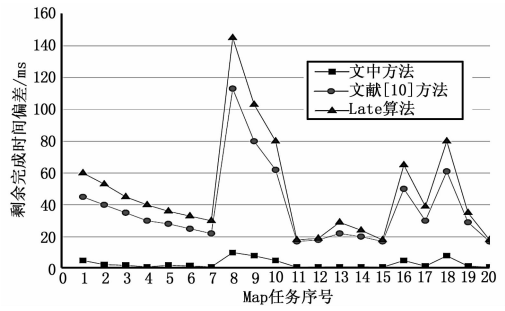

图 5　Map 剩余完成时间偏差

而文献［12］方法和经典 Late 算法的最小偏差均大于 18 ms，且 Late 方法的最大偏差超过 142 ms，很显然，文中调度方法中的预测能力更优。

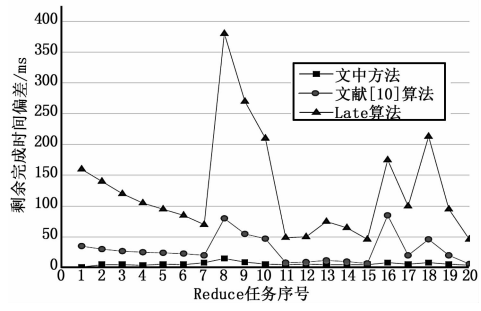

图 6　Reduce 任务剩余完成时间偏差

从图 6 中可以看出，文中方法预测的 Reduce 任务剩余完成时间与真实值的误差最小，Late 算法和文献［12］方法的误差均较大，这是因为文中方法采用基于历史信息的预测方法，对 Reduce 任务的 3 个阶段根据历史信息预测不同的进度值，因此，文中方法的预测更为准确，能有效地实现慢任务迁移，提高了作业调度的效率和减少了所有任务的最大完成时间。

5　结语

作业调度是云计算环境下 Hadoop 集群研究的核心问题，为此，文中研究了一种基于云计算和 Map－Reduce 编程模型的作业调度方法，首先介绍了 Hadoop 框架模型和 Map－Reduce 编程和调度原理，然后，在经典的 Late 作业调度算法的基础上，对其 Map 任务和 Reduce 任务的各阶段所需时间比例进行基于历史信息的预测，同时利用数据局部性尽量对任务进行本地化实现。实验结果表明文中算法能有效克服 Late 调度方法的不足，其预测的任务剩余完成时间更真实，是一种适合异构 Hadoop 集群作业调度的算法。

参考文献：

[1] Vaquero L，Rodero Marino L，Cacerce J，et al. A break in the clouds：towards a cloud definition［J］，SIGCOMM Computer Communication Review，2009，39（1）：50－55.

[2] 林伟伟，齐德昱. 云计算资源调度研究综述［J］. 计算机科学，2012，39（10）：1－6.

[3] Shvachko K，Kuang Hairong，Rania S，et al. The Hadoop distributed file System［A］. Proc of Mass Storage Systems and Technologies［C］. 2010：1－10.

[4] Apache Hadoop. Hadoop［EB/OL］. F2009－03－061 httpt／／hadoop. apache. org/.

[5] Yahoo. YahooI Hadoop Tutorial［EB/OL］［2009－02－27］. http：//publia. yahoo. Com/gogate/hadoop－tutorial/start－tutofiaLhtmL.

[6] Dean J，Ghemawat S. MapReduce：a flexible data processing tcol［J］. Communications of the ACM，2010，53（1）：72－77.

[7] PARK J. Locality－aware dynamic VM reconfiguration on MapReduce clouds［A］. Proc of the 21 st International Symposium on Hilgh Performance Parallel and Distributed Computing［C］. 2012：27－36.

[8] Zaharia M. Job scheduling for multi－user MapReduce clusters，Tech. Rep. UCB/eecs－2009－55［R］. Berkeley：EECS Department，University of California，2009.

[9] 刘亚秋，邢乐乐，景维鹏. 云计算环境下基于时间期限和预算的调度算法［J］. 计算机工程，2013，6（39）：56－58.

[10] 林伟伟，刘　波. 基于动态带宽分配的 Hadoop 数据负载均衡方法［J］. 华南理工大学学报（自然科学版），2012，9（40）：42－47.

[11] 柯何杨，杨　群，王立松，等. 同构 Hadoop 集群环境下改进的延迟调度算法［J］. 计算机应用研究，2013，5（30）：1397－1301.

[12] 魏晓辉，付庆午，李洪亮. Hadoop 平台下基于资源预测的 Delay 调度算法［J］. 吉林大学学报（理学版）2013，1（51）：101－106.

[13] 李玉林，董　晶. 基于 Hadoop 的 MapReduce 模型的研究与改进［J］. 计算机工程与设计，2012，8（33）：3110－3116.

文章编号:1671-4598(2014)06-1850-03　　中图分类号:TP393　　文献标识码:A

基于移动 Agent 和混合蛙跳算法的网格计算资源管理

潘　赟，何　勇
（信阳农林学院 计算机科学系，河南 信阳　464000）

摘要：为了克服传统网格系统采用Globus集中式模型或P2P网络分布式模型具有的无法兼顾全局控制性和扩展性的问题，提出了一种基于移动 Agent 和混合蛙跳算法的网格资源管理模型；首先，引入移动 Agent 将资源管理模型划分为应用管理模块、资源管理模块和资源表示模块，然后通过在经典的混合蛙跳算法加入差分扰动设计了一种改进的混合蛙跳算法，最后，在资源管理模块中的资源分配 Agent 中运行改进的混合蛙跳算法实现用户任务到资源的调度；采用 GridSim 工具进行仿真，实验结果表明，文中的资源管理模型具有较强的全局控制能力和扩展性能，资源利用率高达 95.65%，较其它方法具有较小的时间跨度和较高的资源利用率，是一种适用于网格环境的有效资源管理模型。

关键词：网格；资源管理；混合蛙跳算法；移动 Agent

Grid Resource Management Based on Mobile Agent and Shuffle Frog Leaping Algorithm

Pan Yun，He Yong
（Department of Computer Science，Xinyang College of Agriculture and Forestry，Xinyang　464000，China）

Abstract：In order to conquer the traditional grid system using the Globus concentration model or P2P model can not give consideration of both global control performance and extension ability，a grid resource management model based on mobile agent and shuffled frog leaping algorithm was proposed. Firstly，the resource model was divided to application management model，resource management model and resource representation model，then an improved algorism based on the classic shuffled frog leaping algorithm was proposed by adding the difference disturbance. Finally，the resource allocation Agent in the resource management model was operating the improved shuffled frog leaping algorithm to realize the task scheduling to the resource. The GridSim tool was used to simulate the method in this paper，and the result shows our method can realize the resource scheduling and the resource usage rate is as high as 95.65%，and compared with other methods，it has the least makespan and highest resource usage ratio，so it is an effective model in the grid environment.

Key words：grid；resource management；shuffle frog leaping algorithm；mobile agent

0　引言

Ian Foster将网格定义为[1-2]：通过网格技术能将地域上广泛分布的各类资源组合成一个有机整体。由于网格中物理节点的广域分布性，使得网格中的资源分布具有广泛性、异构性和分布性等特征[3]，因此，如何高效地实现网格资源管理、任务调度和性能监控是是网络系统需要解决的重要问题[4-5]。

目前已有的对资源进行管理和调度的主要工作有：文献[6]提出了一种动态的网格资源预留机制；文献[7]提出了一种新的基于组合双向拍卖机制的资源分配模型；文献[8]设计了一种基于最小代价函数的网格分层资源发现模型；文献[9]设计了一个基于改进蚁群算法的网格资源调度方法，定义了一个网格资源空间所需时间向量；文献[10]设计了一种基于信任机制的网格资源调度算法。

上述算法均研究网格资源管理[11]，具有重要的意义，但往往基于Globus的集中式模型或P2P网络的分布式模型，不能兼顾自适应性、扩展性和体控制性能。因此，本文提出了一种基于移动 Agent 和混合蛙跳算法的网格资源管理方法，并通过实验证明了文中方法的有效性。

1　基于 Agent 的网格资源管理模型

1.1　移动 Agent 技术

移动 Agent 是一段可以在不同设备进行传输的功能代码，特别适合具有大量数据传输和计算的大规模网络应用，Agent 具有移动性、感知性自主性、反应性、交互性和主动性等特征。

由于 Agent 具备上述特点，因此，在网格资源管理中引入移动 Agent 具有下列优势：

（1）移动 Agent 能从地域广泛分布的一个物理节点自主迁移到另一个节点，并与其它物理节点上的 Agent 进行交互，使得资源易于管理和发现；

（2）移动 Agent 移动时只携带功能代码和运算结果，不需要大量数据的传递，能有效降低带宽和提高任务调度的效率；

（3）移动 Agent 自主的在各个节点之间移动可以传递物理节点状态、负载以及网络状态等信息，为网格资源管理和任务调度提供信息依据；

（4）当一个节点创建多个移动 Agent 时，就可以将多个移动 Agent 迁移到多个节点上并行执行；

由于移动 Agent 具有上述优点，所以将其引入到网格资源管理模型中，设计一种基于移动 Agent 的资源管理模型。

1.2　基于移动 Agent 的资源管理模型

文中设计的基于移动 Agent 技术的资源管理模型可以分为

收稿日期：2013-11-21；　**修回日期**：2014-01-13。

作者简介：潘　赟（1982-），女，河南信阳人，硕士，讲师，主要从事网络工程、网络操作系统及网络编程方向的研究。

3 个模块：应用管理模块、资源管理模块和资源表示模块，如图 1 所示。

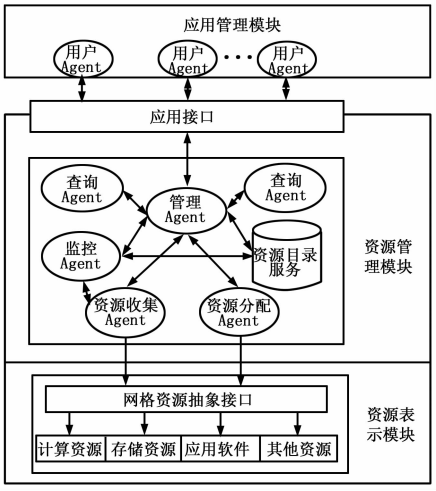

图 1　基于移动 Agent 的网格资源管理模型

在图 1 中，应用管理模块是模型的最上层，主要是为用户提供各类服务，用户可以通过该接口提交各类请求透明地实现对物理网格资源的访问，并在网络执行完后获得执行的返回结果。

资源管理模块是模型的核心，主要是由各类移动 Agent 组成，这些移动 Agent 的主要功能包括查询资源状态、监控节点状态、收集节点信息、对用户提交的任务进行节点资源分配和在资源目录中查询各节点的相关信息等，并为应用管理模块提供访问接口。

资源表示模块是模型的底层，其将各类物理资源如计算资源、存储资源、软件资源进行抽象表示，对这些资源进行统一地描述，并向资源管理模块提供访问硬件资源的接口。

2　基于移动 Agent 的资源调度数学模型

基于移动 Agent 的网格资源调度是网格管理的最关键部分，当用户应用程序被提交到应用管理模块后，应用管理模块将用户程序划分为相互独立的任务集 $T=\{t_1,t_2,\ldots,t_n\}$，然后将其转交给资源管理模块进行处理，在资源管理模块中，各种功能的移动 Agent 协调处理，在查询了各物理节点的各类信息后，通过资源分配 Agent 进行资源调度，在资源分配 Agent 处运行调度算法。

假设资源管理模块获取的可用资源集为 $P=\{p_1,p_2,\ldots,p_m\}$，则资源分配 Agent 的任务就是实现任务集 $T=\{t_1,t_2,\ldots,t_n\}$ 到资源集 $P=\{p_1,p_2,\ldots,p_m\}$ 的一一映射。

采用矩阵 $\boldsymbol{ET}_{n*m}$ 表示任务在资源上的预期执行时间，如下所示：

$$\boldsymbol{EF}=\begin{bmatrix} et_{11} & et_{12}\ldots & et_{1m} \\ \vdots & & \vdots \\ et_{n1} & et_{n2}\ldots & et_{mn} \end{bmatrix} \tag{1}$$

在式（1）中，et_{ij} 表示任务 t_i 在资源 p_j 上的预期执行时间，其可以通过任务的计算量除以资源的处理速度来获取。将其与所分配资源的最早可用时间 s_i 相加，可以得到所有任务调度到不同资源上的最早完成时间，如式（2）所示：

$$\boldsymbol{ETF}=\begin{bmatrix} etf_{11} & etf_{12}\ldots & etf_{1m} \\ \vdots & & \vdots \\ etf_{n1} & etf_{n2}\ldots & etf_{mn} \end{bmatrix} \tag{2}$$

其中：etf_{ij} 表示任务 t_i 在资源 p_j 上的最早完成时间。

移动 Agent 进行网格资源调度的目标就是要使式（2）中的最大值最小化，即：

$$G=\min(\max_{1\leqslant i\leqslant n,1\leqslant j\leqslant m} etf_{ij}) \tag{3}$$

3　基于移动 Agent 和混合蛙跳的调度算法

3.1　混合蛙跳算法

混合蛙跳算法[12]（Shuffle Frog Leaping Algorithm，SFLA）由 Eusuff 和 Lansey 于 2000 年通过模拟青蛙觅食过程而提出，其基本思想是将整个青蛙种群分为主种群和若干子种群进行觅食，通过主种群的全局信息交换和子种群的局部搜索实现个体之间信息的传递，从而获取全局最优解。由于混合蛙跳算法实现简单，收敛速度快，同时需要调节的参数较少，因此较粒子群算法和遗传算法等全局优化算法相比，具有收敛速度快和全局寻优能力强的优点。

3.2　青蛙编码和种群生成

每只青蛙可以编码为求解问题的一个可行解 $X_i=\{x_{i1},x_{i2},\ldots,x_{id}\}$，$d$ 为解空间维数，即用户任务总数，而编码的值 x_{ij} 则为调度的资源 ID，随机初始化种群为 $X=\{X_1,X_2,\ldots,X_N\}$，然后以式（3）的倒数为适应度函数，将其按适应度值的降序排序得到 X'，对其划分生成 M 个子种群。

子种群的划分规则为：X' 中的第一只青蛙分配到第 1 个子种群，第 2 只青蛙分配到第 2 个子种群，第 M 只青蛙分配到第 M 个子种群中，第 $M+1$ 只青蛙分配到第 1 个子种群中，依次类推，则子种群中的青蛙数量 m 为 N/M。

3.3　子种群的局部搜索

青蛙子种群的局部搜索即各子种群进行并行搜索，对于任意子种群 S_i，假设其适应度最好和最差的青蛙分别为为 X_b 和 X_w，则将该子种群中最差的个体更新为：

$$X'=X_w+R(X_b-X_w) \tag{4}$$

在式（4）中，R 为（0，1）之间的随机数。如果 X' 优于 X_w，则将 X_w 更新为 X'；否则随机产生一个解 X' 代替 X_w；

在子种群局部搜索的每次迭代过程中对青蛙的适应度值进行计算，并重复地对最坏适应度值的青蛙进行更新，直到迭代次数达到最大值。

3.4　主种群的全局搜索

各子种群在搜索完毕后合并成主种群，进行全局信息交换，然后对主种群中的个体按适应度从高到低进行排序，再重新按 2.3 划分子种群进行局部搜索。

3.5　差分扰动

经典的混合蛙跳算法在算法前期收敛速度慢，在算法后期容易陷入早熟收敛，针对此缺陷，在混合蛙跳算法中的子种群的局部搜索中加入差分向量，并通过随机扰动来增强种群的多样性，以提高算法的全局搜索能力。

在某子种群 S_i 中任意选择两个青蛙 X_i 和 X_j，其中 $X_i=\{x_{i1},x_{i2},\ldots,x_{id}\}$ 和 $X_j=\{x_{j1},x_{j2},\ldots,x_{jd}\}$，记差分向量为

$RX=\{Rx_1,Rx_2,\ldots,Rx_d\}=X_i-X_j$，当 RX 值越小，扰动越小。因此，将式（4）修改为：

$$X'=\begin{cases}R*(X_b-X_w)r\geqslant CR\\ \alpha*RX+R(X_b-X_w)r<CR\end{cases}\tag{5}$$

其中：CR 为扰动概率，α 和 r 均为服从［0，1］均匀分布的随机数，计算 X' 的适应度值：如果 X' 的适应度值优于 X_w 的适应度值，则采用 X' 代替 X_w；否则，采用 X_b 代替式（5）中的 X_w。

通过上述方法使得最差个体能向比自身更好的个体学习，从而保持群体的多样性，从而获取资源调度的全局最优解。

3.6 算法描述

在移动 Agent 上采用改进的混合蛙跳算法进行资源调度的算法可以描述为：

步骤 1：初始化主种群规模 N，子种群数为 M，子种群搜索迭代次数为 T_l，混合迭代次数为 T_g，扰动概率 CR；

步骤 2：计算主种群中的个体适应度，根据适应度从大到小将主种群中的青蛙分为 M 个子种群；

步骤 3：对每个子种群进行局部搜索，将其最优适应度青蛙和最差适应度青蛙分别标记为 X_b 和 X_w，并根据式（5）更新最差适应度的青蛙，直到达到子种群的最大迭代次数 T_l；

步骤 4：将各个子种群合并成新的主种群，并计算各青蛙的适应度，并按照适应度从高到低对青蛙进行排序，重新划分 M 个子种群，直到混合迭代次数达到最大值 T_g。

4 实验分析

采用网格计算仿真工具 GridSim 对实验进行仿真，网格节点个数为 40，任务数为 1 000～4 000，文中参数设置如下：种群规模 $N=100$，子种群数为 $M=8$，子种群搜索迭代次数为 $T_l=15$，混合迭代次数为 $T_g=10$，扰动概率 $CR=0.5$。

将文中方法得到的结果与经典的随机调度算法 Random、Min－min 算法和 Max－min 算法对应的任务最大完成时间 *MakeSpan* 进行比较，得到的曲线如图 2 所示。

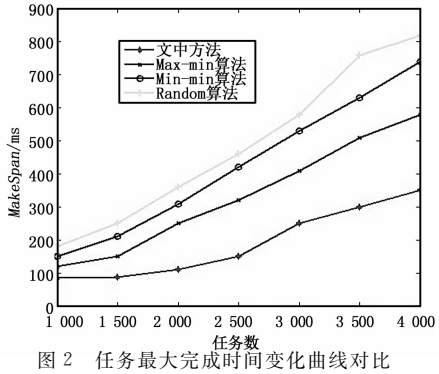

图 2 任务最大完成时间变化曲线对比

从图 2 可以看出，文中算法对应的最大完成时间 *MakeSpan* 最少，即所有任务最迟完成时间最短，随机方法最差，Max－min 算法和 Min－min 算法次之，这是因为随机方法随机为任务选择资源，Max－min 算法和 Min－min 算法每次选择最优的资源进行调度，没有从全局考虑所有任务，因此容易陷入局部最优，而文中方法从全局寻求调度的最优解，因此，具有较小的最大完成时间。

各方法对应的网格资源利用率如表 1 所示。

表 1 网格资源利用率对比

方法	资源利用率
Random 方法	61.54%
Max－min 方法	71.56%
Min－min 方法	69.67%
文中方法	95.65%

网格资源利用率即在各网格上处理任务所需时间的平均值与各网格上最晚完成时间所需时间的比值，从表 1 可以看出，文中方法对应的资源利用率最高，其平均利用率较 Random 方法、Max－min 方法和 Min－min 方法分别高 34.11%、24.09%和 25.98%。

5 结语

为了实现网格计算环境下的资源管理，设计了一种基于移动 Agent 的网格资源管理模型，将网格系统划分为应用管理模块、资源管理模块和资源表示模块，在资源管理模块中各移动 Agent 协作完成资源管理，在资源分配 Agent 处运行改进的混合蛙跳算法实现对用户任务的资源分配。在 GridSim 仿真工具下进行仿真，实验结果证明文中方法能有效地减少任务最大完成时间和较高的资源利用率，较其它方法更优，是一种适合网格分布式计算环境的有效资源管理模型。

参考文献：

[1] Jamali M A J, Sani Y. Adaptive peer to peer resource discovery in grid computing based on reinforcement learning [A]. Proc of the 12th ACIS international conference on software engineering, artificial intelligence, networking and parallel/Distributed computing [C]. Heidelberg: Springer, 2011: 191－196.

[2] 陈建刚，王汝传，张 琳，等. 基于模糊集合的网格资源访问的信任机制 [J]. 计算机学报，2009，8 (32)：1676－1681.

[3] Chen D, Chang G R, Zhang X Y, et al. A novel P2P based grid resource discovery model [J]. Journal of Computers, 2011, 6 (10): 1390－1397.

[4] 陈 晶，孔令富，潘 勋. 结合预测机制和 QoS 约束的网格资源调度算法的研究 [J]. 计算机研究与发展，2008，45 (S)：11－16.

[5] FaisaI N M, Arash O S, Stephan W. et al. Task scheduling strategies for dynamic reconfigurable processors in distributed systems [A]. Istanbul, Turkey: High Performance Computing and Simulation [C]. 2011: 90－97.

[6] 高 瞻，罗四维. 基于资源一预留图的动态网格资源预留机制 [J]. 软件学报，2011，22 (10)：2497－250.

[7] 李 立，刘元安，马晓雷. 基于组合双向拍卖的网格资源分配 [J]. 电子学报，2009，37 (1)：165－169.

[8] 张忠平，贾 倩. 一种基于最小代价的网格资源发现模型 [J]. 计算机应用研究，2012，29 (12)：4683－4690.

[9] 陈雨时，曲海成，龚小川. 基于改进蚁群算法的网格资源调度研究 [J]. 计算机工程与设计，2013，34 (2)：502－506.

[10] 王新生，陈敬男. 王伟杰. 基于信任机制的网格资源调度算法 [J]. 计算机工程，2010，36 (5)：159－164.

[11] 鲍美英，高玉斌，马 礼. 网格环境下基于域的信任模型 [J]. 计算机测量与控制，2010，18 (1)：192－197.

[12] Eusuff M M, Lansey K E. Water distribution network design using the Shuffled Frog leaping algorithm [A]. World Water Congress [C]. 2001.