) = Q_s - e^{-\frac{qT}{T_{ref}}} (Q_s - tq(k)). \quad (17)$$

式(17)产生如图 1 中短虚线 y_r 所示的轨迹, 引导控制输出, 并限定在理想设定值 Q_s 范围内. T_{ref} 是系统响应时间常量, 影响系统收敛速度. 若响应时间常数值越小, 则系统向 Q_s 的逼近速度越快. $W(i)$ 是误差权矩阵, 用来区分集群各节点的误差对整个代价函数值的影响程度. 式(16)的第 2 部分是控制惩罚, 其中惩罚权矩阵 $R(i)$ 防止系统超过限定范围或发生剧烈振荡. 由于 GPU 集群能耗控制本质是一个多输入多输出的优化问题, 在满足约束条件

的前提下尽量减小代价函数值,因此该问题可被转化成一个标准的带约束最小二乘问题,通过最小二乘法求解器来确定控制模型中的参数.算法复杂度与集群节点数是多项式关系,在线计算量适中.

最后是控制器的工作机制:1)测量时刻 k 的 GPU 集群状态 $x(k)$,为综合性能指标值;2)根据系统模型和上一时刻的反馈信息进行时刻 $k+1$ 的预测;3)令 $x(k|k)$,并求解代价函数的优化问题,得出时刻 k 的控制输出;4)对系统施加时刻 k 的控制作用,即控制律 $u(k)=u(k|k)$,为下一时刻提供反馈信息;5)令 $k=k+1$,返回 1).总之,控制器根据系统模型预测未来控制行为,选择满足约束条件且代价函数最小的输入轨迹,该控制输入轨迹是 GPU 核心频率,初始阶段为 $\Delta f(k)$,经过控制时域的反复迭代后为 $\Delta f(k), \Delta f(k+1|k), \dots, \Delta f(k+M-1|k)$.

5 实验分析

5.1 实验方案设计

1) GPU 计算功耗测量. GPU 显卡由主板 PCI 和外接电源供电两部分构成,其中 PCI 供电约占共功耗的 10%,而且必须采用侵入式测量方式,因此国内外 GPU 功耗测量研究中暂时忽略 PCI 供电,仅考虑外接电源产生的功耗^[22-24].实验采用自行设计的电流功耗采集卡来测量外接电源能耗^[12],并用软件记录功率值^①.因为功耗采集卡测量到的是实时功率值,为了方便数据处理,所以用单位时间内平均采样功率作为测量值.

2) GPU 节点能耗状态改变方式.现有硬件调整计算频率^[22]、操作系统级频率调整接口^②和应用软件调整核心频率 3 种方式^[23],然而 3 种方法中都不存在能耗状态切换延迟的问题,这里暂时不考虑切换引起的性能损失,为消除这种延迟,采用下述仿真方法:假设计算任务 W 的计算时间为 T ,根据控制周期将 T 划分为连续时间段序列 $\{t_1, t_2, t_3, \dots, t_k\}$.若能耗状态集 $M=\{m_1, m_2, m_3, \dots, m_l\}$,用每种能耗状态 m_i 来计算 R ,这样一个计算任务被执行 l 次.统计每次执行时功耗变化规律 $P^{m_i}=\{p_1^{m_i}, p_2^{m_i}, \dots, p_k^{m_i}\}$, P^{m_i} 表示能耗状态为 m_i 时的测量功率序列,其中 $p_i^{m_i}$ 表示在执行 m_i 方案 t_i 时的功率值.在计算 R 过程中,能耗控制系统需要实时调整能耗状态,在不同时间段使用不同能耗状态.例如,设任务 W 的控

制输出轨迹为 $W_m=\{m_3, m_5, m_l, \dots, m_8\}$, R_m 表示在 t_1 阶段内的能耗状态为 m_3 , t_2 时调整为 m_5 , t_3 时调整为 $m_l, \dots, t_k$ 时调整为 m_8 .因此计算 W 的实际功率轨迹为 $P_r=\{p_1^{m_3}, p_2^{m_5}, p_3^{m_l}, \dots, p_k^{m_8}\}$.可见只要获得控制系统的理论输出值,该仿真法就能得出较准确的实验数据.实验中的数据记录、分析及汇总程序使用 Python 语言实现.

选择只考虑计算速度和能耗优化的控制模型作为基准方法,进行控制稳定性和鲁棒性的对比.基准控制模型以模型预测控制 MPC 为核心,与所提出控制模型的主要区别是系统模型不同,基准控制模型仅考虑计算速度和能耗之间的均衡,简记为 MPC.另外实验中 GPU 集群实验床分别为 4, 8, 16 个节点,计算任务为大规模 H. 264 视频流解压缩计算.由于本研究仅关注节能控制和计算可靠性,为降低实验复杂性,人工为每个节点分配任务,各节点保持持续计算的状态,不考虑负载均衡等因素.

5.2 控制稳定性

控制稳定性的实验思路如下:在处理相同计算负载的情况下,比较两种控制模型中集群节点调整能耗状态次数的分布,即调整核心频率次数的分布.在趋向理想设定点的过程中节点调整能耗状态越频繁,则说明控制稳定性越差.采用 4 个节点的 GPU 集群实验床,系统控制周期设为 10 s.

图 3 列出连续 100 个控制周期内两个控制模型的能耗状态调整情况, x 轴表示控制周期, y 轴表示节点能耗状态调整的次数 $[0, 4]$, 0 表示该控制周期

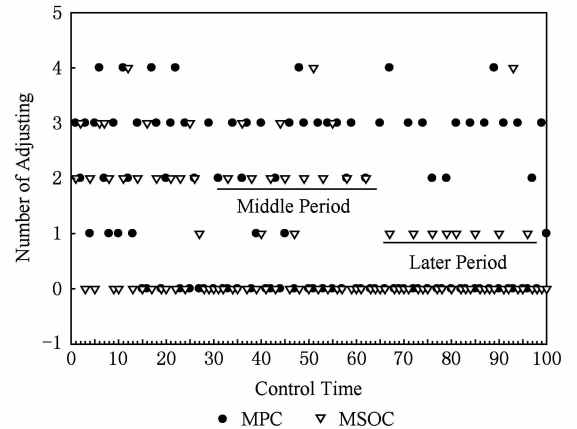


Fig. 3 Distribution of adjusting number of two control models.

图 3 两种控制模型能耗状态调整次数分布

① http://v.youku.com/v_show/id_XNDY0NDAYMzQ0.html

② <http://developer.amd.com/GPU/ADLSDK/Pages/default.aspx>

中没有节点调整能耗状态,4 表示该控制周期中全部节点都发生了能耗状态调整.例如,坐标[5,0]处的空心三角和[5,1]处的实心圆分别表示在控制时刻 t_5 MSOC 中无节点调整能耗状态,而 MPC 有 1 个节点调整能耗状态.由图 3 可见 MSOC 比 MPC 的调整次数少,而且均匀地分布在整个控制时域中. MPC 中 3 个节点(占集群 75%)同时调整能耗状态的次数比较多,呈现均匀分布,这说明 MPC 中能耗状态切换较频繁,必然引起可靠性和能耗的损失.在计算初期 MSOC 中节点状态改变频繁,到中期后多发生 2 个节点调整能耗状态,在后期则均匀地出现 1 个节点能耗调整的现象,这些实验现象表明 MSOC 比较稳定地调整系统能耗.

表 2 统计了能耗状态调整次数,上述分析显示 MSOC 具有更好的控制稳定性,可防止集群出现能耗状态的抖动,降低状态切换造成的成本与可靠性损失.

Table 2 Statistics of Adjusting Number of Power Consumption Status with 4 Nodes

表 2 4 节点集群中能耗状态调整次数统计

Control Model	0	1	2	3	4
MPC	47	7	15	24	7
MSOC	61	12	17	7	3

由表 3 可见,MPC 与 MSOC 相比,平均功率稍低,计算时间略短,而计算可靠性明显低于 MSOC.这说明在计算速度和能耗方面 MPC 稍微优于 MSOC,但是在提高计算可靠性 L 上 MSOC 具有出色的表现,综合性能指标 q 优于 MPC.

Table 3 Comparison of Two Control Models

表 3 两种控制模型比较

Control Model	Run Time/s	Power/W	L	q
MPC	1 886.7	251.4	0.713	0.63
MSOC	1 897.3	256.3	0.876	0.69

5.3 控制鲁棒性

由于大规模流数据计算负载的变化具有不确定性,因此用负载变化的不确定性来验证控制模型的鲁棒性.若能适应变化不确定性越大的计算负载,则表明控制鲁棒性越好.

实验中采用改变仿真子程序数的方式来模拟不确定性的变化.因为一个仿真子程序对应处理一种计算负载类型的子视频流,而仿真视频流由各种类型的子视频流随机组合生成,所以参与组合的子视频流数量越多,则计算负载变化的不确定性越大.具体方法:1)选择不同分辨率的视频流 $V=\{V_1, V_2, \dots, V_t\}$,设计处理视频流 V 的程序集 $O=\{o_1, o_2, \dots, o_t\}$,其中 o_i 为子程序, $|O|$ 表示子程序数量;2)随机组合视频流并循环发送来模拟实时视频流,用相应的 o_i 进行计算.该仿真方法中程序集 O 的大小决定仿真视频流变化的不确定性,参与的子程序越多,则仿真视频流变化的不确定性越大.例如,若由程序集 O^1 与 O^2 分别构造仿真视频流 V^1 和 V^2 , $|O^1|=5, |O^2|=10$,则 V^2 计算负载的变化比 V^1 更剧烈,即 V^2 负载变化的不确定性大于 V^1 .

图 4 提供 MPC 与 MSOC 的综合性能对比.图中 x 轴表示程序集 R 的大小,沿着 x 轴正方向负载变化的不确定性逐渐增大; y 轴是节点计算负载量的饱和程度,分为[30%,60%,80%,100%]四种程

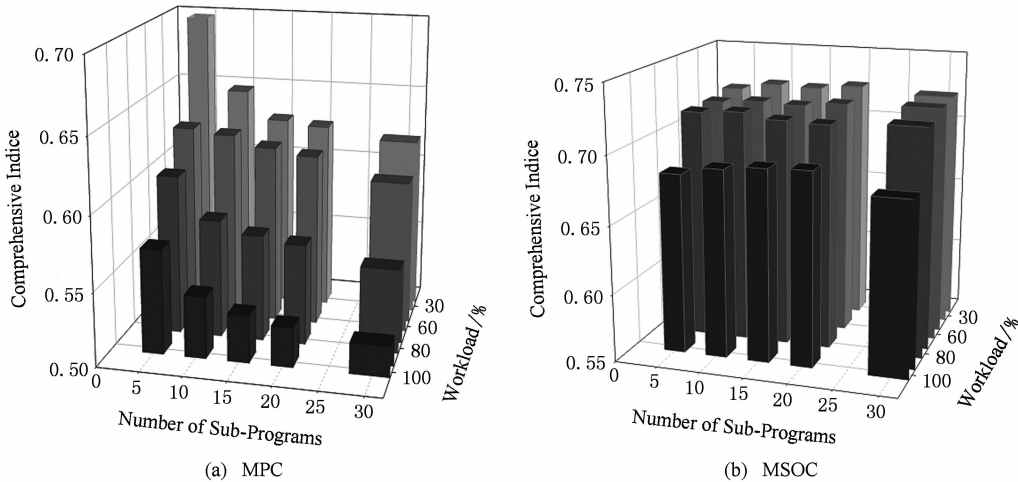


Fig. 4 Comparison of two control models with 4 nodes.

图 4 4 节点集群中两种控制模型比较

度(在此以流多处理器利用率来度量计算负载饱和度,使用 Parallel Nsight 监控该指标);z 轴为综合性能指标,取值范围在 $[0,1]$ 区间内.图 4 显示,随着计算负载变化不确定的增加,MPC 的综合性能指标逐渐下降;随着节点计算负载的增加,综合性能指标也逐渐下降.由此说明 MPC 没有充分考虑计算可靠性,而负载变化的不确定性也影响可靠性,因此导致综合性能指标下降.如图 4 所示,MSOC 的综合性能指标几乎不受计算负载不确定性的影响,始终保持在较稳定的范围内,只是随着负载的增加综合性能指标才稍有下降.以上说明 MSOC 充分考虑计

算速度、能耗和可靠性,在处理负载变化不确定性的计算任务时具有较好鲁棒性,适合实时处理大规模流数据计算的能耗控制.

接下来考虑 GPU 集群规模对控制系统的鲁棒性影响,将集群计算负载的压力控制在近饱和状态(80%),并分别以节点数 4,8,16 这 3 种规模进行实验,仿真数据流中程序集 R 的大小为 $\{5,10,15,20,30\}$.如图 5 所示,随着集群规模的增加,MPC 的综合指标逐渐降低,而 MSOC 的综合指标比较稳定.这表明 MSOC 控制模型有较好的线性可扩展性,并且能够适应较大规模的 GPU 集群控制.

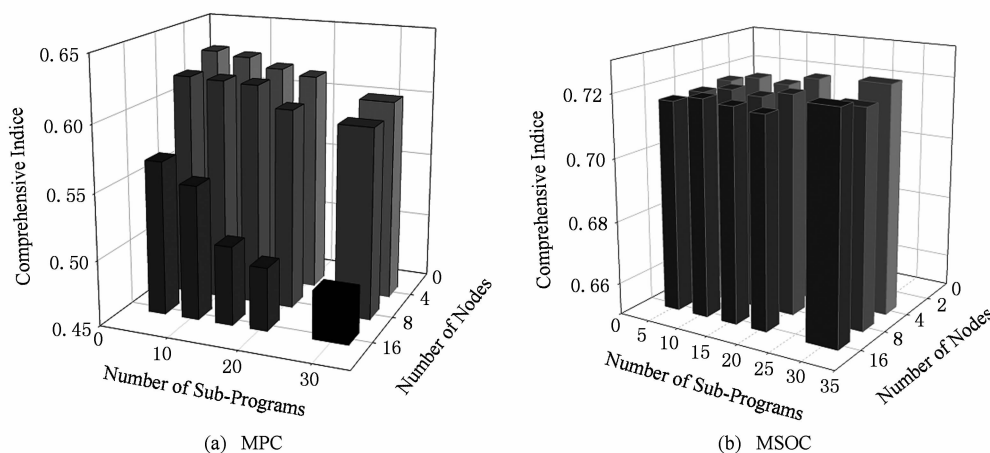


Fig. 5 Comparison of two control models with different cluster sizes.

图 5 不同规模集群中两种控制模型的对比

6 相关研究

GPU 能耗优化的研究集中在单节点的能耗优化,建立合理的能耗预测或优化模型实现各种计算任务的节能.分析各种计算任务的能耗规律,构建能耗预测模型,为能耗优化控制提供基础支持^[12];从软件计算角度量化 GPU 计算能耗,建立指令级功耗分析模型,为应用程序设计节能优化策略^[24];综合考虑 GPU 与 CPU 异构特点,以访存和计算并行度建立功耗分析模型,并为应用程序的运行配置提出优化策略^[25].总之,现有研究仅考虑单 GPU 对某类计算任务模型的节能优化,尚未考虑 GPU 集群的节能优化.

本文关注 GPU 集群计算的可靠性,但是在此方面的研究尚处在起步阶段.国外关注单节点访存中的暂时性故障,建立量化模型并研究对计算可靠性的影响^[26];利用冗余计算来保证 GPU 计算可靠性,

分析各类冗余策略的性能代价^[27];Ameya 探索冗余计算的容错代价和量化方法,提出优化容错代价的措施^①;在时间双模冗余策略的基础上,国内徐新海等人增加了 Lazy 策略,通过减少不可靠数据集降低容错代价^[28];硬件冗余计算也是一种优化容错性能损失的方法,具有透明度高的优点和设计复杂的缺点^[29].涉及 GPU 集群计算可靠性的研究成果较少,Imran 等人分析了蛋白质折叠网络中 5 000 个 GPU 集群的故障率模型^[30];Xu 等人利用应用层检查点机制来提高计算可靠性,实现 HiAL-Ckpt 原型系统^[31];研究异构集群的容错调度机制,该方法经修改后可应用到 GPU 集群任务调度算法中^[32].目前尚未出现 GPU 能耗优化对可靠性影响的研究,特别是均衡考虑计算速度、功耗和可靠性多项指标的工作.在嵌入式处理器领域中需要处理实时任务,对可靠性和能耗都有约束限制,因此出现综合考虑能耗和可靠性的研究成果,然而只是关注可靠性和能耗优化的任务调度

① http://homepages.cae.wise.edu/ece753/papers/Paper_4.pdf

算法. 例如, 分析有共同执行时间限制的多任务可靠性感知 DVFS 调度策略^[8]; 针对非周期任务集的可靠性感知的能耗管理, 提出折中可靠性和能耗的任务级调度策略^[9].

由于 GPU 作为众核体系与嵌入式处理器的能耗特征不同, 现存方法很难直接应用到 GPU 集群的能耗优化与可靠性研究中, 因此与本文相关的文献较少, 最相似的是文献[2], 也是本文选择的基准对比控制模型的来源文献. 文献[2]提出 CPU 集群能耗优化控制模型, 采用闭环反馈控制系统来适应各种计算负载的变化, 保证控制精度和稳定性. 然而文献[2]忽视了 DVFS 对可靠性的负面影响, 即降低处理器的频率会引起瞬时故障率的提高. 本文通过最大熵函数产生综合性能指标, 在保证性能和可靠性的前提下达到节能目的.

7 结 论

GPU 集群是一种性能功耗比较高的并行计算系统, 可广泛用于大规模流数据实时处理领域, 在消减冗余能耗的控制过程中保证 GPU 集群的可靠性成为一个新的研究热点. 本文以模型预测反馈控制来处理计算负载变化的不确定性, 用极大熵函数生成综合性能指标, 实现能耗优化过程中的多指标自趋优. 实验表明 MSOC 具有较好的控制稳定性和鲁棒性, 与未考虑计算可靠性的控制模型相比, 更适合处理负载变化剧烈的计算任务. 总之, MSOC 利用降低未达到饱和节点的计算频率来消减冗余能耗, 以损失可容忍的计算速度和能耗为代价, 提高整个集群系统的计算可靠性, 并最终实现集群级的节能目标, 增加能耗控制系统的应用价值. 然而本文未考虑能耗状态切换的代价, 以后将考虑量化能耗状态切换代价, 并且深入研究节能控制鲁棒性的评价机制, 设计更合理、准确的鲁棒性评价方法. 另外, 考虑集群计算中计算任务之间的依赖关系和负载均衡对能耗控制模型的影响因素, 进一步提高对计算负载变化规律的了解, 从而提高能耗控制的准确性.

致谢 感谢匿名审稿人和编辑提出的宝贵意见!

参 考 文 献

- [1] Wang Haifeng, Chen Qingkui. General purpose computing of graphics processing unit: A survey [J]. Chinese Journal of Computers, 2013, 36(4): 757-772 (in Chinese)
(王海峰, 陈庆奎. 图形处理器通用计算关键技术研究综述 [J]. 计算机学报, 2013, 36(4): 757-772)
- [2] Wang Xiaorui, Chen Ming, Fu Xing. MIMI power control for high-density servers in an enclosure [J]. IEEE Trans on Parallel and Distributed System, 2010, 21(10): 1412-1426
- [3] Repantis T, Gu Xiaohui, Kalogeraki V. Qos-aware shared component composition for distributed stream processing system [J]. IEEE Trans on Parallel and Distributed System, 2010, 20(7): 968-982
- [4] Horvath T, Abdelzaher T, Shadron K, et al. Dynamic voltage scaling in multitier web servers with end-to-end delay control [J]. IEEE Trans on Computers, 2007, 56(4): 444-458
- [5] Abdelzaher T F, Stankovic J, Lu C, et al. Feedback performance control in software services [J]. IEEE Control Systems, 2003, 23(3): 74-90
- [6] Wang X, Jia D, Koutsoukos X. DEUCON: Decentralized end-to-end utilization control for distributed real-time systems [J]. IEEE Trans on Parallel and Distributed System, 2007, 18(7): 996-1009
- [7] Lin Yuhua, Kong Fanxin, Xu Huiting, et al. Minimizing energy consumption for linear speedup parallel real-time tasks [J]. Chinese Journal of Computers, 2013, 36(2): 384-391 (in Chinese)
(林宇哈, 孔繁鑫, 徐惠婷, 等. 线性加速比并行实时任务的节能研究 [J]. 计算机学报, 2013, 36(2): 384-391)
- [8] Zhu Dakai, Aydin H. Reliability-aware energy management for periodic real-time Tasks [J]. IEEE Trans on Computers, 2009, 58(10): 1382-1397
- [9] Zhu Dakai. Reliability-Aware dynamic energy management in dependable embedded real-time systems [J]. ACM Trans on Embedded Computing Systems, 2010, 10(2): 26-53
- [10] Zhao Baoxian, Aydin H, Zhu Dakai. On maximizing reliability of real-time embedded applications under hard energy constraint [J]. IEEE Trans on Industrial Informatics, 2010, 6(3): 316-328
- [11] Zhao Baoxian, Aydin H, Zhu Dakai. Energy management under general task-level reliability constraints [C] //Proc of the 18th IEEE Real-Time and Embedded Technology and Applications Symp. Los Alamitos, CA: IEEE Computer Society, 2012: 285-294
- [12] Wang Haifeng, Chen Qingkui. Power consumption prediction model of general-purpose computing GPU with static program slicing [J]. Journal of Software, 2013, 24(8): 1746-1760 (in Chinese)
(王海峰, 陈庆奎. 静态程序切片的 GPU 通用计算功耗预测模型 [J]. 软件学报, 2013, 24(8): 1746-1760)
- [13] Srinivasan S, Jha N. Safety and reliability driven task allocation in distributed systems [J]. IEEE Trans on Computers, 2006, 55(7): 864-879
- [14] Qu Qi, Pei Yong, Modestino J W. An adaptive motion-based uUnequal error protection approach for real-time video transport over wireless IP networks [J]. IEEE Trans on Multimedia, 2006, 8(5): 1033-1044
- [15] Moorthy A K, Seshadrinathan K. Wireless video quality assessment: A study of subjective scores and objective algorithms [J]. IEEE Trans on Circuits and Systems for Video Technology, 2010, 20(4): 587-599

- [16] Zhang Ying, Wang Lihong. Goodness-of-fit test using residuals in infinite-order nonlinear autoregressive models [J]. Journal of Nanjing University: Mathematical Biquarterly, 2012, 29(1): 93-105 (in Chinese)
(张莹, 王立洪. 基于残差的非线性自回归模型的拟合优度检验[J]. 南京大学学报: 数学半年刊, 2012, 29(1): 93-105)
- [17] Gong Maoguo, Cheng Gang, Jiao Licheng, et al. Nodominated individual selection strategy based on adaptive partition for evolutionary multi-objective optimization [J]. Journal of Computer Research and Development, 2011, 48(4): 545-557 (in Chinese)
(公茂果, 程刚, 焦李成, 等. 基于自适应划分的进化多目标优化非支配个体选择策略[J]. 计算机研究与发展, 2011, 48(4): 545-557)
- [18] Chen Yanping, Zhang Jianke, Sun Jiase, et al. A service selection model using mixed intelligent optimization [J]. Chinese Journal of Computers, 2013, 36(2): 384-391 (in Chinese)
(陈彦萍, 张建科, 孙家泽, 等. 一种基于混合智能优化的服务选择模型[J]. 计算机学报, 2010, 33(11): 2116-2124)
- [19] Li Xingsi. A novel effective solution for non-differentiable optimization problems [J]. Scientia Sinica Mathematica, 1994, 24(4): 371-377 (in Chinese)
(李兴斯. 一类不可微优化问题的有效解法[J]. 中国科学: 数学, 1994, 24(4): 371-377)
- [20] Zeng Jing, Xue Dingyu, Yuan Decheng. Distributed model predictive control: A survey [J]. Journal of System Simulation, 2008, 20(21): 5898-5902 (in Chinese)
(曾静, 薛定宇, 袁德成. 分布式模型预测控制方法的研究[J]. 系统仿真学报, 2008, 20(21): 5898-5902)
- [21] Xi Yugeng, Li Dewei. Fundamental philosophy and status of qualitative synthesis of model predictive control [J]. Acta Automatica Sinica, 2008, 34(10): 1225-1234 (in Chinese)
(席裕庚, 李德伟. 预测控制定性综合理论的基本思路和研究现状[J]. 自动化学报, 2008, 34(10): 1225-1234)
- [22] Wang Guibin, Ren Xiaoguang. Power-efficient work distribution method for CPU-GPU heterogeneous system [C] //Proc of 2010 Int Symp on Parallel and Distributed Processing with Applications. Los Alamitos, CA: IEEE Computer Society, 2010: 122-129
- [23] Sunpyo H, Hyesoon K. An integrated GPU power and performance model [C] //Proc of the 37th Annual Int Symp on Computer Architecture. New York: ACM, 2010: 280-289
- [24] Collange S, Defour D, Tisserand A. Power consumption of gpus from a software perspective [C] //Proc of the 9th Int Conf on Computational Science. Berlin: Springer, 2009: 914-923
- [25] Lin Yisong, Yang Xuejun, Tang Tao, et al. An integrated energy optimization approach for CPU-GPU heterogeneous system based on critical path analysis [J]. Chinese Journal of Computers, 2012, 35(1): 123-133 (in Chinese)
(林一松, 杨学军, 唐滔, 等. 一种基于关键路径分析的 CPU-GPU 异构系统综合能耗优化方法[J]. 计算机学报, 2012, 35(1): 123-133)
- [26] Maruyama N, Nukada A, Matsuoka S. A high-performance fault-tolerant software framework for memory on commodity GPUs [C] //Proc of the 24th IEEE Int Parallel and Distributed Processing Symp. Los Alamitos, CA: IEEE Computer Society, 2010: 1-12
- [27] Dimitrov M, Mantor M, Zhou Huiyang. Understanding software approaches for GPGPU reliability [C] //Proc of the 2nd Workshop on General Purpose Processing on Graphics Processing Units. New York: ACM, 2009: 94-104
- [28] Xu Xinhai, Yang Xuejun, Lin Yufei, et al. Fault-Tolerance method for CPU-GPU heterogeneous system [J]. Journal of Software, 2011, 22(10): 2538-2552 (in Chinese)
(徐新海, 杨学军, 林宇斐, 等. 一种面向 CPU-GPU 异构系统的容错方法[J]. 软件学报, 2011, 22(10): 2538-2552)
- [29] Sheaffer J, Luebke D, Skadron K. A hardware redundancy and recovery mechanism for reliable scientific computation on graphics processors [C] //Proc of 2007 Graphics Hardware. New York: ACM, 2007
- [30] Imran S H, Vijay S P. Hard data on soft errors: A large-scale assessment of real-world error rates in GPGPU [C] //Proc of the 10th IEEE/ACM Int Conf on Cluster, Cloud and Grid Computing. Los Alamitos, CA: IEEE Computer Society, 2010: 691-696
- [31] Xu Xinhai, Lin Yufei, Tang Tao, et al. HiAL-Ckpt: A hierarchical application-level checkpointing for CPU-GPU hybrid system [C] //Proc of the 5th Int Conf on Computer Science & Education. Los Alamitos, CA: IEEE Computer Society, 2010: 1895-1899
- [32] Zhu Xiaomin, Zhu Jiangnan, Ma Manhao. Fault-tolerant scheduling for real-time tasks with QoS requirements on heterogeneous clusters [J]. Journal of Software, 2011, 22(7): 1440-1456 (in Chinese)
(朱晓敏, 祝江汉, 马满好. 异构集群系统中具有 QoS 需求的实时任务容错调度[J]. 软件学报, 2011, 22(7): 1440-1456)



Wang Haifeng, born in 1976. PhD. Associate professor of Linyi University, Linyi, Shandong, China. Member of China Computer Federation. His main research interests include high-performance computing, network security and complex networks.



Chen Qingkui, born in 1966. PhD. Currently professor and PhD supervisor of University of Shanghai for Science and Technology, Shanghai, China. Senior member of China Computer Federation. His current research interests include GPGPU, network computing, and the internet of things (chenqingkui@gmail.com).