

# 求解 Overlay Grid 拓扑问题的模拟退火算法

曹怀虎, 张艳梅, 牛娃

CAO Huai-hu, ZHANG Yan-mei, NIU Wa

中央财经大学 信息学院, 北京 100081

School of Information, the Central University of Finance and Economics, Beijing 100081, China

E-mail: caohhu@163.com

CAO Huai-hu, ZHANG Yan-mei, NIU Wa. Simulated annealing algorithm for Overlay Grid topology design problem. *Computer Engineering and Applications*, 2007, 43(36): 68-70.

**Abstract:** According to the opening and scalable requirement of Grid, we propose the theory model of grid architecture based on overlay network, formulate the key issue: Mix Overlay Network topology design problem, prove this problem has linear complexity under some condition, and bring forward a simulated annealing algorithm for this problem in general condition. Finally carried on the simulation experiment to evaluate the influence on topology cost of this algorithm, the result indicates the algorithm is feasible and can be deployed effectively in Grid environment.

**Key words:** grid architecture; overlay network; topology design; simulated annealing algorithm

**摘要:** 针对网格开放性和扩展性的需求, 提出了基于 Overlay Network 网格体系结构的理论模型。形式化了其核心问题: 混合 Overlay Network 拓扑设计问题, 证明了该问题在满足一定条件下具有线性复杂度, 提出了在一般条件下求解该问题的模拟退火算法。最后, 通过仿真实验分析评价了所提出算法对 Overlay Network 拓扑代价的影响, 结果表明这个模拟退火算法是可行的。

**关键词:** 网格架构; Overlay network; 拓扑设计; 模拟退火

**文章编号:** 1002-8331(2007)36-0068-03 **文献标识码:** A **中图分类号:** TP301.6

## 1 引言

网络和计算技术的发展以及人们对计算能力的需求催生了网格技术, 传统的网格主要用于高性能科学计算, 都是直接基于现有的物理网络来进行研究和构造的, 需要网络底层硬件和协议的支持。由于实际的物理网络的异构、分布、自治性以及动态演化特性, 使得网格计算变的过于复杂而不可行, 并且不能保证其开放性和扩展性。

网络技术和计算技术的融合是网格技术发展的趋势, 本文利用 Overlay Network 的虚拟和自适应的特性, 将传统的网格基于 Overlay Network 之上进行研究。根据网格技术需求的特点, 提出了一种基于 Overlay Network 的网格体系结构 GON (Grid on Overlay Network), 并且研究了其核心问题之一: 混合 Overlay Network 拓扑设计问题 MONT (Mix Overlay Network Topology Design Problem), 提出了相应的模拟退火算法。

## 2 基于 Overlay Network 的网格体系结构(GON)

针对网格的需求, 以及网格所基于的基础, 将基于 Overlay Network 网格(GON)的功能分为四层: 网络应用层、资源管理层、网络覆盖层、物理网络层, 其抽象的模型如图 1 所示。

GON 的网络拓扑结构如图 2 所示, GON 中的节点分为两种: 一种是普通节点, 此类节点数量较多, 性能较低, 稳定性不

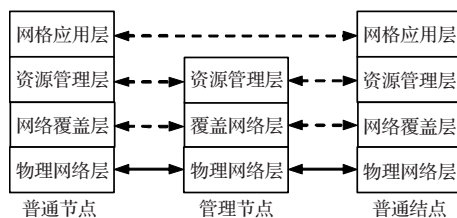


图1 基于 Overlay Network 的网格抽象模型

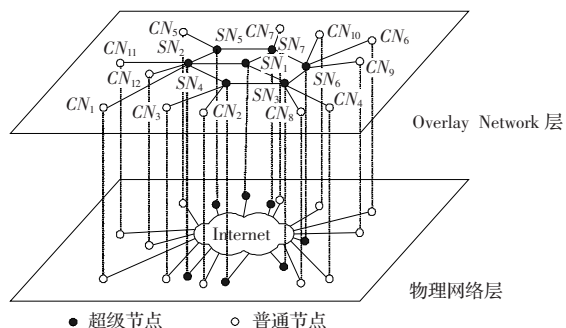


图2 GON 的网络拓扑结构模型

高, 加入和退出网格系统较频繁; 另一种是超级节点, 超级节点相对普通节点来说数量较少, 性能较高, 稳定性较高, 可信度较高, 一般不经常退出网格系统。GON 是一个混合分布式的结

构,超级节点将形成一个分布式的核心 Overlay Network,普通节点通过聚类算法生成一个聚集,称为域,每个域会形成一个聚核,由聚核负责域内的通信和节点的管理。

### 3 网络 Overlay Network 拓扑设计问题

为了构造基于 Overlay Network 的网络,首先要设计用于网络节点通信的 Overlay Network。要实现网络节点间的有效通信,关键是要设计优化的基于底层物理网络的网络 Overlay Network 拓扑结构。网络 Overlay Network 拓扑设计问题分为两个子问题:子问题一称之为核心 Overlay Network 拓扑设计问题 KONT(Kernel Overlay Network Topology Design Problem),设计由超级节点构成的核心 Overlay Network;子问题二称之为混合 Overlay Network 拓扑设计问题 MONT(Mix Overlay Network Topology Design Problem),设计由超级节点和普通节点构成的混合 Overlay Network。KONT 问题已在文献[1]中有详细的阐述,下面将对 WONT 问题进行深入的研究。

#### 3.1 MONT 问题的形式化表示及其变换

MONT 问题的目标是:设计由超级节点和普通节点构成的混合网络 Overlay Network。这个问题的输入是通过 Internet 彼此连接的超级节点(也称为管理节点)和普通节点(如图 2 所示的物理网络层的节点),为每一个普通节点分配一个管理节点,生成网络 Overlay Network 拓扑,以使生成的拓扑结果的代价最小化(这里的代价是抽象的,在实际应用中可以代表费用、延迟、带宽等),这个代价是用来反映 Overlay Network 网络代价的。

MONT 问题(Mix Overlay Network Topology Design Problem)表示如下:

$$\begin{aligned} \text{Min} \sum_{i=1}^M \sum_{k=1}^N R_{ik} a_{ik} y_{ik} + \sum_{i=1}^M \sum_{j=1}^N \sum_{k=1}^M \sum_{l=1}^N y_{ij} y_{kl} w_{ik} b_{jl} + \sum_{i=1}^M \sum_{l=1}^N R_{il} a_{il} y_{il} \\ \text{s.t.} \sum_{j=1}^N y_{ij} = 1, \text{ for } i=1, \dots, M \end{aligned} \quad (1)$$

符号表示意义说明如下:

$CN_i$ (common node):普通节点  $i$ 。

$SN_j$ (super node):管理节点  $j$ 。

$M$ :普通节点数量。

$N$ :管理节点数量。

$a_{ij}$ :普通节点  $CN_i$  与管理节点  $SN_j$  之间通信预留带宽权重(每单位带宽的接入代价)。

$y_{ij}$ :取值为 1 或 0,表示普通节点  $CN_i$  是否分配给管理节点  $SN_j$ 。

$b_{ij}$ :管理节点  $SN_i$  与  $SN_j$  之间最小代价路由的带宽权重(最小代价路由的每单位带宽的代价)。

$w_{ij}$ : $CN_i$  与  $CN_j$  之间的预留带宽。

$R_i$ :从  $CN_i$  到所有其它的普通节点的预留带宽。

公式(1)所表示的优化问题是一种二次分配问题 QAP(Quadratic Assignment Problem)的变形<sup>[2]</sup>,是一个 NP-hard 问题。整个拓扑构造问题的复杂性是受普通节点和管理结点的数量影响的,复杂性为  $O(N^M)$ 。

现在按照矩阵的观点重新考虑管理结点网络拓扑问题的参数,令矩阵  $\mathbf{R}=\{w_{ij}\}$ ,  $\mathbf{B}=\{b_{ij}\}$ ,  $\mathbf{a}=\{a_{ij}\}$ ,分别代表带宽需求,传输代价权重和访问代价权重。令  $u$  表示  $i$  的一个映射,  $u(i)=j$ , 这里  $i \in \{1, \dots, M\}$ ,  $j \in \{1, \dots, N\}$ 。根据等式(1),如果  $y_{ij}=1$ ,令  $u(i)=l$ 。注意在  $u(i) \leq N(i=1, \dots, M)$  的情况下,向量  $\mathbf{u}=(u(1), u(2), \dots,$

$u(M))$ 将对普通节点给出一个合理的分配。如果  $CN_i$  对应的  $a_{ik}=\infty$ ,则  $u(i)=k$  将不是合理解的一部分。

现在对等式(1)中的访问代价进行变换:

$$\sum_{i=1}^M \sum_{k=1}^N R_{ik} a_{ik} y_{ik} = \sum_{i=1}^M R_i \sum_{k=1}^N a_{ik} y_{ik} \quad (2)$$

$$= \sum_{i=1}^M \sum_{j=1}^M w_{ij} \sum_{k=1}^N a_{ik} y_{ik} \quad (3)$$

$$= \sum_{i=1}^M \sum_{j=1}^M w_{ij} \sum_{k=1}^N a_{ik} y_{ik} \left( \sum_{l=1}^N y_{il} \right) \quad (4)$$

$$= \sum_{i=1}^M \sum_{j=1}^M w_{ij} \left( \sum_{k=1}^N \sum_{l=1}^N a_{ik} y_{ik} y_{il} \right) \quad (5)$$

注意,由于  $R_i$  并不依赖于  $k$ ,所以等式(2)成立,根据等式(5),可以将等式(1)所表示的拓扑优化问题转换如下:

$$\begin{aligned} \text{Min} \sum_{i=1}^M \sum_{k=1}^N w_{ij} \sum_{k=1}^N \sum_{l=1}^N (a_{ik} + b_{ik} + a_{il}) y_{ik} y_{il} \\ \text{s.t.} \sum_{j=1}^N y_{ij} = 1, \text{ for } i=1, \dots, M \end{aligned} \quad (6)$$

对于指定  $i$  和  $j$  的值,总存在  $k$  和  $l$  的一个值,使得  $y_{ik} y_{il} = 1$ 。假设  $y_{iu(i)} y_{ju(j)} = 1$ ,为了保证将管理结点分配给普通节点,这里对于  $k \neq u(i)$ ,  $l \neq u(j)$ ,令  $y_{ik} y_{jl} = 0$ 。因此等式(2)所表示的目标函数可以进一步转化为:

$$Z(u) = \sum_{i=1}^M \sum_{j=1}^M w_{ij} (a_{iu(i)} + b_{u(i)u(j)} + a_{ju(j)}) \quad (7)$$

这里  $Z(u)$  就是初始问题的目标函数,一个没有约束的关于向量  $\mathbf{u}$  的最小化问题,它将给出管理节点拓扑优化问题的解。

选择一个  $v_i$ ,使得  $a_{v_i} = \min\{a_{ij}\}$ ,如果满足下面的条件这个问题将具有线性复杂度:

$$C1: b_{ij} \leq b_{ik} + b_{kj}, i, j, k \leq N_0$$

$$C2: a_{ij} \geq a_{v_i} + b_{v_i j}, i \leq M, i \leq j, v_i \leq N_0$$

引理 1 对于映射  $u(i)=v_i$ ,当  $a_{v_i} = \min\{a_{ij}\}$  时,如果矩阵  $\mathbf{A}$  和  $\mathbf{B}$  满足条件 C1 和 C2 时,目标函数  $Z(u)$  是最小的,表示为  $Z(u_1)$ 。

证明  $Z(u) = w_{11}(a_{1u_1} + b_{u_1 u_1} + a_{1u_1}) + \dots + w_{1M}(a_{1u_1} + b_{u_1 u_M} + a_{M u_M}) + \dots + w_{M1}(a_{M u_M} + b_{u_M u_1} + a_{1 u_1}) + \dots + w_{MM}(a_{M u_M} + b_{u_M u_M} + a_{M u_M})$

由于满足条件 C2,并且  $b_{v_i v_i} = 0$ ,所以:  $Z(u) \geq w_{11}(2a_{1v_1} + 2b_{v_1 v_1})$

$+ w_{12}(a_{1v_1} + a_{2v_2} + b_{v_1 v_2} + b_{v_2 v_1}) + \dots + w_{MM}(2a_{M v_M} + 2b_{v_M v_M})$

由于满足条件 C1,得到:

$$Z(u) \geq w_{11}(2a_{1v_1} + 2b_{v_1 v_1}) + \dots + w_{12}(a_{1v_1} + a_{2v_2} + b_{v_1 v_2}) + \dots + w_{MM}(2a_{M v_M} + 2b_{v_M v_M})$$

又因为:

$$Z(u_1) = w_{11}(2a_{1v_1}) + \dots + w_{12}(a_{1v_1} + a_{2v_2} + b_{v_1 v_2}) + \dots + w_{MM}(2a_{M v_M})$$

所以:  $Z(u_1) \leq Z(u)$ 。问题得证。

#### 3.2 求解 MONT 问题的模拟退火算法

如果网络不满足上面所提出的条件 C1 和 C2,那么对于大规模的网络拓扑优化问题,想通过精确的方法来求解是不可能的。本文选择了模拟退火算法,因为它对于求解二次分配问题具有很好的性能<sup>[3]</sup>。下面将提出一个基于访问代价的用于拓扑优化问题的模拟退火算法。

(1)模拟退火算法的主要思想

模拟退火的思想源于统计物理学和组合优化问题,它是仿

效冷却的金属结晶化的过程,也就是退火的过程满足一个热能的平衡式<sup>[4]</sup>。遵守这个热能等式的系统处于不同的能量水平  $E_k$  和温度  $t$ , 然后对这个系统施加一个随机的扰动,同时分析评价相应的能量的改变。如果新的能量水平  $E_j$  低于  $E_k$ ,那么这个扰动将被接受,并且这个系统将演变到一个新的状态。如果这个能量水平增长了,那么这个系统将按照一个概率演变到一个新

的状态,这个概率为  $e^{-\frac{E_k-E_j}{t}}$ 。经过对所产生的大量的状态的评价分析认为,温度降低,新的状态将产生。当温度降低时,接受一个新的扰动的概率也是降低的,这个扰动是增加目前状态能量的。当进一步的扰动不再增加系统的能量水平时,这个算法结束。

#### (2)求解 MONT 问题的模拟退火算法

温度用  $t$  表示,初始值为  $t_0$ ,  $t_0$  将被设定为一个高值(这里,设为 100),初始解用  $S_0$  表示,变量  $S_{best}$ 、 $S_{cur}$ 、 $S_{new}$  分别代表最好解,当前解,当前迭代所得到的最新解。这里用  $Z_0$ 、 $Z_{best}$ 、 $Z_{cur}$ 、 $Z_{new}$  分别表示目标函数的初始解,新解,最好和当前解。利用局部随机搜索算法  $SEARCH(S_{cur})$ ,从一个当前解  $S_{cur}$  得到一个新解  $S_{new}$ 。假设现在执行的情况是,普通节点  $CN_k$  被分配给管理节点  $SN_j$ ,通过下面的等式可以变换目标函数:

$$\Delta(Z_{new}, Z_{cur}) = \sum_{j=1, j \neq k}^M w_{kj} (a_{kl} + b_{lj} - a_{ku} - b_{uj}) + \sum_{i=1, i \neq k}^M w_{ik} (a_{kl} + b_{ul} - a_{ku} - b_{uj})$$

如果  $\Delta(Z_{new}, Z_{cur}) < 0$ ,那么这个新解可以被接受,在这种情况下,将检查最好解是否可以被改善;

如果  $\Delta(Z_{new}, Z_{cur}) \geq 0$ ,则这个新解将以温度  $t$  降低的概率被接受,一个 0-1 选择函数  $Rand(0, 1)$  将被用来决定是否  $Z_{new} = Z_{cur} + \Delta(Z_{new}, Z_{cur})$ 。

温度是以几何级数降低的,这里  $t = t_c \cdot r$ ,  $0 < r_c < 1$ , 本文  $r_c$  的取值为 0.9。在每一个温度水平,一定数量的解是被评价的,这个数量是用循环参数  $Rep_{max}$  来表示的。这个循环参数应当足够大,以使在每一个温度水平更好的解可以被发现。这个处理过程一直在继续,直至达到某一温度,在这个温度下,目标函数的解不能够进一步的改善。在这个温度点,算法结束并且产生目标函数最好的解  $S_{best}$  的值  $Z_{best}$ 。算法的伪代码如下:

Simulated Annealing Algorithm( $S_0, Z_0, t_0$ );

begin

$S_{best} \leftarrow S_{cur} \leftarrow S_0$

$Z_{best} \leftarrow Z_{cur} \leftarrow Z_0$

$t \leftarrow t_0$

repeat

for  $i=1$  to  $Rep_{max}$  do

$S_{new} \leftarrow SEARCH(S_{cur})$

if  $\Delta(Z_{new}, Z_{cur}) < 0$

$S_{new} \leftarrow S_{cur}$

if  $Z_{new} < Z_{best}$

$S_{best} \leftarrow S_{new}$

$Z_{best} \leftarrow Z_{new}$

endif

else

if  $e^{-\Delta(Z_{new}, Z_{cur})/t} > Rand(0, 1)$

$S_{new} \leftarrow S_{cur}$

endif

endif

endfor

$t \leftarrow r_c \cdot t$

until no changes in the objective function

return  $S_{best}, Z_{best}$

end

## 4 算法的仿真性能分析

下面考虑不满足条件 C2 情况下,也就是得到最小代价的网络是不可能的情况下,比较用模拟退火算法和贪婪算法计算网络拓扑问题的性能<sup>[5]</sup>。这个实验环境是一个随机生成的网络,普通节点和管理节点的数量在 100~1 000 之间变化,这里  $M=N$ ,边概率  $P=0.2, 0.6$  或  $0.8, p_a=0.8$ 。对于每一个普通节点  $CN$  随机选择一个管理节点  $SN_{v_i}$ ,  $a_{v_i}$  是按分布  $[1, 100]$  随机选择。对于管理节点  $j \neq v_i$ ,  $a_{ij}$  按分布  $[1, 100]$  随机选择,强制  $a_{v_i} \leq a_{ij} \leq a_{v_i} + b_{v_j}$ , 以确保将满足条件 C2 的情况排除。

结果展示在图 3~图 5 中,实验的结果进行了标准化的处理,图中的代价值称为相对拓扑代价。这里将贪婪算法的拓扑代价设为 1,相对拓扑代价为其它算法的拓扑代价与贪婪算法的拓扑代价的比值。

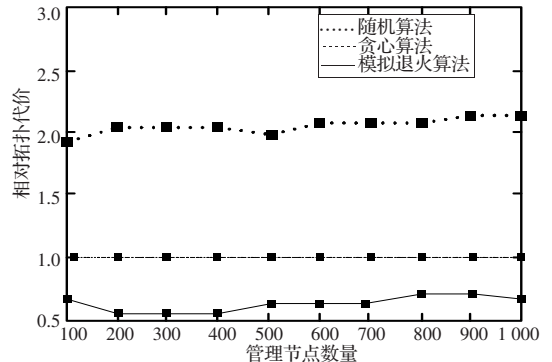


图3  $P=0.2$  时算法的相对拓扑代价比较

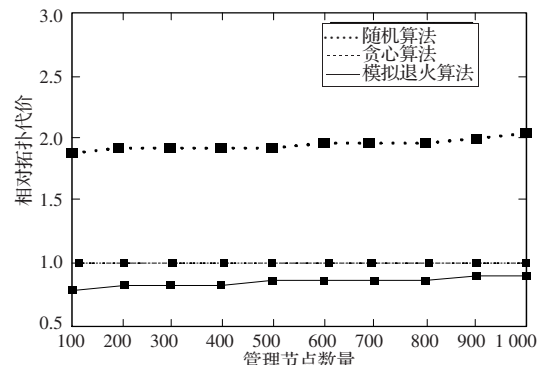


图4  $P=0.6$  时算法的相对拓扑代价比较

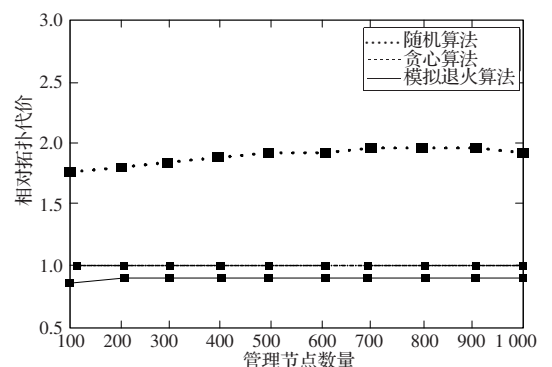


图5  $P=0.8$  时算法的相对拓扑代价比较