

# 软硬件节能原理深度融合之绿色异构调度算法\*

王静莲<sup>1,2</sup>, 龚斌<sup>2,3</sup>, 刘弘<sup>4</sup>, 李少辉<sup>4</sup>

<sup>1</sup>(鲁东大学 信息与电气工程学院, 山东 烟台 264025)

<sup>2</sup>(山东大学 软件学院, 山东 济南 250101)

<sup>3</sup>(山东省高性能计算中心, 山东 济南 250101)

<sup>4</sup>(山东师范大学 信息科学与工程学院, 山东 济南 250014)

通讯作者: 王静莲, E-mail: wjljing@163.com



**摘要:** 虚拟云高性能向高效能计算演进, 已是环境保护、人类可持续发展的迫切需求。然而目前, 一方面, 硬件级物理节能空间需要适度延展; 另一方面, 以遗传或人工免疫算法为代表的元启发式调度中间件大多存在进化动力不足, 以致收敛性和分布性冲突难平衡等瓶颈。事实上, 每个候选解(调度方案)都蕴含一定的物理反馈效应, 而拟配资源的非线性和异构性, 则意味着不同方案间与能效相关的实时动态反馈的巨大差异化。因此, 尊重科学规律, 巧妙地借力于硬件节能原理, 给算法优化动力注入新能量, 并进一步增强软件方法的节能主导性, 是本文研究方法; 继而提出一种着眼于软硬件节能原理深度融合的新的绿色异构调度算法(GHSA\_di/II), 以多角度、全方位提升元启发式算法之协同进化模拟的内驱力。大量仿真实验结果显示: 无论对于数据密集型还是计算密集型实例, GHSA\_di/II 算法较其他3种元启发式异构调度算法, 在整体性能、节能降耗以及可扩展性等方面都具明显优势。

**关键词:** 虚拟云; 异构调度; 绿色计算; 协同进化算法; 动力方程; 节能原理; 深度融合

**中图分类号:** TP311

中文引用格式: 王静莲, 龚斌, 刘弘, 李少辉. 软硬件节能原理深度融合之绿色异构调度算法. 软件学报, 2021, 32(12): 3768-3781. <http://www.jos.org.cn/1000-9825/6133.htm>

英文引用格式: Wang JL, Gong B, Liu H, Li SH. Green heterogeneous scheduling algorithm through deep integration of hardware and software energy saving principles. Ruan Jian Xue Bao/Journal of Software, 2021, 32(12): 3768-3781 (in Chinese). <http://www.jos.org.cn/1000-9825/6133.htm>

## Green Heterogeneous Scheduling Algorithm Through Deep Integration of Hardware and Software Energy Saving Principles

WANG Jing-Lian<sup>1,2</sup>, GONG Bin<sup>2,3</sup>, LIU Hong<sup>4</sup>, LI Shao-Hui<sup>4</sup>

<sup>1</sup>(School of Information and Electrical Engineering, Ludong University, Yantai 264025, China)

<sup>2</sup>(School of Software, Shandong University, Ji'nan 250101, China)

<sup>3</sup>(Shandong High Performance Computing Center, Ji'nan 250101, China)

<sup>4</sup>(School of Information Science and Engineering, Shandong Normal University, Ji'nan 250014, China)

**Abstract:** The computing evolution from high performance to high efficiency of the virtual cloud is an urgent need of environmental protection and human sustainable developments. However, on the one hand, nowadays there are moderate extension demands of the hardware energy-saving space; on the other hand, meta-heuristics scheduling algorithms, such as genetic algorithms and artificial immune algorithms, underperform in the optimization dynamics with the balance conflict between convergence and distribution. In fact, there are some inevitable and logical relationships between every candidate solution (scheduling scheme) and some physical feedback; and

\* 基金项目: 国家自然科学基金(61702248, 61070017, 61272094); 国家高技术研究发展计划(863) (2006AA01A113, 2012AA01A306)

Foundation item: National Natural Science Foundation of China (61702248, 61070017, 61272094); National High Technology Research and Development Program of China (863) (2006AA01A113, 2012AA01A306)

收稿时间: 2020-03-15; 修改时间: 2020-07-06; 采用时间: 2020-08-13

nonlinearity and heterogeneity of the allocated resources means a big discrepancy in the feedback effects between different scheduling schemes, such as the energy-efficiencies related. Therefore, the research methods of this study are to respect the scientific laws, and to ingeniously follow the hardware energy-saving principle, in order for injecting new energy into the algorithm optimization power, and also for further enhancing the energy-saving dominance of software methods. Then, the green heterogeneous scheduling algorithm through deep integration of hardware and software energy saving principles, is presented in this paper (i.e., GHSA\_di/II), with the multi angle and all-round improvements of the internal drive of co-evolutionary simulation in the meta-heuristics algorithms. The experimental results show that compared with the other three meta-heuristic heterogeneous scheduling algorithms, GHSA\_di/II algorithm has obvious advantages in overall performance, energy saving, and scalability, for both data intensive and computing intensive instances.

**Key words:** virtual cloud; heterogeneous scheduling; green computing; co-evolutionary algorithm; dynamic equation; energy saving principles; deep integration

近年来,虚拟云聚合广域分布的各类硬件资源(同构或异构集群、存储设备及高级仪器等),形成对用户相对透明、虚拟的高性能环境,深受世界青睐.然而,自 2010 年开始,信息通信技术(information communication technology,简称 ICT)领域因其全球 25%能耗之巨,及当年跟全世界航空运输业相当的 CO<sub>2</sub> 排放量,成为第五大耗能产业<sup>[1]</sup>;另有数据显示,ICT 产业存在很大能源浪费:中国数据中心 PUE(power usage effectiveness)普遍大于 2.2(同期资料显示:美国数据中心的 PUE 也基本保持在 1.9).随着全球能源短缺、极端天气频现以及低碳经济或人类可持续发展的迫切需要,虚拟云高性能向高效能计算的演进,已引起国内外各界广泛关注<sup>[2]</sup>.

目前,相关研究大致集中在硬件和软件两个层面.

通常,云物理层可采用嵌入低功耗组件改变系统架构方法,实现操作过程中能耗监管和优化调节.这种方法一定程度有效,但设计和更换费用过于昂贵<sup>[3,4]</sup>.

事实上,云效能的高低很大程度上由调度中间件决定,即基于服务质量(quality of service,简称 QoS)指标,在一组具有任意特性的云处理器中进行资源分配和任务排序.元启发式(meta-heuristics)优化算法(如遗传算法、人工免疫算法等)是调度引擎核心,其基于 QoS 指标定义的个体适应度或免疫亲和度函数,是仿生群体迭代进化、更新模拟的内驱力<sup>[5,6]</sup>.

这里,将硬件云不影响计算性能而实现能耗值极小化作为评价目标之一,是软件节能的主要途径<sup>[7]</sup>.

具体讲,静态功耗由 CMOS(complementary-symmetry metal oxide semiconductor)电路泄漏电流产生,具有稳态性.动态能耗则源于硬件的物理反馈原理,例如,DVFS(dynamic voltage frequency scaling)或 DPM(dynamic power management)可依据工作负载启用情况更细粒度地进行处理器核内“频率”调节<sup>[8]</sup>;但值得注意的是,众核处理器囿于微结构级 DVFS 或 DPM 配置等区别,会产生不同的频率-电压等级,即“异构性”;表现为即使具有同样的工作负载,异构处理器动态功耗开销也差异极大.因而,存在相当大节能空间是微体系结构级所无法涉足而亟待调度中间件宏观调控的.

与此同时,动态能效指标量化,国内外目前大多沿用模糊估算方法:基于性能计数器,实时统计与耗能关联度大的硬件事件,进而借助经验得到粗略值.由于硬件应用事件总以螺旋式上升发展,这种动态功耗的模糊估算模式大多仅适于同构集群.

换言之,以元启发式算法为基础的调度中间件面对异构高维优化难题大多存在进化动力不足、个体多样性不够或收敛速度过慢等不足<sup>[9,10]</sup>.

着眼于多角度、全方位提升算法之协同进化模拟内驱力,本文提出一种新的绿色异构调度算法(GHSA\_di/II).本研究属于异构集群体系、云调度中间件和分布式人工智能等多学科交叉方向,是文献[11]后续 3 年的最新成果.

## 1 相关工作

目前,通用多核微处理器(central processing unit,简称 CPU)与定制加速协处理器(graphic processing unit,简称 GPU)混合体系已成为千万亿次高性能计算机一种发展趋势.由于存在 DVFS 配置及 CMOS 制造工艺等方面

的差异,实时处理器可调电压边界、对应不同负载启用粒度的电压和频率级别构成以及工作频率与动态能耗相互之间影响规律非常复杂<sup>[3,4,8]</sup>,因此,面对计算资源异构性,仅靠调频核心技术之“负载→频率”应答机制,还存在诸多节能约束;而对异构集群主要耗能部件的实时动态功耗,从测量方法到预估模型现在都面临一系列挑战。

大规模实时调度一般要求在性能不降低的情况下减少能耗值。现有的实时调度研究大多关注同构系统的两类简单任务对象:周期任务和静态任务。在这两种类型中,假定“任务”是不能被抢占的最小单位,作业都被假设为无限调用任务序列;且模型中不存在作业内并行性,即在任意时刻,每个作业实例最多在一个处理器上执行<sup>[7]</sup>。

异构实时调度,重点介绍 4 个代表性新成果。文献[12]提出的 EEVS 引擎在调度目标中引入通信成本参数  $K$ : 设定好  $K$  值后,把当前任务映射到某处理单元,如果增加的通信量边大于平均通信量与  $K$  值的积,则更改映射; $K$  的最优取值与具体任务图相关,可通过实验确定。文献[13]提出了求解最佳处理单元数目算法,实现了处理机数目和时钟频率的有效平衡,但由于处理机数目的上下界范围过宽,使用二分查找和线性查找确定处理机数目的方式增加了算法执行时间和复杂度。GOA(greedy online algorithm)及 TOAA(2-approximation online algorithm)是文献[14]分别面向同构、异构系统提出的实时动态任务引擎的最新研究。TOAA<sup>[14]</sup>面对计算资源的异构性,对每个任务组反复查找最小化能耗的最佳匹配,但任务分配完全基于静态分组和复制结果。文献[15]提出了异构系统下并行应用引擎 p-BFF(power-aware backfilling-first-fit):p-BFF<sup>[15]</sup>忽略调频处理器动态能耗量化,仅优化静态功耗,不适合数据密集应用。

另外,上述调度算法大多为基于启发式算法(heuristic algorithm)。对 QoS 高维指标需求,启发式调度会将多目标归一聚合至单目标函数,得到决策空间的一个可行解:降低了最终解的质量,缺乏灵活性和扩展性<sup>[16-18]</sup>。

近年来,模拟达尔文“适者生存”自然进化论或生物免疫等机制的一类仿生智能算法(如遗传算法、人工免疫算法等),被广泛应用于云调度优化问题<sup>[5,6]</sup>。这类方法基于仿生个体(候选解)特征编/解码,元启发式(meta-heuristics)搜索解空间,具体包括遗传算法(genetic algorithms,简称 GA)、人工免疫算法(artificial immune algorithms)、粒子群算法(particle swarm optimization,简称 PSO)、模因算法(memetic algorithms,简称 MA)和细胞模因算法(cellular MA,简称 cMA)等,以及元启发式算法与其他一些技术的融合,比如神经网络(neural networks)、变邻域搜索(variable neighborhood search)和列表调度技术(list scheduling techniques)<sup>[19,20]</sup>。

元启发式算法的并行与分布式设计具体可划为主从模型(master-slave model)、细粒度模型(fine-grained model)和粗粒度模型(coarse-grained model),其中,粗粒度模型也称为孤岛模型(island model),被广泛应用在众核 CPU+GPU 混合结构的超计算机调度服务器上,孤岛模型相关研究主要集中在选择函数、替代函数、迁移率或拓扑结构的设计上<sup>[21-23]</sup>。

前期成果<sup>[11]</sup>为解决异构资源管理存在问题以适应新环境、新应用、新需求及新特征,围绕节能减排及调度协同等热点,尝试将一些硬件特性结合到异构调度中间件的目标评测中。对比实验结果显示:调度引擎在节能等方面优势明显,但由于技术、时间及资金等方面限制,调度模型中的能耗参数仅是凭借先验大体预估。

## 2 软硬件节能原理深度融合的绿色异构调度算法

通常,云体系包含用户层、中间件层、虚拟资源层和基础设施层(如图 1 所示)。用户作业可解析产生若干并行依赖任务集,一般建模为 DAG(directed acyclic graph)集合;中间件负责作业调度、执行,也是压力最集中组件。鉴于 CPU 和 GPU 都是并行计算资源,本文在称谓上不作区分,采用统一标识并进行异构属性说明。

调度 QoS 评价指标体系与相应元启发式(meta-heuristics)算法(如遗传算法、人工免疫算法等)是调度引擎核心;受达尔文“适者生存”进化论启发的遗传算法,通常直接用 QoS 多目标多约束函数评价个体适应度;类似地,在模拟生物免疫系统的人工免疫算法(artificial immune algorithms)中,导致系统产生抗体的抗原也被定义为调度 QoS 评价函数。

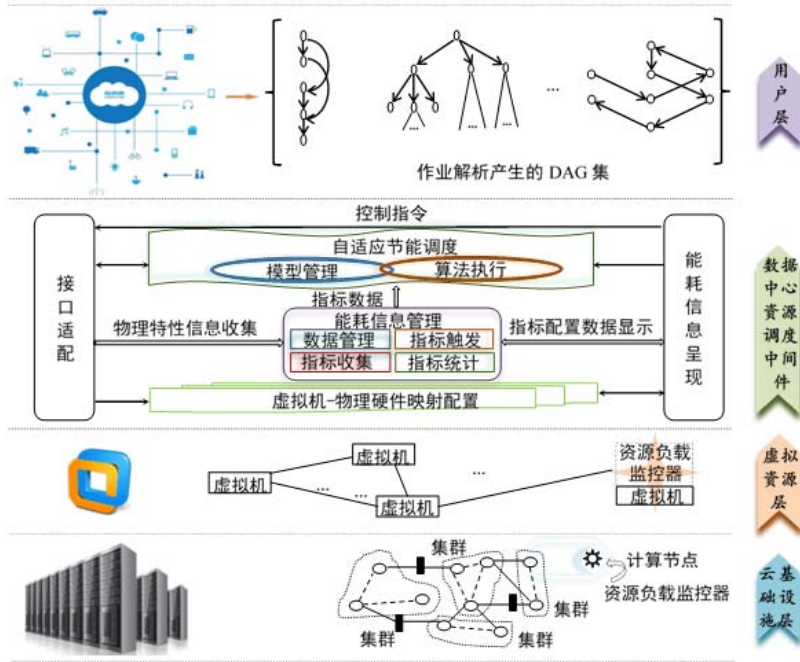


Fig.1 General architecture of the virtual cloud

图 1 虚拟云的常规体系结构

2.1 一种重视并兼发硬件节能正反馈优势的调度寻优动力方程

计算云异构众核处理器虽然品牌众多,但按微结构级 DVFS 配置或动态能耗型区分,大致可划分为 3 类:片外独立 DVFS 支持、片上全局 DVFS 支持和片上每个核的 DVFS 支持。

异构系统实时调度 QoS 评价通常包括性能、经济或技术等指标.借助数字传感及物联网等实验技术,历经异构众核体系电路(非线性)特征信号实时采集、节能机理模拟大数据的回归集成,参数间显、隐性关系深度挖掘以及代表性参数筛选等核心步骤,本文突破异构众核体系动态能耗的多元非线性回归量化,并数学定义一系列可预知(拟匹配)硬件资源物理反馈的调度 QoS 评价指标,亦即调度算法中的进化寻优动力方程构建的重要组成部分.其中,表 1 显示了本文进化动力方程中的定义变量及代表意义。

**定义 1(动态能耗).** 计算云处理器按微结构级 DVFS 配置或动态能耗型区分,大致可划分为 3 类,即  $v \in \{1,2,3\}$ .本质上,动态能耗(单位:Wh)等于动态功耗(单位:W)与执行时间(用  $\Delta T_i$  表示)的乘积。

这里,即使具有同样的工作负载,而由于对应不同的“频率-电压等级”,能耗异构型处理器的动态实时功耗开销差别极大.历经参数间显、隐性关系深度挖掘、代表性参数筛选以及实验数据的多元非线性回归量化等核心步骤,异构众核处理器体系的动态能耗(用  $D\_energy(\phi)$  表示)可定义为公式(1):

$$D\_energy(\phi) = \sum_{k=1}^{n_k} \sum_{h=1}^{n_h^k} \sum_{i=1}^{n_{F_i^v}^k} \{load^v [\delta_+(F_i^v) - \delta_-(F_i^v)] + \delta_+(F_i^v)\} \times \Delta T_i \tag{1}$$

**定义 2(响应时间).** 作为重要的异构实时调度 QoS 度量指标之一,预估响应时间主要考量虚拟机的执行时间,具体涉及两方面因素:所有虚拟机必须要执行的指令数目以及每个虚拟机的指令处理能力(比如每秒可处理的指令数目).因而,响应时间(用  $R\_time(\phi)$  表示)可定义为公式(2):

$$R\_time(\phi) = \max_{k=1}^{n_k} \max_{h=1}^{n_h^k} \max_{\theta=1}^{\omega_h^k} (N_{Ins}^\theta / \omega_{F_i^v}^{k,h,\theta}) \tag{2}$$

公式(2)中,虚拟机  $\theta$  可映射到能耗型  $v \in \{1,2,3\}$  的计算节点  $k_h$  上,而且其工作频率是  $F_i^v$ 。

**Table 1** Related variables of evolutionary dynamic equation and their representative meanings**表 1** 进化动力方程相关变量及代表意义

变量	描述
$v$	按微结构级 DVFS 配置或动态能耗特征划分的云处理器类型
$n_k$	异构众核体系中计算集群个数
$k_h$	集群 $k$ 中执行任务的计算节点
$n_h^k$	集群 $k$ 中执行任务的计算节点个数
$F_i^v$	能耗型 $v(v \in \{1,2,3\})$ 处理器的实时工作频率
$n_v^v$	能耗型 $v(v \in \{1,2,3\})$ 处理器可应用的工作频率集合的元素个数
$load^v$	能耗型 $v(v \in \{1,2,3\})$ 处理器以工作频率 $F_i^v$ 运行时动态功耗相关系数
$\delta_v(F_i^v)$	满载的能耗型 $v(v \in \{1,2,3\})$ 处理器被强制按工作频率 $F_i^v$ 运行时实验获取的动态功耗
$\delta_v(F_i^v)$	空载的能耗型 $v(v \in \{1,2,3\})$ 处理器被强制按工作频率 $F_i^v$ 运行时实验获取的动态功耗
$N_{Ins}^\theta$	虚拟机 $\theta$ 必须要执行的指令数目
$\omega_{F_i^v}^{k_h, \theta}$	计算节点 $k_h$ 映射的虚拟机 $\theta$ 每秒按工作频率 $F_i^v$ 运行时可处理的指令数目
$\tau_v^{k_h}$	计算机点 $k_h \in R^+$ 的最大计算力
$\varpi_h^k$	映射到计算节点 $k_h \in R^+$ 的虚拟机数目
$\alpha_i^j$	任务 $\alpha_i(i \in \{1, \dots, m\})$ 对第 $j(j \in \{1, \dots, q\})$ 项安全的需求权重
$s_i^j$	任务 $\alpha_i(i \in \{1, \dots, m\})$ 第 $j(j \in \{1, \dots, q\})$ 项安全需求可获取的安全增益
$\rho_i$	任务 $\alpha_i(i \in \{1, \dots, m\})$ 是否被调度

**定义 3(可扩展性).** 对可具动态电压频率缩放的异构众核处理器体系,扩展性(用  $Scal(\phi)$ )表示)代表运行中的所有计算节点的最大计算潜力,亦即暂不新增计算节点情况下的平均计算力。

具体讲,扩展性取决于计算机点  $k_h \in R^+$  的最大计算力以及当前计算力:

$$Scal(\phi) = \left\{ \sum_{k=1}^{n_k} \sum_{h=1}^{n_h^k} (\tau_v^{k_h} - \omega_{F_i^v}^{k_h, \theta}) \right\} / \left( \sum_{k=1}^{n_k} n_h^k \right) \quad (3)$$

**定义 4(系统鲁棒性).** 为预防网络故障和黑客攻击,系统鲁棒性(用  $Robust(\phi)$ )表示)是异构实时调度 QoS 评价的又一重要指标。

具体可解释为每个计算节点的虚拟机平均映射数目,或者计算节点  $k_h \in R^+$  失联,需要迁移或移植的虚拟机数目:

$$Robust(\phi) = \left( \sum_{k=1}^{n_k} \sum_{h=1}^{n_h^k} \varpi_h^k \right) / \left( \sum_{k=1}^{n_k} n_h^k \right) \quad (4)$$

**定义 5(安全增益).** 异构实时调度 QoS 又一重要指标是最大化保证所有可调度任务的安全性.鉴于云服务安全需求的差异化,某一任务(用  $\alpha_i(i \in \{1, \dots, m\})$ )表示)的安全增益可定义为公式(5):

$$\chi_i(\phi) = \sum_{j=1}^q \alpha_i^j s_i^j \quad (5)$$

进而,所有任务的安全增益(用  $Security(\phi)$ 表示)可定义为公式(6):

$$Security(\phi) = \sum_{i=1}^m \rho_i \chi_i = \sum_{i=1}^m \rho_i \sum_{j=1}^q (\alpha_i^j s_i^j) \quad (6)$$

公式(6)中,针对实时任务要求存在不能满足的情况(比如服务延迟或负载约束等), $\rho_i(i \in \{1, \dots, m\})$ 表示任务  $\alpha_i(i \in \{1, \dots, m\})$  是否被调度; $\rho_i=1(i \in \{1, \dots, m\})$ 表示任务  $\alpha_i(i \in \{1, \dots, m\})$  被调度;反之, $\rho_i=0(i \in \{1, \dots, m\})$ 表示任务  $\alpha_i(i \in \{1, \dots, m\})$  没被调度。

自适应绿色调度的目标就是在满足 QoS 强约束的同时,寻求调度方案( $\phi \in \Phi$ ),使时效、能效收益最优,系统鲁棒性最小,以及可扩展性、安全性最强等.综合上述各类指标量化定义,本文进化动力方程如公式(7)所示( $A_i$ 表示 QoS 指标的权重因子):

$$G(\phi) = \min_{\phi \in \Phi} [A_1 \cdot D\_energy(\phi) + A_2 \cdot R\_time(\phi) + A_3 \cdot Robust(\phi) - A_4 \cdot Scal(\phi) - A_5 \cdot Security(\phi)] \quad (7)$$

2.2 异构调度候选解的三维编/解码设计

仿照生物遗传或人工免疫理论,一个智能调度(候选)方案看作是基因组组合的进化个体;本算法中,候选解(仿生个体) $Ch_r(i \in \{r \in R^+\})$ 的基因特征  $G_i^r(i \in \{r \in R^+\})$ 拟表示为三维编码  $(X_i^r, Y_i^r, Z_i^r)$ ,代表随机任务  $X_i^r(i \in \{r \in R^+\})$ 被分配到计算节点  $Z_i^r(i \in \{r \in R^+\})$ 的虚拟机  $Y_i^r(i \in \{r \in R^+\})$ 上。

如图 2 所示,每个立方体代表候选解不同维度的取值范围,而其中彩色填充部分则表示一个候选解(仿生个体),即所有任务与计算节点及虚拟机编号的某种配置映射.因而,候选解个体  $Ch_r(i \in \{r \in R^+\})$ 可编码为三维矩阵(如公式(8)):

$$Ch_r = \begin{bmatrix} X_1^r & Y_1^r & Z_1^r \\ X_2^r & Y_2^r & Z_2^r \\ \vdots & \vdots & \vdots \\ X_i^r & Y_i^r & Z_i^r \\ \vdots & \vdots & \vdots \\ X_m^r & Y_m^r & Z_m^r \end{bmatrix} \tag{8}$$

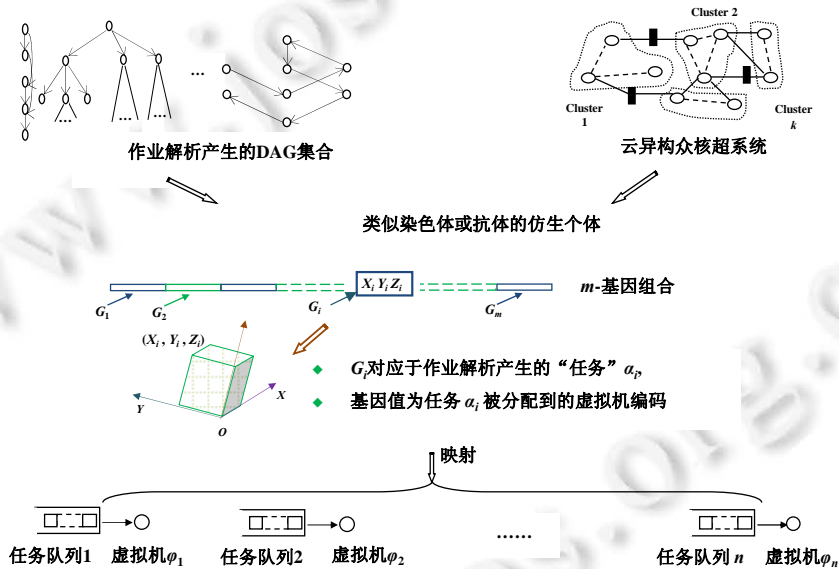


Fig.2 Three-dimensional coding of the candidate solution (bionic individual)

图 2 候选解(仿生个体)的三维编码表示

鉴于编码的有效合理性,所有仿生个体第一维列向量  $[X_1^r, X_2^r, \dots, X_i^r, \dots, X_m^r]^T$  对应随机任务  $[1, 2, \dots, i, \dots, m]^T$ , 是确定的值;第二维列向量元素(虚拟机编号)的上限取决于单节点虚拟机配置的最大值;第三维列向量元素(节点编号)的最大值则由计算节点总数确定.这里,  $m$  表示随机任务个数.

本算法解码细则考虑不同任务分配到同一虚拟机两种情形.

- (1) 逻辑深度不同:遵循深度值排序原则,以避免任务之间长时间等待甚至发生死锁;
- (2) 逻辑深度相同:遵循关联耦合强度排序原则,以缩短关键路径长度达到最优效果.

2.3 基于三维编码的协同进化算子定义

基于三维编码的基因组进化模拟包括个体选择、交叉、变异以及克隆等智能算子定义.

克隆算子在 GHSA\_di/II 算法中对绿色异构调度候选解之多样性和逼近性起着重要作用,通常,其对仿生种群  $Ch = \{Ch_1, Ch_2, \dots, Ch_c, \dots, Ch_\theta\}$  的克隆操作  $\Gamma_C$  可以定义为公式(9):

$$\begin{aligned}
Ch^*(t) &= \Gamma_C\{Ch_1(t), Ch_2(t), \dots, Ch_\epsilon(t), \dots, Ch_{\theta(t)}(t)\} \\
&= \Gamma_C(Ch_1(t)) + \dots + \Gamma_C(Ch_\epsilon(t)) + \dots + \Gamma_C(Ch_{\theta(t)}(t)) \\
&= \{Ch_1^1(t), Ch_1^2(t), \dots, Ch_1^\epsilon(t), \dots, Ch_1^{\theta(t)}(t)\} + \dots + \{Ch_{\theta(t)}^1(t), Ch_{\theta(t)}^2(t), \dots, Ch_{\theta(t)}^\epsilon(t), \dots, Ch_{\theta(t)}^{\theta(t)}(t)\}
\end{aligned} \tag{9}$$

这里:  $\Gamma_C(Ch_i(t)) = \{Ch_i^1(t), Ch_i^2(t), \dots, Ch_i^\epsilon(t), \dots, Ch_i^{\theta(t)}(t)\}$ ,  $i = 1, 2, \dots, \theta(t)$ ;  $\mathbf{z}i \in [1, m_c]$  是可调节参数, 表示克隆概率;  $m_c$  是克隆概率上限;  $\mathbf{z}i=1$  代表不对  $Ch_i(t)$  进行克隆操作。

在 GHSA\_di/II 算法中, 对每个仿生个体采用一致的克隆概率  $\mathbf{z}$ , 使得优化过程中的可行非支配解集规模几乎成倍上升, 保持个体多样性同时, 还可加速群体收敛. 具体如公式(10)所示:

$$Ch^*(t) = \{Ch_1^1(t), Ch_1^2(t), \dots, Ch_1^{\mathbf{z}}(t)\} + \dots + \{Ch_{\theta(t)}^1(t), Ch_{\theta(t)}^2(t), \dots, Ch_{\theta(t)}^{\mathbf{z}}(t)\} \tag{10}$$

与克隆相反, 选择操作将种群个体划分为非劣解或劣解, 且只有非劣解可进入下一代. 对于任何仿生个体  $Ch^\#(t) \in Ch^*(t)$ , 如果  $Ch^\#(t)$  满足公式(11)称为非劣解, 否则是劣解:

$$\neg \exists \delta_\kappa^{\sigma^\#}(t) \neq \delta^\#(t) (\kappa = 1, 2, \dots, \theta; \sigma = 1, 2, \dots, \mathbf{z}) \in \delta^{\sigma^\#}(t) : (\forall i \in \{1, \dots, m\} : f_i(\delta^\#(t)) \geq f_i(\delta_\kappa^{\sigma^\#}(t))) \tag{11}$$

而对仿生种群的选择操作  $\Gamma_S$ , 可定义为公式(12):

$$\begin{aligned}
Ch^{**}(t) &= \Gamma_S(Ch^*(t)) \\
&= \Gamma_S(\{Ch_1^1(t), Ch_1^2(t), \dots, Ch_1^{\mathbf{z}}(t)\} + \dots + \{Ch_{\theta(t)}^1(t), Ch_{\theta(t)}^2(t), \dots, Ch_{\theta(t)}^{\mathbf{z}}(t)\}) \\
&= \Gamma_S(\{Ch_1^1(t), Ch_1^2(t), \dots, Ch_1^{\mathbf{z}}(t), \dots, Ch_{\theta(t)}^1(t), Ch_{\theta(t)}^2(t), \dots, Ch_{\theta(t)}^{\mathbf{z}}(t)\}) \\
&= \{Ch_1^\#(t), Ch_2^\#(t), \dots, Ch_\epsilon^\#(t), \dots, Ch_{\theta_s(t)}^\#(t)\}
\end{aligned} \tag{12}$$

其中,  $Ch_\epsilon^\#(t)$  ( $\epsilon = 1, \dots, \theta_s(t)$ ) 表示仿生种群  $Ch^{**}(t)$  中的非劣个体,  $\theta_s(t)$  是非劣个体的数目. 通常, 协同进化系统根据当代个体的约束偏离值选取可行解集, 然后根据个体的目标函数值选取可行非劣解集进行个体选择.

而 GHSA\_di/II 算法可依据第 2.1 节定义的进化动力信息直接在种群中选取非劣仿生个体, 算法的效率极大提升.

与此同时, 采用不同的基因交叉、变异策略, 可有助于种群多样性保持及仿生个体间协同或信息交换. GHSA\_di/II 算法对仿生个体种群的基因操作  $\Gamma_G$  可定义为公式(13):

$$\begin{aligned}
Ch^{**}(t) &= \Gamma_G(Ch^*(t)) \\
&= \Gamma_G(\{Ch_1^1(t), Ch_1^2(t), \dots, Ch_1^{\mathbf{z}}(t)\} + \dots + \{Ch_{\theta(t)}^1(t), Ch_{\theta(t)}^2(t), \dots, Ch_{\theta(t)}^{\mathbf{z}}(t)\}) \\
&= \{\Gamma_G(Ch_1^1(t)) + \Gamma_G(Ch_1^2(t)) + \dots + \Gamma_G(Ch_1^{\mathbf{z}}(t))\} + \dots + \{\Gamma_G(Ch_{\theta(t)}^1(t)) + \Gamma_G(Ch_{\theta(t)}^2(t)) + \dots + \Gamma_G(Ch_{\theta(t)}^{\mathbf{z}}(t))\} \\
&= \{Ch_1^{\#}(t), Ch_1^{2\#}(t), \dots, Ch_1^{\mathbf{z}\#}(t)\} + \dots + \{Ch_{\theta(t)}^{\#}(t), Ch_{\theta(t)}^{2\#}(t), \dots, Ch_{\theta(t)}^{\mathbf{z}\#}(t)\}
\end{aligned} \tag{13}$$

通常, 协同进化系统模拟二进交叉(SBX cross-over)算子或多项式变异(polynomial mutation)算子. 在 GHSA\_di/II 算法中, 交叉点或变异点的选取同样可依据第 2.1 节定义的进化动力信息.

#### 2.4 融合非传统主从式和粗粒度的多层次并行化设计及算法描述

目前, 通用多核微处理器与定制加速协处理器混合体系已成为千万亿次高性能计算机一种发展趋势, 因而, 为适于调度服务器的众核处理器超混合硬件体系, GHSA\_di/II 算法采用非传统主从式和粗粒度相融合的多层次并行与分布式设计.

一方面, 依照粗粒度模型, 算法将整个进化群体划分成若干子群(第 3 步), 子群被分配到不同节点上进行相关的协同进化模拟计算, 并适时采取有效的迁移策略(第 12 步~第 19 步); 另一方面, 在每一个节点上, 大量的进化动力方程(如遗传算法的适应度函数或人工免疫算法的抗原)计算采用 CPU-GPU 非传统的主从式模型(第 5 步~第 11 步). 这里, 主服务器是 CPU, 而在 GPU 众核上大量执行的并行线程可看作客户端.

##### GHSA\_di/II algorithm.

- 1: **Initialize** the iteration ( $t$ ) and the subpopulation  $\mathcal{E}(t) = \{Ch_1(t), Ch_2(t), \dots, Ch_\epsilon(t), \dots, Ch_\theta(t)\}$ , each subpopulation of  $\theta$  individuals;
- 2: **While** ( $t < t_{\max}$ ) and (other termination criteria are not satisfied)

```

3:   Do in parallel for each island /*Obtain coarse-grained model, one of the parallel models*/
4:      $t=t+1$ ;
5:   Do in parallel /*Obtain master-slave model, another parallel model*/
6:     Evaluate genome fitness on the dynamic equation (Eq.(7)) in the current subpopulation:
        $\Gamma(\mathbf{Ch}_r(\tau))(r \in R^+, \mathbf{Ch}_r(\tau) \in \Xi(\tau))$ ;
7:     Sort the individuals fitness:  $\Gamma(\mathbf{Ch}_r(\tau))(r \in R^+, \mathbf{Ch}_r(\tau) \in \Xi(\tau))$ , and save the fittest individual  $\mathbf{Ch}_{elite}(t)$ 
       in the external memory;
8:     Perform local search strategies to ensure the Cesaro average convergence of the irreducible
       aperiodic Markov chain of the population sequence;
9:     Perform clonal operation;
10:    Perform gene operations, such as crossover and mutation defined as Section 2.3;
11:  End Do in parallel
12:  If  $t=\tau$  (migration interval) then
13:    Create  $\Psi_\delta$  for the current subpopulation;
14:    Send  $\Psi_\delta$  to the neighboring subpopulation;
15:    Receive  $\Psi_\delta$  from the neighboring subpopulation;
16:    Construct the founding subpopulation  $\Xi$ ;
17:    Select  $\phi$  individuals into  $\Xi$ ;
18:    Replace the subpopulation  $\Psi_\delta$  with  $\Psi_\delta^r$ ;
19:  End If
20:  End Do in parallel
21: End While
22: Output the best individual.

```

### 2.5 算法时间复杂度分析

设在每一代进化中,种群  $FeaNonPop$  和  $ModNonPop$  的规模都为  $\theta$ ,克隆倍数为  $3$ ,变量的维数为  $\mathcal{L}$ ,约束维数为  $\mathcal{H}$ ,目标函数维数为  $m$ ,则:

- 在每次克隆种群  $ModNonPop$  所用的复杂度为  $O(3\theta)$ ;
- 交叉操作所需复杂度为  $O(\mathcal{L}3\theta/2)$ ;
- 变异操作所需复杂度为  $O(\mathcal{L}3\theta)$ ;
- 计算种群  $Pop$  基因亲和度值的时间复杂度为  $O(\mathcal{L}3\theta)$ ;
- 合并种群  $ModNonPop$  所需复杂度为  $O(\mathcal{L}\theta + \mathcal{H}\theta + m\theta)$ ;
- 修正种群  $Pop$  中个体模因值所需复杂度为  $O(3m(3+1)\theta + 2\mathcal{H}(3+1)\theta)$ ;
- 选取并更新可行非支配解集所需复杂度为  $O((23+6+m3+2m)\theta + m(3+2)^2\theta^{2+(3+2)}(m+1)\theta \log_2((3+2)\theta))$ ;
- 选取并更新非支配解集所需复杂度为  $O((m+1)(3+1)\theta + \theta + m(3+1)^2\theta^{2+(m+1)(3+1)}\theta \log_2((3+1)\theta))$ .

因此,算法 GHSA\_di/II 的复杂度为多项式时间(polynomial time).

## 3 仿真实验及结果

实验在山东省高性能计算中心进行,中心拥有先进异构集群、存储和网络资源,可满足日益增长的云研究及应用服务需求.同时,中心长期为各种数据密集应用提供服务,可就多种输入实例展开绿色驱动调度研究,以保证其实际通用性.具体包括如下配置.

- 曙光天潮高性能计算集群;



- 浪潮 TS10000 计算集群(100 个节点,200 颗 2.8GHz Intel®Xeon®CPU);
- 浪潮 TS20000 计算集群(6 个节点,24 颗 1.3GHz Intel®ItaniumII CPU);
- HP DL580 高性能服务器;
- Dell r720 服务器、Dell 交换机等大数据处理平台;
- IBM p690(32 颗 1.7GHz Power4+CPU,128G 内存,6TB 硬盘);
- IBM NAS300G 存储系统(6.6TB 容量);
- 网络系统(InfiniBand QDR,链路速率 40Gbps,聚合带宽 69.6TB/s);
- 操作系统(Windows Server 2008);
- 编程语言及环境(C-CUDA, Fortran, Java, MPI);
- 互联网出口(教育网和联通运营商专线).

### 3.1 实验参数设置

实验分为两部分:第 1 部分探讨 GHSA\_di/II 算法求解异构调度优化问题的整体性能,对比算法是新发表的 3 个实时调度算法代表——PPADE 算法<sup>[21]</sup>、MOCTS-AI 算法<sup>[22]</sup>和 MaOEA/C 算法<sup>[23]</sup>;第 2 部分则比较分析软件节能原理深度融合给异构调度优化解集带来的绿色感知影响,此部分采用的任务实例包括计算密集型和数据密集型两类.

- (1) 调度实验采用 200 个计算集群,包括常见的 3 种能耗异构类型( $v \in \{1,2,3\}$ ),而且基本各占 1/3;
- (2) 集群编号按能耗类型依次设置,即能耗类型  $v=1$  集群的编号集中排在前面,然后依次是能耗类型  $v=2$ ,  $v=3$  集群的编号;
- (3) 集群节点处理器和硬盘各自的初始利用率大致范围在[10%,40%];
- (4) 在后续小节中,“*processor-optim*”和“*disk-optim*”分别表示集群节点能效最高时,处理器和硬盘各自的理论最优利用率.具体讲, {*disk-optim*: [0.75,0.8],*processor-optim*: [0.8,0.9], $v:1$ } 表示对于能耗型  $v=1$  的计算节点,其处理器和硬盘各自的利用率范围分别为 [80%,90%] 和 [75%,80%],能效最高;同理,对于  $v=2$  和  $v=3$  的计算节点,存在理论最优值: {*disk-optim*: [0.6,0.65],*processor-optim*: [0.6,0.7], $v:2$ } 和 {*disk-optim*: [0.45,0.5],*processor-optim*: [0.4,0.5], $v:3$ }.

### 3.2 整体性能比较

首先,实验探讨 GHSA\_di/II 算法求解异构调度优化问题的整体性能,对比算法是新发表的 3 个元启发式实时调度算法代表:PPADE 算法<sup>[21]</sup>、MOCTS-AI 算法<sup>[22]</sup>和 MaOEA/C 算法<sup>[23]</sup>.

评价指标包括能耗值(公式(1))以及系统总体性能(overall system performance,简称 OSP).这里,保证率(guarantee ratio)是针对实时任务要求存在不能满足的情况(比如服务延迟或负载约束等),统计的可调度任务数所占比例;系统总体性能是标准化的安全值与保证率乘积.

可调度计算节点从 8 个增加至 256 个实验过程中,图 3 显示了 4 种算法各自的 4 项性能评价.

具体讲,从节能、安全性以及系统总体性能这 3 项评价指标看,GHSA\_di/II 算法的解质量明显优于 PPADE 算法<sup>[21]</sup>、MOCTS-AI 算法<sup>[22]</sup>和 MaOEA/C 算法<sup>[23]</sup>;这里,仅从保证率指标看,GHSA\_di/II 算法与 MaOEA/C 算法<sup>[23]</sup>接近.

更重要的是:对比 3 种算法——PPADE 算法<sup>[21]</sup>、MOCTS-AI 算法<sup>[22]</sup>和 MaOEA/C 算法<sup>[23]</sup>,GHSA\_di/II 算法不仅具备显著的高能效优化性能,而且随着可调度计算节点从 8 到 256 的增加,这种优势迅速提升.

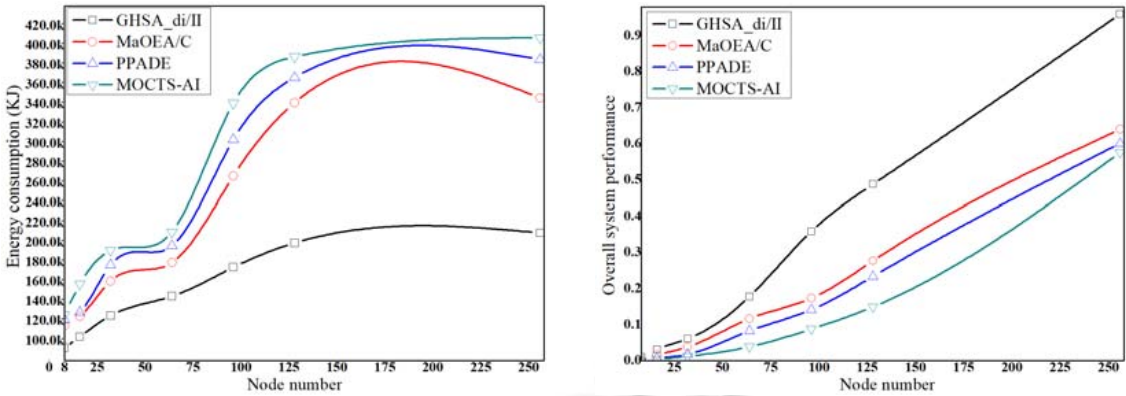


Fig.3 Performance comparison between four meta-heuristics heterogeneous scheduling algorithms

图 3 4 种元启发式异构调度算法的整体性能对比

3.3 软硬件节能原理深度融合的绿色感知优化影响

本节实验重点探讨软硬件节能原理深度融合给以元启发式算法(如遗传算法、人工免疫算法等)为基础异构调度算法带来的绿色感知优化影响,对比算法是在上节实验中显现与 GHS\_A-di/II 算法之调度保证率性能指标值接近的 MaOEA/C 算法<sup>[23]</sup>.

实验按实时任务类型分成两部分:计算密集型和数据密集型.

3.3.1 计算密集型实时任务集的异构调度实验结果及分析

对于计算密集型任务,每个应用实例被划分成 20 000 个子任务,即  $m=20000$ ,并将虚拟机数量设置为 5 000.

- 如图 4(a)所示:200 个计算集群尽管具有 3 种动态能耗异构类型的处理器,但经 MaOEA/C 算法<sup>[23]</sup>调度后,异构处理器利用率之间没有明显差异;
- 反之,从图 4(b)分析,GHS\_A-di/II 算法得到的调度优化解显示,200 个计算集群 3 种异构处理器的资源利用率可分别接近 0.9,0.7 和 0.5,在相应的理论最优值范围内.

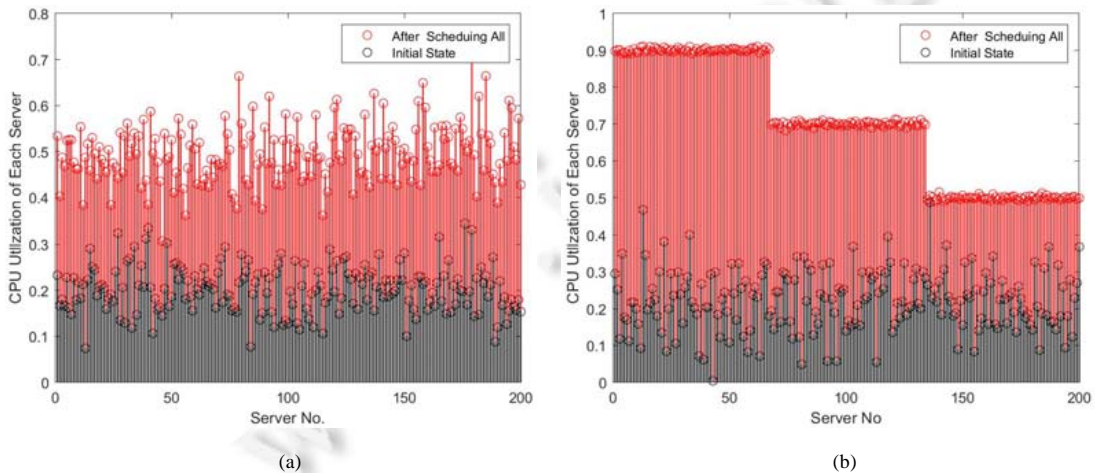


Fig.4 Comparison of CPU utilization after scheduling computing tasks by MaOEA/C<sup>[23]</sup> and GHS\_A-di/II

图 4 MaOEA/C 算法<sup>[23]</sup>和 GHS\_A-di/II 算法调度计算密集型任务后的处理器利用率对比

类似情形也出现在两种算法调度解产生的硬盘利用率差异上(如图 5 所示).

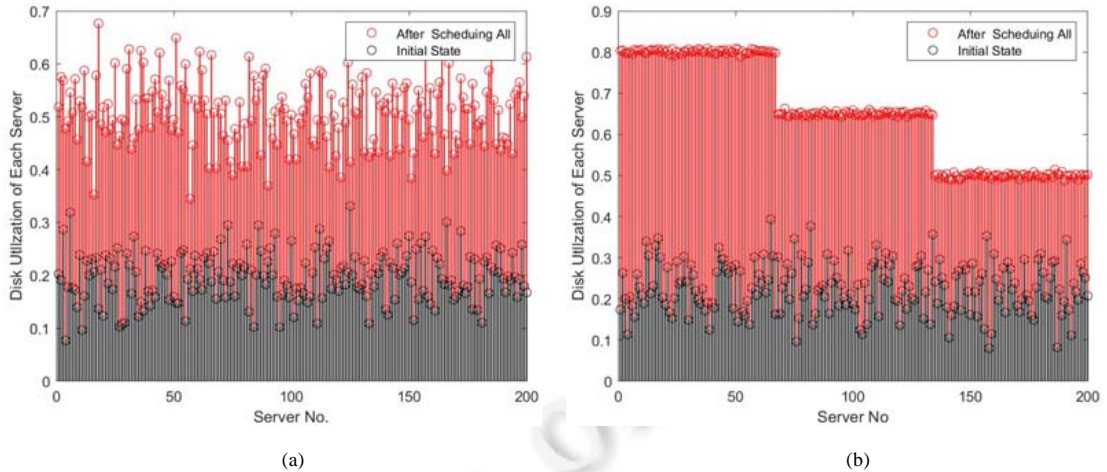


Fig.5 Comparison of disk utilization after scheduling computing tasks by MaOEA/C<sup>[23]</sup> and GHSA\_di/II  
图5 MaOEA/C 算法<sup>[23]</sup>和 GHSA\_di/II 算法调度计算密集型任务后的硬盘利用率对比

### 3.3.2 数据密集型实时任务集的异构调度实验结果及分析

对于数据密集型任务,每个应用实例同样被划分成 20 000 个子任务,并将虚拟机数量设置为 5 000.

- 如图 6(a)、图 7(a)所示:与计算密集型实时调度情形类似,经 MaOEA/C 算法<sup>[23]</sup>调度后,异构处理器和硬盘利用率之间没有明显差异;
- 反之,从图 6(b)、图 7(b)分析,GHSA\_di/II 算法得到的调度优化解显示:200 个计算集群 3 种异构处理器的资源利用率可分别接近 0.8,0.6 和 0.4,相应硬盘的资源利用率可分别接近 0.75,0.6 和 0.45,皆在相应的理论最优值范围内.

两组对比数据说明,GHSA\_di/II 算法具有考量两类密集型任务对硬盘存储硬件不同偏好的性能优势.这进一步显示,软硬件节能原理深度融合可给实时异构调度算法带来有效的绿色感知优化效果.

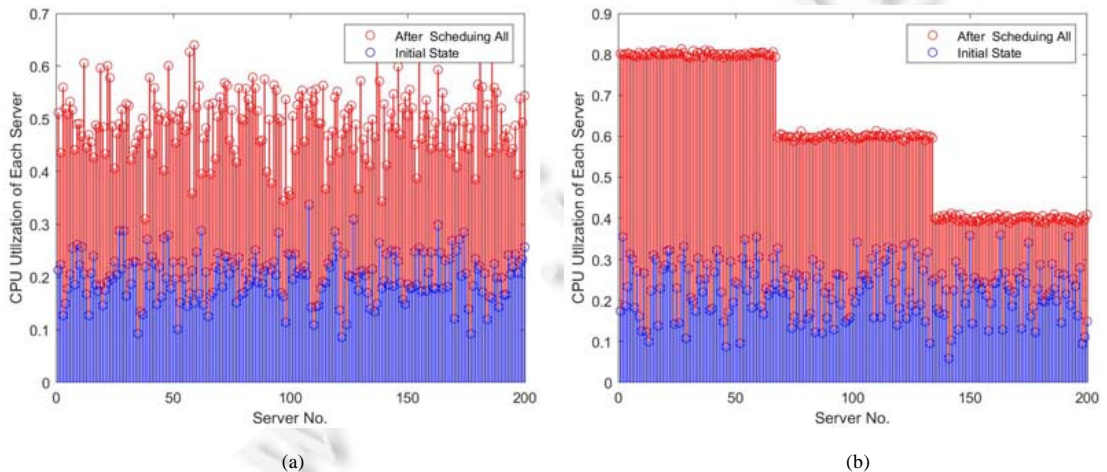


Fig.6 Comparison of CPU utilization after scheduling data intensive tasks by MaOEA/C<sup>[23]</sup> and GHSA\_di/II  
图6 MaOEA/C 算法<sup>[23]</sup>和 GHSA\_di/II 算法调度数据密集型任务后的处理器利用率对比

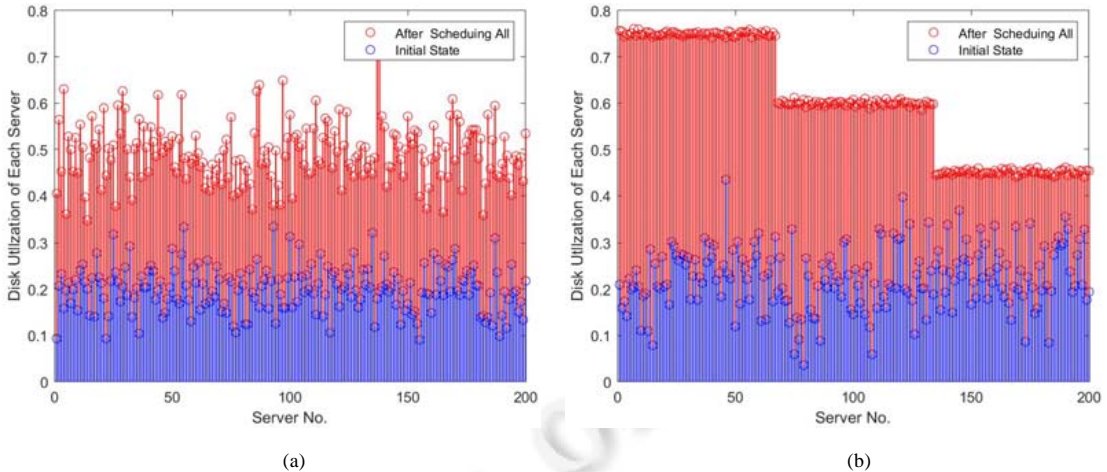


Fig.7 Comparison of disk utilization after scheduling data intensive tasks by MaOEA/C<sup>[23]</sup> and GHSA\_di/II  
图7 MaOEA/C 算法<sup>[23]</sup>和 GHSA\_di/II 算法调度数据密集型任务后的硬盘利用率对比

## 4 结 论

计算云高性能向高效能演进,是人类可持续发展迫切需求;同时,减少广域计算成本、降低密集应用开销、兼顾云服务买卖双方利益并保证双层负载均衡性,也是基础设施运维商增强竞争优势所在.随着新常态(云计算、异构计算)、新应用(计算密集型应用、数据密集型应用)和 QoS 新指标(能效)的出现,绿色感知的异构实时调度研究具有重大的理论和应用价值.

针对现有元启发式算法为基础的实时调度算法大多存在的进化动力不足、个体多样性不够或收敛速度过慢等缺陷,本文着眼于软硬件节能原理的深度融合,提出一种新的绿色异构调度算法.

大量仿真实验结果显示:无论对于数据密集型还是计算密集型实例,GHSA\_di/II 算法较其他 3 种元启发式异构调度算法,在整体性能、节能降耗以及可扩展性等方面,都具明显优势.

概言之,本研究是文献[11]的后续工作,而无论辐射广度、攻关难度、理论深度还是创新高度,本文的 GHSA\_di/II 算法都是其进一步的拓展和延伸,并可为推进云模式中软、硬件节能联动以提高其衔接实效探索新的道路,或奠定一定的理论和技术基础.

## References:

- [1] Belkhir L, Elmeligi A. Assessing ICT global emissions footprint: Trends to 2040 & recommendations. *Journal of Cleaner Production*, 2018,177:448–463.
- [2] Ganji M, Kazemipoor H, Molana SMH, *et al.* A green multi-objective integrated scheduling of production and distribution with heterogeneous fleet vehicle routing and time windows. *Journal of Cleaner Production*, 2020,259:120824.
- [3] Constantinescu DA, Navarro A, Corbera F, *et al.* Efficiency and productivity for decision making on low-power heterogeneous CPU+GPU SoCs. *The Journal of Supercomputing*, <https://doi.org/10.1007/s11227-020-03257-3>
- [4] Nejat M, Manivannan M, Pericàs M, *et al.* Coordinated management of DVFS and cache partitioning under QoS constraints to save energy in multi-core systems. *Journal of Parallel and Distributed Computing*, 2020,144:246–259.
- [5] Asghari A, Sohrabi MK, Yaghmaee F. Task scheduling, resource provisioning, and load balancing on scientific workflows using parallel SARSA reinforcement learning agents and genetic algorithm. *The Journal of Supercomputing*, <https://doi.org/10.1007/s11227-020-03364-1>
- [6] Saroja S, Revathi T. Hybrid dual-objective parallel genetic algorithm for heterogeneous multiprocessor scheduling. *Cluster Computing—The Journal of Networks Software Tools and Applications*, 2020,23:441–450. <https://doi.org/10.1007/s10586-019-02934-0>

- [7] Ahmed U, Lin JCW, Srivastava G, *et al.* A load balance multi-scheduling model for OpenCL kernel tasks in an integrated cluster. *Soft Computing*, <https://doi.org/10.1007/s00500-020-05152-8>
- [8] Amulu LM, Ramraj R. Combinatorial meta-heuristics approaches for DVFS-enabled green clouds. *The Journal of Supercomputing*, 2020,76:5825–5834.
- [9] Chen XJ, Shi C, Zhou AM, *et al.* Multiobjective evolutionary algorithm based on hybrid individual selection mechanism. *Ruan Jian Xue Bao/Journal of Software*, 2019,30(12):3651–3664 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5602.htm> [doi: 10.13328/j.cnki.jos.005602]
- [10] Zheng JH, Dong NJ, Ruan G, *et al.* High-dimensional multi-objective optimization strategy based on decision space oriented search. *Ruan Jian Xue Bao/Journal of Software*, 2019,30(9):2686–2704 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5842.htm> [doi: 10.13328/j.cnki.jos.005842]
- [11] Wang JL, Gong B, Liu H, *et al.* Model and algorithm of energy-efficiency aware scheduling for green heterogeneous computing. *Ruan Jian Xue Bao/Journal of Software*, 2016,27(9):2414–2425 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4849.htm> [doi: 10.13328/j.cnki.jos.004849]
- [12] Liang B, Dong XS, Wang YF, *et al.* A low-power task scheduling algorithm for heterogeneous cloud computing. *The Journal of Supercomputing*, 2020,76:7290–7314.
- [13] Chen WH, Xie GQ, Li RF, *et al.* Execution cost minimization scheduling algorithms for deadlineconstrained parallel applications on heterogeneous clouds. *Cluster Computing—The Journal of Networks Software Tools and Applications*, <https://doi.org/10.1007/s10586-020-03151-w>
- [14] Muhuri PK, Biswas SK. Bayesian optimization algorithm for multi-objective scheduling of time and precedence constrained tasks in heterogeneous multiprocessor systems. *Applied Soft Computing Journal*, 2020,92:106274.
- [15] Nik SSM, Naghibzadeh M, Sedaghat Y. Cost-driven workflow scheduling on the cloud with deadline and reliability constraints. *Computing*, 2020,102:477–500. <https://doi.org/10.1007/s00607-019-00740-5>
- [16] Abazari F, Analoui M, Takabi H, *et al.* MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm. *Simulation Modelling Practice and Theory*, 2019,93:119–132.
- [17] Rizvi N, Ramesh D. HBDCWS: Heuristic-based budget and deadline constrained workflow scheduling approach for heterogeneous clouds. *Soft Computing*, <https://doi.org/10.1007/s00500-020-05127-9>
- [18] Pandey V, Saini P. A heuristic method towards deadline-aware energy-efficient mapreduce scheduling problem in Hadoop YARN. *Cluster Computing—The Journal of Networks Software Tools and Applications*, <https://doi.org/10.1007/s10586-020-03146-7>
- [19] Taheri G, Khonsari A, Entezari-Maleki R, *et al.* A hybrid algorithm for task scheduling on heterogeneous multiprocessor embedded systems. *Applied Soft Computing Journal*, 2020,91:106202.
- [20] Aziza H, Krichen S. A hybrid genetic algorithm for scientific workflow scheduling in cloud environment. *Neural Computing and Applications*, <https://doi.org/10.1007/s00521-020-04878-8>
- [21] Xie Y, Wu JZ. Multi-objective constraint task scheduling algorithm for multi-core processors. *Cluster Computing—The Journal of Networks Software Tools and Applications*, 2019,22:953–964.
- [22] He ZZ, Zhou JZ, Qin H, *et al.* Long-term joint scheduling of hydropower station group in the upper reaches of the Yangtze River using partition parameter adaptation differential evolution. *Engineering Applications of Artificial Intelligence*, 2019, 81:1–13.
- [23] Song YJ, Ma X, Li XJ, *et al.* Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem. *Swarm and Evolutionary Computation*, 2019,49:194–205.

#### 附中文参考文献:

- [9] 陈晓纪,石川,周爱民,等.混合个体选择机制的多目标进化算法.软件学报,2019,30(12):3651–3664. <http://www.jos.org.cn/1000-9825/5602.htm> [doi: 10.13328/j.cnki.jos.005602]
- [10] 郑金华,董南江,阮干,等.决策空间定向搜索的高维多目标优化策略.软件学报,2019,30(9):2686–2704. <http://www.jos.org.cn/1000-9825/5842.htm> [doi: 10.13328/j.cnki.jos.005842]
- [11] 王静莲,龚斌,刘弘,等.支持绿色异构计算的能效感知调度模型与算法.软件学报,2016,27(9):2414–2425. <http://www.jos.org.cn/1000-9825/4849.htm> [doi: 10.13328/j.cnki.jos.004849]



王静莲(1979—),女,博士,副教授,主要研究领域为异构集群体系,云调度中间件与分布式人工智能等多学科交叉方向.



龚斌(1964—),男,博士,教授,博士生导师,主要研究领域为异构集群体系,云调度中间件与分布式人工智能等多学科交叉方向.



刘弘(1955—),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为群体智能算法,协同计算机多 agent 系统的理论与应用.



李少辉(1978—),男,讲师,主要研究领域为异构集群体系,云调度中间件与布式人工智能等多学科交叉方向.

www.jos.org.cn

www.jos.org.cn