文章编号:2095-6134(2021)05-0696-06

基于 EDA 仿真软件的多资源调度算法*

王静[1,2],陈岚[1†],张贺[1],王海永[1]

(1 中国科学院微电子研究所 三维及纳米集成电路设计自动化技术北京市重点实验室,北京 100029;2 中国科学院大学,北京 100049)

(2019 年 12 月 12 日收稿; 2020 年 1 月 21 日收修改稿)

Wang J, Chen L, Zhang H, et al. Multi-resource scheduling algorithm based on EDA simulation software[J]. Journal of University of Chinese Academy of Sciences, 2021, 38(5): 696-701.

摘 要 为提高电子设计自动化(electronic design automation,EDA)并行仿真任务的资源利用率并保证公平性,在占优资源公平分配机制(dominant resource fariness,DRF)的基础上,提出考虑 license 的占优资源公平分配(dominant resource fairness allocation algorithm considering license,LDRF)算法。一方面考虑多类型资源的公平调度问题,满足 EDA 任务对资源种类多样性的需求;另一方面考虑对 EDA 工具的 license 资源调度,避免任务占有硬件资源但是没有获得 license 授权不能运行,导致资源利用率下降。实验仿真结果表明,在 license 有限的条件下,LDRF 的平均 CPU 资源利用率比 DRF 算法提高 60%,平均内存资源利用率比 DRF 算法提高 34%。

关键词 资源分配;DRF 算法;公平性;EDA 软件;license 调度

中图分类号:TP301.6 **文献标志码**:A **doi**:10.7523/j.issn.2095-6134.2021.05.014

Multi-resource scheduling algorithm based on EDA simulation software

WANG Jing[1,2], CHEN Lan[1], ZHANG He[1], WANG Haiyong[1]

(1 *Beijing Key Laboratory of Three-dimensional and Nanometer Integrated Circuit Design Automation Technology, Institute of Microelectronics, Chinese Academy of Sciences, Beijing* 100029, *China*; 2 *University of Chinese Academy of Sciences, Beijing* 100049, *China*)

Abstract In order to improve the resource utilization of electronic design automation (EDA) parallel simulation tasks and ensure fairness, on the basis of dominant resource fairness allocation (DRF), dominant resource fairness allocation algorithm considering license (LDRF) is proposed. On the one hand, the problem of fair scheduling of multiple types of resources is considered, satisfying the diversity of resource requirements of EDA tasks. On the other hand, the license resource scheduling of EDA tool is considered to avoid the reduction of utilization caused by the lack of license for hardware resources. The experimental simulation results show that under the condition of limited license, the CPU resource utilization of LDRF is increased by 60% compared with DRF, and the memory resource utilization rate is increased by 34%.

Keywords resource allocation; DRF algorithm; fairness; EDA software; license scheduling

* 国家重点研发计划高性能计算专项(2017YFB0203501)和北京市科技新星与领军人才专项(Z171100001117147)资助

† 通信作者,E-mail:chenlan@ime.ac.cn

随着集成电路的发展进入到纳米级,电子设计自动化(electronic design automation, EDA)工具对 CPU 和内存等资源的要求越来越高。传统的单机平台已不能满足 EDA 仿真任务的高复杂性、高密度计算需求[1]。机器学习和大数据的发展也促进开发人员研究 EDA 工具并行仿真算法及架构[2-4],减少集成电路设计周期,改善用户体验。需要考虑将 EDA 并行仿真任务放在高性能集群上,借用集群上的资源快速完成仿真。因此,本文研究适用于 EDA 并行仿真任务的 LDRF(dominant resource fairness allocation algorithm considering license)调度算法。

单一资源调度已研究成熟,比如 max-min fairness,核心思想是在多用户的前提下,最大化每个用户的最小资源需求,已经在很多工程中得到应用,比如网络队列中的 round-robin 算法。而 EDA 仿真任务对资源需求具有多样性,可分为 CPU 密集型、内存密集型及 IO 密集型等,考虑多种资源公平分配时,占优资源公平分配(dominant resource fairness, DRF)[5]是基于“主导份额”的 max-min fairness 公平调度算法,已经得到很多研究及改进。如文献[6]提出异构环境下的占优资源公平分配(DRF in heterogeneous cloud,DRFH)算法,将 DRF 运用于异构云系统;文献[7]提出用户任务对资源的需求随时间变化的算法;文献[8]在 DRF 思想基础上提出动态情况下的分配算法即动态占优资源公平分配机制(dynamic dominant resource fairness mechanism, DDRF)。

综上所述,多资源调度的研究已经取得一定进展[9-13]。由于 EDA 任务必须获得 license 授权才能执行,而 license 比较昂贵且数量有限,如全流程 IC 设计所包含的全套 EDA 工具 license 一年需要花费上千万元,针对 EDA 并行仿真任务,需要将多资源调度和 license 调度结合,避免任务获得所需的计算资源、存储资源等,但却因未获得 license 授权不能运行,导致资源浪费。常用的调度算法没有考虑 license 调度,仅适用于 map-reduce 等计算任务。因此,本文提出将 license 调度和多资源调度结合的基于 EDA 并行仿真任务的 LDRF 调度算法。

1　主流公平调度算法

1.1　DRF 调度算法

DRF 的核心思想是根据每个计算任务的资源需求向量和系统总资源向量,得到各个计算任务的主导份额。通过平衡各个计算任务的主导份额,可以确定每个计算任务的子任务数及最终分配的资源向量。文献[5]已证明 DRF 具有 4 个性质:

1)共享性:不同用户的任务是通过共享资源而不是独占资源的形式来提高资源利用率,每个用户都能均衡占用资源。2)真实性:系统中任何谎报资源的用户将不会得到更多的资源。3)非抢占性[14]:任何任务都不能在获得计算资源后,通过已有的资源,去获得(或交换)另一个任务的资源。4)帕累托效率性[15]:集群中的所有计算任务都不能够在不减少其他任务资源拥有量的前提下增加自己的资源拥有量。文献[16-17]证明满足这 4 个性质的调度算法具有公平性。该算法的描述如下:

假设系统存在 2 种资源 r_1 和 r_2,如 CPU、内存。资源总量为 R_1 和 R_2,系统存在 2 个用户 i 和 j,资源需求向量分别为 $\boldsymbol{D}_i = (d_{i,r1}, d_{i,r2})$,$\boldsymbol{D}_j = (d_{j,r1}, d_{j,r2})$。如果满足 $d_{i,r_1}/R_1 > d_{i,r_2}/R_2$,$d_{j,r_1}/R_1 < d_{j,r_2}/R_2$,则用户 i 的占优资源为 r_1,用户 j 的占优资源为 r_2,假设 x_i,x_j 分别为 i,j 的子任务数,则满足下列约束条件:

$$\begin{cases} x_i \times d_{i,r_1} + x_j \times d_{j,r_1} \leqslant R_1, \\ x_i \times d_{i,r_2} + x_j \times d_{j,r_2} \leqslant R_2, \\ x_j \times d_{j,r_2}/R_2 = x_i \times d_{i,r_1}/R_1. \end{cases} \tag{1}$$

由公式(1)可知,用户最终分配的子任务数由占优资源决定。

1.2　license 管理及调度

常见的 EDA 工具 license[18]大部分基于浮动 license 进行授权管理,即 license 不与节点绑定,用户只要获得 license 授权便可在任意节点使用。调度系统通过心跳反应与 license 管理工具交互,确保即将分配资源的任务必须获得 license 授权。

license 常用调度算法为:先来先服务算法,将任务按照到达时间放入队列,如果先提交的任务 license 不能满足则考虑后面任务;公平分配算法,对于需要相同 license 的任务平均分配 license,如果存在任务 license 分配超额则可暂时分给别的任务;轮转调度算法,如果几个用户的任务同时需要一种或者几种 license,且 license 有限,为减少用户等待时间,可通过时间片轮询将任务挂起,并将 licesne 分给别的任务。

2 LDRF 算法

LDRF 算法使用 DRF 最大化最小占优资源的思想并加以改进,将多资源调度和 license 调度结合。DRF 算法假设用户分配的子任务数量是无限的,而几个任务消耗集群所有资源是不符合实际要求的。EDA 并行任务数是根据任务种类、规模及客户需求等确定的,并且当并行任务数增加到一定程度时加速比会下降,因此用户分配的任务数应该有约束。

由于 EDA 工具的 license 分配以核为单位,比如 Cadence 公司的 EDA 软件 Calibre DRC、LVS、XRC 及 DFM 等都以 CPU 核数为单位分配所需 license 数,其 license 数目及 CPU 核数的节省比例为 1 个 license 支持 1 核 CPU,2 个 license 支持 4 核 CPU,3 个 license 支持 8 核 CPU 等。因此,LDRF 算法计算资源将以 CPU 核数为单位,而不是 DRF 中的以 CPU 数量为单位。其次,真实环境中,可能存在紧急任务,因此需要根据重要程度给每个用户添加权重,确保调度的公平性。LDRF 算法具体阐述如下:

集群资源种类数为 m,包括 CPU 核数、内存、磁盘及 IO 等,用户数为 n,license 资源种类数为 s。集群中可分配的资源向量为 $\boldsymbol{r}=(r_1,r_2,\cdots,r_m)$,可分配的 license 向量为 $\boldsymbol{L}=(L_1,L_2,\cdots,L_s)$。用户 i 一个子任务的资源需求向量为 $\boldsymbol{d_i}=(d_{i1},d_{i2},\cdots,d_{im})$,$d_{ij}$ 表示用户 i 对资源 j 的需求量。用户 i 一个子任务的 license 需求向量为 $\boldsymbol{l_i}=(l_{i1},l_{i2},\cdots,l_{is})$。用户 i 已经分配的子任务数为 x_i,初始值为 0。用户 i 的权重为 w_i,最多可分配的任务数为 m_i。则考虑权重时,用户 i 的占优资源份额定义为

$$\mu_i=\max_k\left\{\frac{x_i\times d_{ik}}{r_k}\right\}\Big/w_i,k=1,2,\cdots,m, \quad (2)$$

满足下列约束:

$$\begin{cases}\sum_{i=1}^{n}d_{ik}\times x_i\leqslant r_k,\ k=1,2,\cdots,m,\\ \mu_i\approx\mu_j,\ i=1,2,\cdots,n,i\neq j,\\ x_i\leqslant m_i,\ i=1,2,\cdots,n,\\ \sum_{i=1}^{n}x_i\times l_{iq}\leqslant L_q,\ q=1,2,\cdots,s.\end{cases} \quad (3)$$

实际资源分配并不是直接根据式(2)、式(3)计算最优结果,每个用户的占优资源份额不是绝对相等,而是在获得 license 授权的条件下趋于平衡。其次,任务执行完成后会释放资源,空闲资源发生变化,因此实际中是以式(3)为约束条件,通过循环方式分配给任务资源。每次循环选择一个任务分配资源,不存在其他任务竞争。因此,license 调度使用的是 FIFO(first input first output)算法。因为调度过程中优先级是根据占优资源份额来确定的,初始份额均为 0,当计算资源充足的情况下,初始任务随机选择不影响结果。但是当计算资源有限时,为更加公平地调度,任务应该有初始优先级 p_{i0},与任务大小成反比,与用户权重成正比,如下所示:

$$p_{i0}=w_i\Big/\left(\sum_{k=1}^{m}\frac{d_{ik}}{r_k}\right). \quad (4)$$

LDRF 算法的步骤如算法 1 所示。

算法 1 LDRF 算法

$\boldsymbol{r}=(r_1,r_2,\cdots,r_m)$	集群总的资源向量
$\boldsymbol{L}=(L_1,L_2,\cdots,L_s)$	集群总的 license 向量
$\boldsymbol{c}=(c_1,c_2,\cdots,c_m)$	集群已经使用的资源
$\boldsymbol{d_i}=(d_{i1},d_{i2},\cdots,d_{im})$	用户 i 一个子任务所需资源向量
$\boldsymbol{l_i}=(l_{i1},l_{i2},\cdots,l_{is})$	用户 i 一个子任务所需 license 向量
x_i	用户 i 已分配的任务数
m_i	用户 i 最多分配的任务数
w_i	用户 i 的权重

步骤:

1. 根据公式(4)计算每个用户 i 的初始优先级 p_{i0},从大到小排序并存入队列 P;
2. for p_i in P//依次选择初始优先级高的用户 i;
3. while $r_j-d_{ij}>c_j,j=1,2,\cdots,m$
4. $c_j+=d_{ij},j=1,2,\cdots,m;x_i++$
5. end while
6. end for
7. while $r_j-c_j>0,j=1,2,\cdots,m$
8. 根据公式计算(2)计算每个用户的占优资源 μ_i 并排序;
9. 选择 μ_i 最小的用户 i;
10. if $r_j-c_j>d_{ij},j=1,2,\cdots,m$ and $l_{ik}<L_k,k=1,2,\cdots,s$ and $x_i<m_i$ then $c_j+=d_{ij},j=1,2,\cdots,m;L_k-=l_{ik},k=1,2,\cdots,s;x_i++$; break
11. else 根据 9 选择占优资源 μ_i 次小的用户,循环执行 10;
12. end if
13. end while
14. return 资源分配方案。

3 LDRF 性能分析

3.1 资源利用率分析

1) LDRF 和 DRF 资源利用率比较

LDRF 以 license 资源调度为前提,如果任务

缺乏 license 则不分配计算资源,这部分资源可分配给别的任务,保证资源充分利用。而 DRF 没有考虑 license 调度,调度系统可能给任务分配资源但缺乏 license 授权,任务已分配充足的资源却不能执行,导致资源利用率下降。

图 1 为不同用户数条件下,多次随机产生每个用户所需资源向量及集群总资源向量时,2 种算法 CPU 资源、内存资源平均占用情况直观比较图。其中图 1(a)、1(c)为 license 数量有限的情况,图 1(b)、1(d)为 license 数量比较充足的情况。由此可知,license 有限的条件下,LDRF 资源利用率明显高于 DRF 资源利用率,CPU 核数平均资源利用率增长 60%,内存平均资源利用率增长 34%。license 比较充足时,CPU 核数平均资源利用率增长 28%,内存平均资源利用率增长 10%。

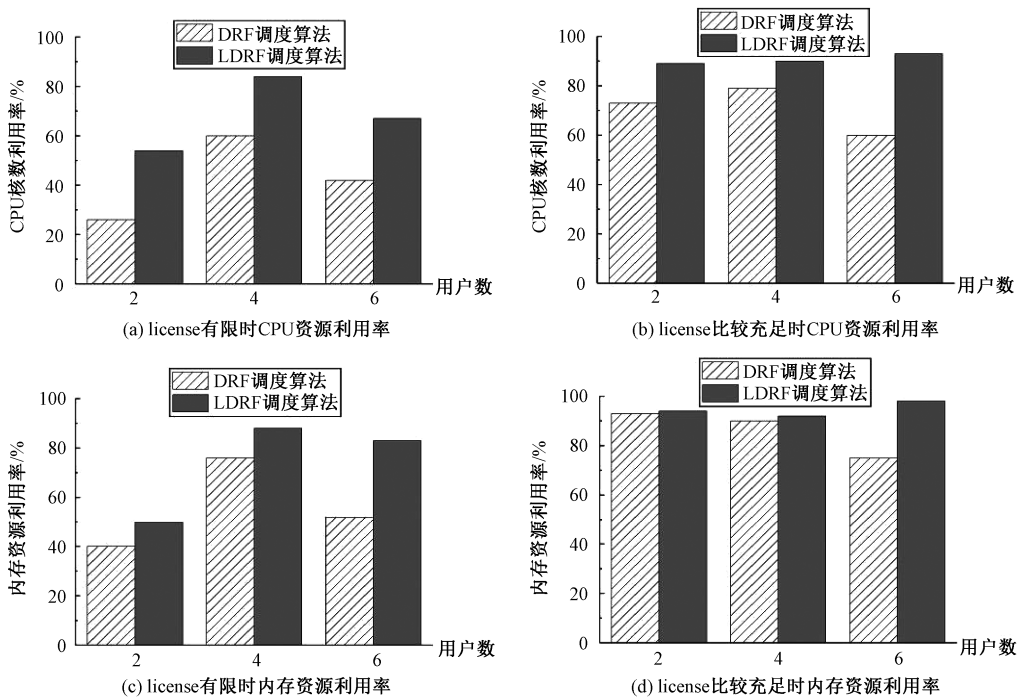

图 1　DRF、LDRF 算法资源利用率

Fig. 1　DRF and LDRF algorithm resource utilization

2) LDRF、FIFO、CPU fair 算法比较

常用资源调度算法有 FIFO 和 max-min fairness 调度算法。FIFO 即先到达任务先得资源,依次执行,其他任务只能处于等待状态。max-min fairness 调度算法即最大化每个任务所需最小资源。由于 max-min fairness 只针对单一资源调度,因此在多资源仿真中以 CPU 为标准即 CPU fair。另外在此基础上,将 2 种算法添加 license 调度以适用于 EDA 仿真任务。图 2 是在不同用户数条件下,多次随机产生所需输入资源向量及总的空闲资源向量,3 种算法平均资源利用率的比较图。由图可知,LDRF 算法的 CPU 资源利用率及内存资源利用率明显优于其他算法。

3) 实测数据性能分析

前述分析已经展示出 LDRF 算法资源利用率最高。为使性能对比更具说服力,将 2 个用户的并行 DFM 分析 EDA 仿真任务在高性能集群上测试。集群上有 8 个节点,每个节点 24 个 CPU 核。图 3 为使用 LDRF 算法和 FIFO 调度算法时,执行任务所对应的平均资源占用情况对比图。

由图 3 可知,使用 LDRF 算法时,CPU 资源利用率及内存资源利用率均最优。其次,使用 LDRF 算法时 2 个用户执行完任务所需时间约为 650 s,使用 FIFO 算法时任务执行时间约为 800 s,执行效率提升 23%。

3.2　公平性分析

LDRF 算法借鉴 DRF 算法最大化最小占优资源的思想并加以改进,满足共享性等 4 个衡量公平性的指标。从算法过程来看,多重优先级的设定如初始优先级和占优资源份额,以及多轮分配资源的方法保证了每个用户都能得到资源,满足共享性。如果用户谎报资源超额得到分配,下一

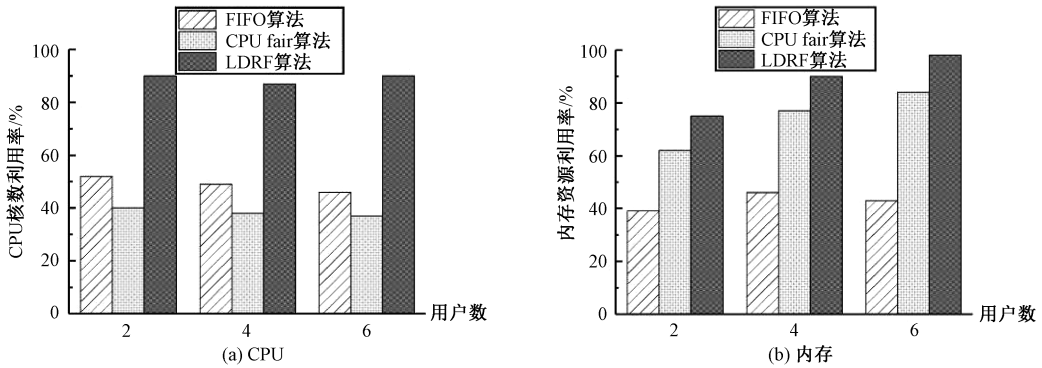

图 2 FIFO、CPU fair 和 LDRF 算法资源利用率

Fig. 2 FIFO, CPU fair, and LDRF algorithm resource utilization

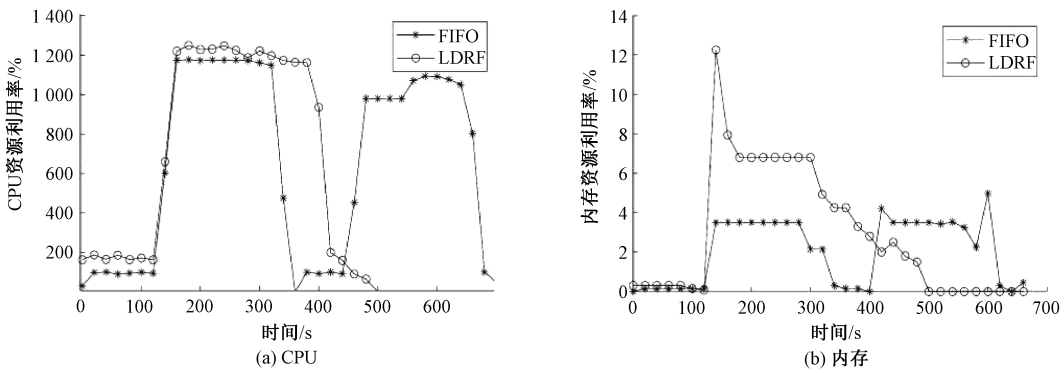

图 3 FIFO、LDRF 实测数据资源利用率

Fig. 3 FIFO and LDRF measured data resource utilization

轮分配会减少用户资源的分配,最终满足主导资源均衡性要求,并且由于资源按需计价收费,谎报资源需求会增加花费,满足真实性及非抢占性。从算法整体来讲,每个用户均共享资源池中的资源,增加一个用户分配的资源时,必然会减少其他用户分配的资源,满足帕累托效率性。其次,LDRF 是以 license 调度为前提的多资源调度算法,可以防止用户的任务占据硬件资源但缺乏 license 无法执行,保证了用户公平性。因此,LDRF 算法可以保证公平性资源分配。

4 总结

考虑到 EDA 工具 license 昂贵且稀缺、EDA 并行任务的子任务数有限制、每个用户的权重不同、初始优先级以及不同 EDA 仿真任务可能有不同的占优资源,本文研究适用于 EDA 并行仿真任务的多资源调度 LDRF 算法,提高了 EDA 并行任务的执行效率和资源利用率,并保证了调度的公平性。由于 EDA 仿真过程中步骤繁琐,复杂性高,任务之间可能有依赖性,下一步工作将是基于 EDA 多任务流算法的研究。保证有依赖关系的任务在 license 数量充足的前提下依次执行。

参考文献

[1] Stok, Leon. The next 25 years in EDA: a cloudy future? [J]. IEEE Design & Test, 2014, 31(2): 40-46.

[2] 刘军华, 杨海钢. 一种快速并行协同仿真的验证方法[J]. 微电子学, 2008, 38(4): 66-69,89.

[3] 王超, 陈香兰, 周学海, 等. 异构多核平台上基于任务划分和调度的性能评估方法[J]. 中国科学院研究生院学报, 2012, 29(2): 257-263.

[4] 常天海, 胡鉴. 基于 FPGA 的 CRC 并行算法研究与实现[J]. 微处理机, 2010, 31(2): 47-50.

[5] Ghodsi A, Zaharia M, Hindman B, et al. Dominant resource fairness: fair allocation of multiple resource types [C] // Proceedings of the 8th USENIX Conference on NSDI. Berkeley: USENIX Association, 2011: 24-37.

[6] Wang W, Liang B, Li B. Multi-resource fair allocation in heterogeneous cloud computing systems [J]. IEEE Transactions on Parallel & Distributed Systems, 2015, 26(10): 2822-2835.

[7] 李杰, 张静, 李伟东, 等. 一种基于共享公平和时变资源需求的公平分配策略[J]. 计算机研究与发展, 2019, 56(7): 1534-1544.

[8] Li W, Liu X, Zhang X, et al. A further analysis of the

dynamic dominant resource fairness mechanism [C] // Proceedings of the 11th International Conference on Frontiers in Algorithmics. Berlin: Springer, 2017: 163-174.

[9] 马宏星, 周学海, 高妍妍, 等. 一种集成可重构硬件的多核片上系统的软硬件任务划分与调度算法[J]. 中国科学院研究生院学报, 2010, 27(5): 664-669.

[10] Doulamis N, Varvarigos E, Varvarigou T. Fair scheduling algorithms in grids [J]. IEEE Transactions on Parallel & Distributed Systems, 2007, 18(11): 1630-1648.

[11] 柯尊旺, 于炯, 廖彬. 适应异构集群的 Mesos 多资源调度 DRF 增强算法[J]. 计算机应用, 2016, 36(5): 1216-1221.

[12] 卢笛, 马建峰, 王一川, 等. 云计算下保障公平性的多资源分配算法[J]. 西安电子科技大学学报, 2014, 41(3): 162-168.

[13] 刘晓茜, 杨寿保, 郭磊涛, 等. 网格市场中基于成本计算的任务调度研究[J]. 中国科学院研究生院学报, 2008, 25(3): 379-385.

[14] Bouveret S, Lang J. Efficiency and envy-freeness in fair division of indivisible goods: logical representation and complexity[J]. Journal of Artificial Intelligence Research, 2008, 32: 525-564.

[15] 洪雁, 何晓林. 基于帕累托最优的公平性探讨[J]. 科技创业月刊, 2006, 19(11): 175-176.

[16] Lan T, Kao D, Chiang M, et al. An axiomatic theory of fairness in network resource allocation[C]//proceedings of the IEEE INFOCOM. Piscataway: IEEE, 2010: 1-9.

[17] Kuenne R E. Equity: in theory and practice[J]. Southern Economic Journal, 1995, 61(3): 885-886.

[18] 王寅峰, 董小社, 郭华, 等. 网格环境中软件共享系统的 License 管理器[J]. 华中科技大学学报(自然科学版), 2006, 34(s1): 5-8.