第 29 卷 第 8 期
2006 年 8 月
计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS
Vol. 29 No. 8
Aug. 2006

并行网络蠕虫模拟中任务优化划分的研究

王晓锋 方滨兴 云晓春 张宏莉
（哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001）

摘 要 为提高并行网络蠕虫模拟的性能，需要对蠕虫模拟任务进行合理的划分. 鉴于基于图划分工具的任务划分方法存在的不足，提出了并行网络蠕虫模拟任务的优化划分方法：以并行网络蠕虫模拟运行时间估计模型作为优化目标函数，采用改进的模拟退火算法实现对蠕虫模拟任务的划分. 在 PDNS 上进行的 Slammer 蠕虫传播模拟实验表明，该优化划分方法较基于图划分工具的方法提高模拟性能 20% 以上.

关键词 蠕虫模拟；并行网络模拟；网络模拟性能；拓扑划分；模拟退火算法
中图法分类号 TP309

Optimized Partitioning for Parallel Simulation Task of Network Worms

WANG Xiao-Feng FANG Bin-Xing YUN Xiao-Chun ZHANG Hong-Li
(*School of Computer Science and Technology, Harbin Institute of Technology, Harbin* 150001)

Abstract To improve the performance of parallel simulation of network worms, the simulation task of worms should be partitioned reasonably. As there are shortcomings of the partitioning method based on graph partitioning tools, a method for optimized partitioning for parallel simulation task of network worms is developed: By treating the model for estimating the running time of parallel simulation of network worms as optimization object function, the simulation task of worms is partitioned by the improved simulated annealing. Experiments of Slammer worm propagation simulation in PDNS show that the optimized partitioning method can improve the performance of simulation by over 20% compared to the partitioning method based on graph partitioning tools.

Keywords simulation of worms; parallel network simulation; performance of network simulation; topology partitioning; simulated annealing

1 引 言

网络蠕虫由于具有传播范围广、速度快、破坏性大的特点，已成为 Internet 的最大安全威胁之一，并得到了广泛的研究[1]. 当前，研究蠕虫的方法分为两种：数学模型和模拟. 数学模型的主要特点是计算开销少，成本低；而模拟能够更细致地分析蠕虫，研究网络拓扑、链路带宽延迟、路由协议等对蠕虫的影响.

近些年来，网络蠕虫模拟得到了普遍的研究，文献[2～4]的工作都实现了网络蠕虫的模拟. 这些蠕虫模拟方法都采用并行分布式技术以满足蠕虫模拟所需的规模以及降低模拟所需的运行时间. 并行网络蠕虫模拟的一个主要的问题是如何合理地划分模

收稿日期：2006-04-04；修改稿收到日期：2006-06-04. 本课题得到国家“九七三”重点基础研究发展规划项目基金（2005CB321806）和新世纪优秀人才支持计划项目基金（NCET-04-0338）资助. 王晓锋，男，1978 年生，博士研究生，研究兴趣为网络模拟、网络安全. E-mail：wxf@pact518.hit.edu.cn. 方滨兴，男，1960 年生，博士，教授，博士生导师，中国工程院院士，研究兴趣为网络信息安全、并行计算等. 云晓春，男，1971 年生，博士，教授，博士生导师，研究兴趣为计算机网络、信息安全等. 张宏莉，女，1973 年生，博士，教授，博士生导师，研究兴趣为网络信息安全、并行处理等.

拟任务，以提高蠕虫模拟的性能. 由于并行网络模拟一般采用“拓扑划分”实现模拟任务的并行运行[5]（即将所要模拟的整个网络划分成若干子网，每个子网的模拟任务由单独的模拟节点承担），因此一个很直接的划分方法是，采用 METIS[6] 等成熟的图划分工具对所要模拟的网络拓扑进行划分[7～9]. 然而，这些基于 METIS 工具的划分存在着 3 个缺点：

(1) METIS 使用贪婪算法寻求网络拓扑的最优划分，而贪婪算法易使得划分结果陷入局部最优.

(2) 不能合理地将安全模拟时间窗口长度(lookahead)的最大化作为划分目标.

定义 1. 安全模拟时间窗口是一个模拟时间段，模拟节点可以安全地处理这个模拟时间段内的所有事件，而不必担心会出现因果错误[9].

安全模拟时间窗口长度越大，需要的同步次数越少，因此并行模拟的额外开销就越少. 对于并行模拟而言，安全模拟时间窗口长度为所有被划分的链路延迟的最小值[10].

METIS 划分的目标是：各子图点权值和满足一定均衡的情况下，被划分边权值和最小. 而并行模拟任务划分的目标有 3 个，即负载均衡、远程通信开销的最小化以及安全模拟时间窗口长度的最大化[9]. 为了依靠 METIS 划分目标实现并行网络模拟任务划分的 3 个目标，需要合理地确定点和边的权值. 文献[7]中的点权值与边权值分别为节点与链路需要转发的数据流大小，这种方法可以利用 METIS 的划分目标实现负载的一定均衡以及通信量的最小化，但不能将安全模拟时间窗口长度的最大化作为划分目标. 文献[8,9]将点权值定义为节点需要转发的数据流大小，边权值定义为链路需要转发的数据流大小与链路延迟倒数的加权和. 而这种方法也没有能够合理地将安全模拟时间窗口长度的最大化作为划分目标，因为被划分链路延迟倒数和最小并不意味着被划分链路延迟最小值最大. 一个简单的例子如图 1 所示：将图 1 划分为两份，如果追求被划分链路延迟倒数和最小应该划分 0 节点与 1 节点间的链路，此时安全模拟时间窗口长度为 2ms；如果追求被划分链路延迟最小值最大，则显然可以使安全模拟时间窗口长度等于 3ms.

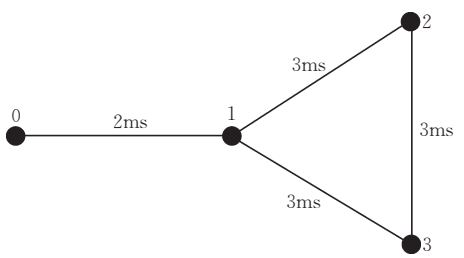

图 1 一个简单的拓扑

(3) 不能综合考虑模拟运行环境、具体的模拟应用等影响模拟性能的因素进行最合理的划分.

并行模拟任务划分的 3 个目标哪个“更重要”需要根据模拟运行环境和模拟任务确定. 对于通信延迟较小的硬件环境，可以重点考虑负载均衡；而对于通信延迟较大的硬件环境，应该注重考虑如何降低远程通信量. 同样，如果某个模拟应用需要进行大量的同步，同步开销相对于本地计算、远程通信开销很大，则需要重点考虑如何降低同步开销. 文献[7～9]没有考虑这些目标的重要性是随着模拟硬件环境以及模拟应用不同而变化的，因此不能实现对模拟任务的最合理划分.

鉴于上述分析，本文提出了并行网络蠕虫模拟任务的优化划分方法，以提高蠕虫模拟性能，降低模拟所需的运行时间：将并行网络蠕虫模拟运行时间估计模型作为目标函数，采用改进的模拟退火算法对模拟任务进行划分. 由于模拟退火算法是一种全局优化算法[11]，因此该方法能够避免陷入局部极值；同时，所使用的运行时间估计模型包括了安全模拟时间窗口长度、负载均衡、通信开销、模拟应用、运行环境等因素，基于该模型的划分能够综合考虑这些因素，实现对蠕虫模拟任务的合理划分.

本文第 2 节阐述了并行网络蠕虫模拟的优化划分方法；第 3 节通过在 PDNS[5]（由佐治亚理工学院开发的并行分布式网络模拟器，实现了 NS 的并行化）上进行的 Slammer 蠕虫传播模拟实验，验证该优化划分方法对蠕虫模拟任务划分的有效性；第 4 节是结论及以后的工作.

2 并行网络蠕虫模拟的优化划分

图 2 描述了蠕虫模拟任务优化划分的过程，它的核心是采用改进的模拟退火算法对蠕虫模拟任务进行划分以产生优化的划分结果，模拟退火算法所采用的目标函数是并行网络蠕虫模拟运行时间估计模型. 运行时间估计模型可以根据硬件参数、网络拓扑中节点和链路的核心程度、链路的延迟、蠕虫模拟事件数、蠕虫模拟时间以及划分情况等因素估计蠕虫模拟运行所需要的时间. 硬件参数根据模拟运行环境确定，节点与链路的核心程度及链路延迟根据模拟拓扑确定. 模拟事件数与模拟时间根据设定的

蠕虫模拟应用确定. 划分情况就是由模拟退火算法进行划分后得到的结果. 值得注意的是，估计模型与模拟退火算法是互为反馈的：估计模型针对模拟退火算法产生的划分结果进行估计，而模拟退火算法需要估计模型作为目标函数. 下面依次对并行网络蠕虫模拟运行时间估计模型、模型中各参数的确定方法及改进的模拟退火算法进行介绍.

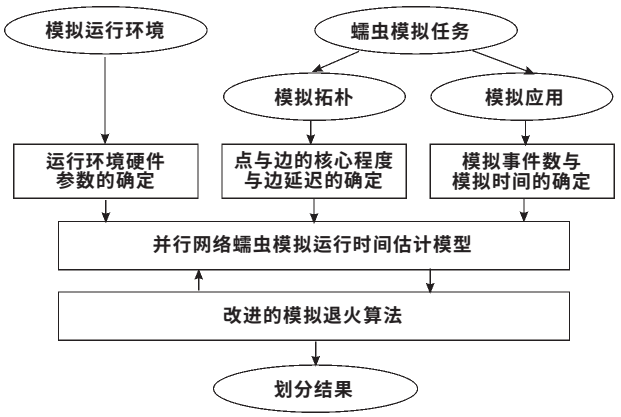

图 2　蠕虫模拟任务优化划分过程

2.1　并行网络蠕虫模拟运行时间估计模型

定义 2.　事件：数据包从一个网络节点通过链路转发到另一个相邻的节点的模拟处理过程.

定义 3.　模拟性能：网络模拟器处理事件的效率，可以用事件总数除以模拟运行时间得到[5].

定义 4.　子网：将拓扑图划分成若个子拓扑图，每一个子拓扑图即为一个子网.

定义 5.　远程链路：网络拓扑图被划分成若干个子网后，两个端点不在同一子网上的链路.

定义 6.　远程事件：所模拟的数据包信息通过物理链路从一个模拟节点传到另一个模拟节点的过程.

文献[12]提出了一个并行网络模拟性能的估计模型，该模型综合考虑了各种对模拟性能有影响的因素，提高了模型的准确性，同时它对并行网络模拟任务的划分有指导意义. 由定义 3 可知，模拟运行时间是模拟事件总数除以模拟性能，因此依靠文献[12]中的性能估计模型，得到本文所需的并行网络蠕虫模拟运行时间估计模型：

$$\begin{aligned} T &= GE_{\text{comp}} + K_{\text{comm}}E_{\text{comm}} + K_{\text{syn}}MN \\ &= G(f_bE/M) + K_{\text{comm}}(f_cE) + K_{\text{syn}}M(T_{\text{sim}}/f_l) \end{aligned} \tag{1}$$

其中 M 为模拟节点个数；G 为计算粒度，即模拟节点处理一个事件需要的时间；K_{comm} 为通信开销粒度，即模拟节点每发送或接收一个远程事件需要的时间；$K_{\text{syn}}M$ 为 M 个模拟节点进行一次同步需要的时间开销，而 K_{syn} 称为同步开销粒度；E_{comp}（即 f_bE/M）为划分最多事件数，即网络拓扑被划分成若干子网后，具有最多事件的子网所包含的事件数，f_b 称为负载不均衡因素，E 为模拟事件总数；E_{comm}（即 f_cE）为划分最多远程事件数，即网络拓扑被划分后，具有最多远程事件数的子网所包含的远程事件数，f_c 被称为通信开销因素；N（即 T_{sim}/f_l）为同步次数，即并行模拟过程中进行的同步次数总合，f_l 被称为安全模拟时间窗口长度因素，T_{sim} 为模拟时间. 其中 $M, G, K_{\text{comm}}, K_{\text{syn}}$ 由模拟运行的硬件环境决定；T_{sim} 与 E 由具体的蠕虫模拟应用决定；f_b, f_c 和 f_l 由模拟应用以及模拟任务的划分决定.

2.2　估计模型中各参数的确定方法

2.2.1　硬件参数的确定

模型中的硬件参数主要包括 $M, G, K_{\text{comm}}, K_{\text{syn}}$ 等. 文献[12]中提出通过运行标准的模拟应用来确定模型的硬件参数 G, K_{comm} 与 K_{syn}. 以 G 为例，具体步骤如下：

(1) 设计一组标准的模拟应用，这些模拟应用除了划分最多事件数 E_{comp} 不同外，其它参数（包括划分最多远程事件数 E_{comm} 与同步次数 N）都相同；

(2) 逐个运行这些模拟应用，并记录相应的模拟运行时间 T，得到一系列离散点（$(E_{\text{comp}})_i, T_i$）；

(3) 对上述离散点进行直线拟合，得到直线方程 $T = kE_{\text{comp}} + b$，根据式(1)，斜率 k 即为 G.

同理，设计一组除划分最多远程事件数 E_{comm} 不同外，其它参数都相同模拟应用，采用类似步骤可得到 K_{comm}；设计一组除同步次数 N 不同外，其它参数都相同模拟应用，采用类似步骤可得到 K_{syn}.

本文所使用的实验环境见 3.1 节，模拟节点为 4 个（即 $M=4$），通过上述方法得到计算粒度 $G=4.54\text{E-}5$，通信开销粒度 $K_{\text{comm}}=1.374\text{E-}5$，同步开销粒度 $K_{\text{syn}}=1.14\text{E-}4$.

2.2.2　与蠕虫模拟应用有关的参数的确定

与蠕虫模拟应用有关的参数包括 T_{sim} 与 E. 其中 T_{sim} 在描述蠕虫模拟应用时能被确定（对于 NS，就是脚本中设定的蠕虫模拟时间）. 由于蠕虫模拟具有随机性，在模拟之前无法准确地知道 E，因此必须通过估计的方法. E 可以由下式估计[5]：

$$E = N_F \times P_F \times H_F \tag{2}$$

其中 N_F 为所要模拟的数据流的估计；P_F 为平均每一个数据流所需要发送的数据包数；H_F 为平均每一个数据包需要的跳数.

以蠕虫传播模拟为例，N_F 为蠕虫模拟总扫描次数；P_F 为每一次扫描发送的数据包个数，对于 Slammer 蠕虫来说为 1；H_F 是拓扑图中所有易感染主机节点间跳数的平均值. 蠕虫模拟的总扫描次数可以通过下式得到

$$N_F=\sum_{i=1}^{K} H_i R \tag{3}$$

其中 $K=[T_{\text{sim}}/\text{单位时间}]$，表示模拟时间 T_{sim} 包含的单位时间个数；H_i 为第 i 个单位时间开始时已感染主机节点的个数，它可以通过蠕虫传播数学模型估计得到；R 是蠕虫每单位时间内的扫描次数.

结合式(2)和(3)，估计了三组 Slammer 蠕虫传播模拟应用的模拟事件数 E. 三组模拟应用的路由器规模为 200，400 和 800，易感染主机个数为 3424，6368，12384. E 的估计值与真实值的之间的平均误差为 7.6%.

2.2.3 与划分有关的参数的确定

与划分有关的参数包括 f_b，f_c 和 f_l. 事实上，这些参数不仅仅与拓扑划分情况有关系，而且也与模拟任务的拓扑有关系.

设网络拓扑中节点个数为 N，并用 1 到 N 的自然数进行编号，在模拟过程中，经过第 i 个节点的数据包个数(即事件个数)为 E_i^N；网络拓扑中链路条数为 L，并用 1 到 L 的自然数进行编号，在模拟过程中，经过第 j 条链路的数据包个数为 E_j^L. 由定义 2 可知，模拟事件总和与经过节点的数据包个数总和以及经过链路的数据包个数总和等价，即

$$\sum E_i^N=\sum E_j^L=E \tag{4}$$

由于蠕虫模拟应用具有随机性，因此无法像文献[7～9]那样，首先明确每一个所要模拟的数据流的大小，然后确定模拟过程中，每一个节点与链路经过的数据包个数. 考虑到在计算机网络运行中，尤其在蠕虫等大规模网络安全事件爆发时，核心程度越高的节点与链路所需要转发的数据包个数越多，这一点同样适用于网络模拟：节点与链路的核心程度反映了网络模拟过程中节点与链路事件数的相对大小. 鉴于上述思想，可以首先确定节点、链路在拓扑图中的核心程度，然后依据核心程度将 E 个事件按比例分配给所有的节点与链路. 核心程度的确定以及有效性验证可以见文献[13].

通过上述方法，可以得到经过每一个节点与链路的数据包个数. 设网络被划分成 M 个子网，第 l 个子网中所有节点数据包总和为 E_l^S，第 l 个子网的所有远程链路数据包总和为 E_l^R. 于是可以确定划分的最多事件数为

$$E_{\text{comp}}=f_b E/M=\max\{E_1^S,E_2^S,\cdots,E_M^S\} \tag{5}$$

划分的最多远程事件数为

$$E_{\text{comm}}=f_c E=\max\{E_1^R,E_2^R,\cdots,E_M^R\} \tag{6}$$

根据式(5)，(6)，可以确定 f_b 与 f_c.

f_l 为平均每次同步推进的模拟时间长度，即安全模拟时间窗口长度因素，它等于所有远程链路延迟的最小值.

2.3 改进模拟退火算法的实现

当前最常用的优化方法包括：遗传算法、蚂蚁算法、爬山法以及模拟退火算法. 遗传算法以及模拟退火算法已被用于网络拓扑图的划分中：文献[14]采用遗传算法实现对计算机网络的最优划分以改善网络性能；文献[15]采用传统的模拟退火算法实现网络仿真中虚拟节点与物理节点匹配，以满足网络仿真的软硬件需求，并减少网络仿真所要占用的资源. 遗传算法以及蚂蚁算法是基于种群的优化方法，它每一步迭代需要计算种群中所有个体的目标函数值，当目标函数比较复杂时，每一步需要很长的计算时间. 由于蠕虫模拟运行时间估计模型(即目标函数)每一次计算需要比较长的时间，因此采用遗传算法或蚂蚁算法对并行网络蠕虫模拟任务进行划分需要的时间开销太大. 实现对模拟任务优化划分的目标是降低并行网络模拟的运行时间，而如果并行网络模拟任务划分需要占用相当长时间的话，优化划分就显得没有意义. 爬山法以及模拟退火算法是基于个体的优化方法，因此每一步迭代需要的计算开销很小. 爬山法采用“以退为进”的策略，但是易陷入局部最优而不能获得较优的划分. 模拟退火算法具有寻找全局最优解的能力，但也具有收敛速度慢、执行时间长的缺点.

鉴于上述分析，本文采用模拟退火算法作为优化方法，并通过对模拟退火算法的改进，提高模拟退火算法的优化速度. 改进的模拟退火算法的基本思想是：基于模拟退火算法的全局优化特性，引入对蠕虫模拟运行时间估计模型中与划分有关的参数(f_b，f_c 和 f_l)的随机局部优化，以加快优化速度. 在介绍算法之前，有必要对下列名词进行描述.

定义 7. 划分结果表示：拓扑中包含的节点个数为 N，采用 $1\sim N$ 进行编号，拓扑被划分成 M 个子网，用 $1\sim M$ 进行编号；划分结果表示为数组 $S[N]$，其中数组元素 $S[i]=k$ 表示编号为 i 的节点被划分到第 k 个子网.

为了简化起见，划分时忽略了网络拓扑中的主机节点，而只对路由器进行划分.

定义 8.　边界节点. 远程链路的端点.

定义 9.　节点-子网关联强度. 节点 j 与子网 k 的节点-子网关联强度表示为 $R(j,k)$，其值等于以节点 j 作为一个端点，以子网 k 中的某一个节点作为另一个端点的所有链路事件数总和.

改进模拟退火算法描述如下：

//初始化

1. 温度 $T=10000$，降温参数 $k=0.9$；

2. 无进展退火次数 $M=0$；

3. 当前最优划分 $S_{best}=S_0$；

// f_l 局部优化

4. $S_{init}=S_{best}$，$S_{temp}=S_{best}$，$L_{delay}=\varnothing$；

5. 确定 S_{temp} 划分中远程链路延迟最小的链路集 L_{delay}；

6. for L_{delay} 中每一条远程链路 l_i

{

(a) 随机选定 l_i 的一个端点，设其编号为 N_i^1，设另一个端点的编号为 N_i^2；

(b) $S_{temp}[N_i^1]=S_{temp}[N_i^2]$；

}

7. 用式(1)估计 S_{temp} 与 S_{best} 的模拟运行时间得到 t_{temp} 与 t_{best}，$\Delta t=t_{temp}-t_{best}$；

8. 如果 $\Delta t<0$，$S_{best}=S_{temp}$；否则以概率 $e^{-\Delta t/T}$ 执行 $S_{best}=S_{temp}$；

// f_b 局部优化

9. $S_{temp}=S_{best}$，$N_{split}=\varnothing$；

10. 确定 S_{temp} 划分中事件数总和最多的子网 m 及其边界节点集 N_{split}；

11. 随机选择 N_{split} 中的某一节点 i，确定与节点 i 有最大节点-子网关联强度的子网 $p(p\neq m)$，$S_{temp}[i]=p$；

12. 用式(1)估计 S_{temp} 与 S_{best} 的模拟运行时间得到 t_{temp} 与 t_{best}，$\Delta t=t_{temp}-t_{best}$；

13. 如果 $\Delta t<0$，$S_{best}=S_{temp}$；否则以概率 $e^{-\Delta t/T}$ 执行 $S_{best}=S_{temp}$；

// f_c 局部优化

14. $S_{temp}=S_{best}$，$N_{link}=\varnothing$；

15. 确定 S_{temp} 划分中远程事件数总和最多的子网 n 以及节点集合 N_{link}，其中，$N_{link}=\left\{i \mid S_{temp}[i]=n \wedge R(i,n)<\sum_{k=1,k\neq n}^{M} R(i,k)\right\}$；

16. 随机选择 N_{link} 中的某一节点 j，确定与节点 j 有最大节点-子网关联强度的子网 $q(q\neq n)$，$S_{temp}[j]=q$；

17. 用式(1)估计 S_{temp} 与 S_{best} 的模拟运行时间得到 t_{temp} 与 t_{best}，$\Delta t=t_{temp}-t_{best}$；

18. 如果 $\Delta t<0$，$S_{best}=S_{temp}$；否则以概率 $e^{-\Delta t/T}$ 执行 $S_{best}=S_{temp}$；

//结束条件判断

19. S_{best} 与 S_{init} 相比是否有更新，如果有，则 $M=0$；否则，$M=M+1$；

20. 若 $M\geq 10$，报告最优划分 S_{best} 并退出；否则，$T=kT$，并转到步 4.

算法中 4～8 行是对 f_l 的局部优化，它通过消除被划分链路中具有最小延迟链路的划分来提高 f_l 值；9～13 行是对 f_b 的局部优化，它通过移走具有最大事件数子网中的某个边界节点来降低 f_b 值；14～18 行是对 f_c 的局部优化，它通过移走具有最大远程事件数子网中的某个边界节点(该节点与所在子网的关联强度小于该节点与其它所有子网关联强度的总和)来降低 f_c 值. 这些局部优化都带有随机性，同时又使得对应的参数向较优的方向发展. 由于 METIS 划分方法在负载均衡以及减少通信开销方面具有一定的优势，因此为减少划分运行时间，将 METIS 划分结果作为初始划分 S_0. 当连续 10 次最优划分没有改变，算法停止运行并返回最优划分.

表 1 列出了利用改进模拟退火算法对蠕虫模拟任务进行划分所需的时间，可以看出该算法能够快速地收敛. 从第 3 节中的实验结果可以发现，该算法能够得到比 METIS 更优的划分. 由于模拟时间估计模型计算开销是 $O(N^2)$，因此随着网络拓扑中路由器个数变大，模拟退火算法运行时间也相应变长. 采用传统的模拟退火算法进行划分时发现，划分过程中的当前最优划分长时间在初始划分附近震荡，而不能获得比初始划分更优的划分.

表 1　改进模拟退火算法的运行时间

路由器个数	运行时间(s)
200	3.8
400	15.0
800	36.0

3　实验结果与分析

3.1　实验环境

4 台曙光节点机，每台节点机有 4 个 1GHz 的 CPU 及 2GB 的 RAM；节点间采用千兆局域网互联.

3.2　模拟的拓扑与应用

使用 NEM[16] 拓扑生成工具生成 3 组路由器级别的网络拓扑，分别具有 200，400 及 800 个路由器节点，将这些拓扑中每一个度为 1 的路由器(终端路由器)连接 32 个主机节点，这 3 个拓扑图所包含的弱点主机分别为 3424，6368，12384. 路由器间的延

迟按概率在 0.1～40ms 之间随机给定. 设定蠕虫扫描速率为 5/s,在第 1 秒时有 1 台已感染主机. 设置模拟时间为 100s.

3.3 实验结果分析与比较

针对上述 3 个拓扑,分别采用 METIS 划分方法(MP,其中拓扑图中的节点权值与链路权值分别为经过该节点与链路的数据包个数)以及本文提出的优化划分方法(OP)进行划分. 根据划分结果,生成相应的 Slammer 蠕虫模拟脚本并在 PDNS 上运行,运行结果比较如图 3～图 6 所示.

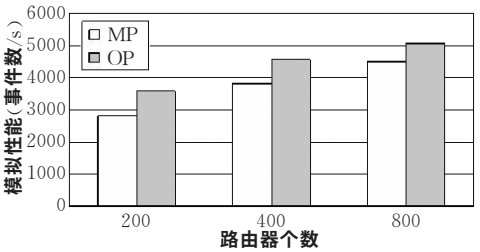

图 3 模拟性能比较

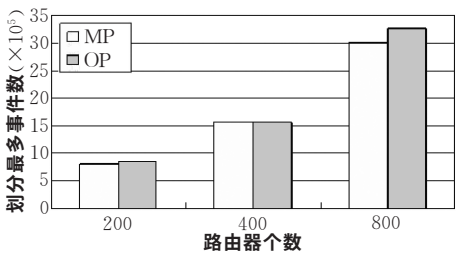

图 4 划分最多事件数比较

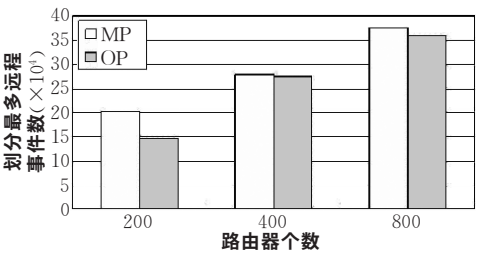

图 5 划分最多远程事件数比较

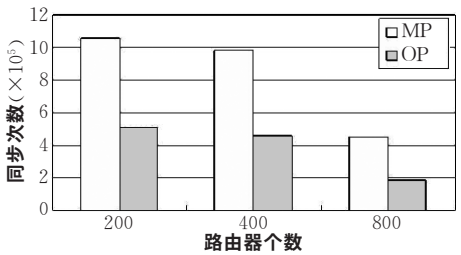

图 6 同步次数比较

图 3 是模拟性能的比较. 从图 3 中可以看出 OP 相对于 MP 能明显的提高蠕虫模拟的性能,三组实验平均性能提高了 20.4%. 根据式(1)以及定义 3,在给定的硬件环境以及模拟任务下,模拟性能主要受划分最多事件数、划分最多远程事件数、同步次数的影响;图 4～图 6 分别比较了这三方面的因素. 从图 4 中可以看出,OP 并不能够比 MP 获得更高的负载均衡,对于 400 个路由器的拓扑,最多事件数降低了 1.11%,而对于 200 与 400 个节点的路由器,最多事件数反而提高了 5.35%与 8.27%. 这一方面是由于 METIS 划分的首要目标是实现负载的一定均衡,并且在这一方面效果很不错;另一方面是 OP 所追求的是综合考虑各种因素,实现优化划分,因此允许负载均衡有一定的降低,而总体性能得到显著提高. 从图 5 中可以看出,OP 相对于 MP 能够降低划分最多远程事件数:对于 3 个拓扑,其最多远程事件数分别降低了:27.6%,1.79%,4.46%. 这主要是由于,METIS 使用贪婪算法对被划分边权值和的优化易陷入局部最小,而 OP 依靠退火算法,能够跳出局部最小点,进一步地降低划分最多远程事件数. 从图 6 中可以看出,OP 相对于 MP 大大地降低了同步次数. 这是由于 MP 并不能将减少同步次数作为目标,而依靠模拟运行时间估计模型,采用模拟退火算法的划分能够综合考虑划分最多事件数、划分最多远程事件数以及同步次数.

与 MP 相比,OP 需要额外进行模拟退火算法划分的过程,表 1 中列出了一次划分所需要的时间. 对于蠕虫模拟来说,每次运行的时间相当长(几小时,甚至几十小时),因此多几十秒钟的划分开销而能减少几个小时的蠕虫模拟运行时间是有价值的.

4 结论及以后的工作

由于网络蠕虫模拟需要相当大的时间开销,为提高模拟性能,减少模拟运行时间,有必要对蠕虫模拟任务进行合理的划分. 本文提出了并行网络蠕虫模拟任务的优化划分方法,以弥补基于 METIS 的划分方法存在的不足,并提高蠕虫模拟的性能. 本文将并行网络蠕虫模拟运行时间模型作为优化划分目标函数,采用改进的模拟退火算法对蠕虫模拟任务进行优化划分. 该方法能够综合考虑各种因素,包括模拟运行的硬件环境、负载均衡、远程通信开销、安全模拟时间窗口长度等,实现蠕虫模拟任务的优化划分;由于模拟退火算法具有全局寻优的特点,因此该方法具有寻找全局最优划分的能力,避免陷入局部最优. 多组实验表明,相对于基于 METIS 工具的

划分方法，该方法能够有效地提高并行网络蠕虫模拟的性能（平均提高了 20.4%）.

当然，目前的工作还存在着不足：虽然实验表明改进的模拟退火算法运算时间短，同时基于改进模拟退火算法进行网络模拟任务的划分能够显著提高蠕虫模拟的性能，但是缺乏对此算法的详细分析. 今后的工作主要会在这方面做进一步完善.

参 考 文 献

1 Wen Wei-Ping, Qing Si-Han, Jiang Jian-Chun *et al*. Research and development of internet worms. Journal of Software, 2004, 15(8): 1208～1219(in Chinese)
（文伟平，卿思汉，蒋建春等. 网络蠕虫研究与进展. 软件学报，2004，15(8)：1208～1219）

2 Liljenstam M., Nicol D., Berk V. *et al*. Simulating realistic network worm traffic for worm warning system design and testing. In: Proceedings of the ACM Workshop on Rapid Malcode (WORM'03), Washington DC, USA, 2003, 24～33

3 Perumalla K., Sundaragopalan S.. High-fidelity modeling of computer network worms. In: Proceedings of the 20th Annual Computer Security Applications Conference, Tucson, AZ, USA, 2004, 126～135

4 Riley G., Sharif M., Lee W.. Simulating internet worms. In: Proceedings of the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, Volendam, The Netherlands, 2004, 268～274

5 Fujimoto R., Perumalla K., Park A. *et al*. Large-scale network simulation: How big? how fast? In: Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, Orlando, Florida, USA, 2003, 166～173

6 Karypis G., Kumar V.. Multilevel *k*-way partitioning scheme for irregular graphs. Journal of Parallel and Distributed Computing, 1998, 48(1): 96～129

7 Yocum K., Eade E., Degesys J. *et al*. Toward scaling network emulation using topology partitioning. In: Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, Orlando, Florida, USA, 2003, 242～245

8 Liu X., Chien A.. Traffic-based load balance for scalable network emulation. In: Proceedings of the ACM Conference on High Performance Computing and Networking, Phoenix, Arizona, USA, 2003, 40～50

9 Xu D., Ammar M.. BencHMAP: Benchmark-based, hardware and model-aware partitioning for parallel and distributed network simulation. In: Proceedings of the IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, Volendam, The Netherlands, 2004, 455～463

10 Nicol D., Liu J.. Composite synchronization in parallel discrete-event simulation. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(5): 433～446

11 Lu Qing-Wen, Chen Wu-Fan. Image segmentation based on mutual information. Chinese Journal of Computers, 2006, 29(2): 296～301(in Chinese)
（吕庆文，陈武凡. 基于互信息量的图像分割. 计算机学报，2006，29(2)：296～301）

12 Wang Xiao-Feng, Fang Bin-Xing, Zhang Hong-Li *et al*. A model for estimating the performance of synchronous parallel network simulation. In: Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing and Systems, Phoenix, Arizona, USA, 2005, 548～553

13 Wang Xiao-Feng, Fang Bin-Xing, Yun Xiao-Chun *et al*. Approach for topology partitioning in parallel network simulation. Journal on Communications, 2006, 27(2): 16～21(in Chinese)
（王晓锋，方滨兴，云晓春等. 并行网络模拟中的一种拓扑划分方法. 通信学报，2006，27(2)：16～21）

14 Zou Ling, Shi Bing-Xin, Zhao Er-Dun *et al*. An application of genetic algorithm in the optimization of computer network segmentation. Journal of China Institute of Communications, 1999, 20(4): 42～47(in Chinese)
（邹　玲，石冰心，赵尔敦等. 遗传算法在计算机网络划分中的应用. 通信学报，1999，20(4)：42～47）

15 Ricci R., Alfeld C., Lepreau J.. A solver for the network testbed mapping problem. SIGCOMM Computer Communications Review, 2003, 33(2): 65～81

16 Magoni D.. Nem: A software for network topology analysis and modeling. In: Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, Fort Worth, TX, USA, 2002, 364～371

WANG Xiao-Feng, born in 1978, Ph. D. candidate. His research interests include network simulation, network security.

FANG Bin-Xing, born in 1960, Ph. D., professor, Ph. D. supervisor, member of Chinese Academy of Engineering. His research interests include network and information security, parallel computing.

YUN Xiao-Chun, born in 1971, Ph. D., professor, Ph. D. supervisor. His research interests include computer network, network security.

ZHANG Hong-Li, born in 1973, Ph. D., professor, Ph. D. supervisor. Her research interests include network and information security, parallel processing.

Background

This work is mainly supported by the National Basic Research Program of China (973 Program) under grant No. 2005CB321806 and the Program for New Century Excellent Talents in University (NCET-04-0338). One of the aims of the project is to develop an efficient and accurate simulation platform for doing research on large-scale network security incidents, such as network worms and DDoS attacks. The project group has developed a practicable parallel network simulation platform, which can be easily used to simulate network worms and DDoS attacks, and has established a model for estimating the performance of parallel network simulation.

Network simulation is an efficient method to do research on network worms, and parallel network simulation is the key method to satisfy the requirement of large-scale network simulation of worms. A key problem for parallel simulation of worms is how to partition the simulation task reasonably in order to improve the performance of worm simulation efficiently. Traditional method for partitioning the simulation task is to partition the topology for simulation based on graph partitioning tools. As there are shortcomings of partitioning method based on graph partitioning tools, this paper puts forward a new method for optimized partitioning for parallel simulation of network worms, which can improve the performance of worm simulation efficiently. The work is the important part of the efforts to develop an efficient and accurate simulation platform for doing research on large-scale network security incidents.