

Blogel-SPINAL: 分布式 PPI 网络比对算法

周文剑¹, 靳 婷²

(1. 复旦大学计算机科学技术学院, 上海 200433; 2. 海南大学信息科学技术学院, 海口 570228)

摘 要: 高通量技术的发展使蛋白质-蛋白质相互作用(PPI)网络的规模日益增大, 需要高速算法对其进行全局比对。为此, 分析集中式全局比对算法 SPINAL, 将该算法中耗时超过 95% 的计算估计值阶段移植到分布式平台 Blogel 下运算, 求比对图阶段则仍保持集中式运算, 以此得到 Blogel-SPINAL 算法。理论分析和实验结果表明, 与 SPINAL 相比, Blogel-SPINAL 能提升比对速度, 具有较好的扩展性。

关键词: 分布式计算; 蛋白质-蛋白质相互作用网络; 全局网络比对; 积图; 顶点划分; 可扩展性

中文引用格式: 周文剑, 靳 婷. Blogel-SPINAL: 分布式 PPI 网络比对算法[J]. 计算机工程, 2016, 42(4): 209-214, 220.

英文引用格式: Zhou Wenjian, Jin Ting. Blogel-SPINAL: Distributed PPI Network Alignment Algorithm[J]. Computer Engineering, 2016, 42(4): 209-214, 220.

Blogel-SPINAL: Distributed PPI Network Alignment Algorithm

ZHOU Wenjian¹, JIN Ting²

(1. School of Computer Science, Fudan University, Shanghai 200433, China;

2. College of Information Science and Technology, Hainan University, Haikou 570228, China)

【Abstract】 With the development of high throughput technology, the size of Protein-protein Interaction (PPI) network grows larger and larger, thus high speed PPI network global alignment algorithm is needed. Aiming at this problem, this paper analyzes the SPINAL algorithm. It modifies the coarse-grained phase which occupies more than 95% time cost to adapt for distributed platform named Blogel, and keeps the fine-grained phase unchanged. Hence, the Blogel-SPINAL algorithm is gotten. Theoretical analysis and experimental results show that Blogel-SPINAL is much faster than SPINAL, and its scalability is better.

【Key words】 distributed computation; Protein-protein Interaction (PPI) network; global network alignment; product graph; vertex partition; scalability

DOI: 10.3969/j.issn.1000-3428.2016.04.037

1 概述

蛋白质是生命体的物质基础, 是构成细胞的基本有机物, 直接参与了生命活动中几乎所有的进程。这些进程极少由单个蛋白质孤立工作来完成, 大多是通过蛋白质彼此之间相互作用来实现^[1]。一个生物体内的所有蛋白质相互作用形成的网络被称作蛋白质-蛋白质相互作用 (Protein-protein Interaction, PPI) 网络^[2]。PPI 网络可以视为无权无向图, 蛋白质是图的顶点, 相互作用是图的边。

PPI 网络比对是将 2 个或多个不同物种的 PPI 网络中的蛋白质进行一对一的映射。比对的结果可以用来进行知识迁移, 通过已知生物来研究未知生

物, 例如蛋白质相互作用的预测^[3]、蛋白质功能预测^[4-5]以及不同物种的蛋白质同源关系预测^[6]。

目前已有很多 PPI 网络比对算法^[7]。网络比对的算法可以分成局部比对和全局比对 2 种框架, 其中局部网络比对算法大多是寻找 2 个或多个 PPI 网络中相互匹配的子图, 允许一个蛋白质被匹配到多个蛋白质, 可能会产生歧义, 因此, 近年来的比对算法大多是全局网络比对算法。

全局网络比对算法最常见的计算模型是: 首先建立一个融合了所有 PPI 网络的积图, 积图的每个顶点是来自各个网络的一组蛋白质, 通常是同源蛋白质。积图中 2 个顶点对应的 2 组蛋白质如果在各自的网络中都有边相连, 则在积图中这 2 个顶点也

基金项目: 国家“863”计划基金资助项目 (2012AA020403)。

作者简介: 周文剑 (1989 -), 男, 硕士, 主研方向为生物信息学; 靳 婷 (通讯作者), 讲师、博士。

收稿日期: 2015-04-20 **修回日期:** 2015-05-13 **E-mail:** tingj@fudan.edu.cn

有边相连;然后定义某种打分函数对积图中的顶点或边进行打分;最后用启发式的方法从积图中找到一个比对图,比对图中任意 2 个顶点包含的 2 组蛋白质不会出现重叠,且比对图中顶点和边的总分值较大。虽然建立的积图中顶点和边的条件不同,有的算法没有显式地建立积图,打分函数的定义和算法不同,比对图的搜寻策略也不同,但 Græmlin 算法^[8]、IsoRank 算法^[6]、文献[9]算法、GRAAL 算法^[10]、MI-GRAAL 算法^[11]、C-GRAAL 算法^[12]和 SPINAL 算法^[13]都可以看成是这个模型。除了这个模型之外,文献[14]提出了一种将网络比对的问题规约为非线性整数规划问题来求解的算法。另外, PISwap 算法^[15]是先求出一个最初的映射,然后进行多次迭代,每次根据定义的打分函数交换一部分蛋白质的映射关系以优化总得分,最后求出一个局部最优解。

近年来,随着酵母双杂交技术^[16]、串联亲和纯化技术^[17]、质谱分析技术^[18]、蛋白质芯片技术^[19]等高通量技术的发展,大量可用的蛋白质相互作用数据在互联网上公开发布,这为研究 PPI 网络提供了方便,同时也使得 PPI 网络的规模越加庞大,因此,需要高速的算法进行网络比对。

PPI 网络是无权无向图,很多全局网络比对算法也都是图论算法,而 Blogel^[20]正是一种分布式图计算平台,将集中式算法移植到 Blogel 平台就可以进行加速。在上述全局网络比对算法中,SPINAL 是一个比对结果相当优秀的算法,而且该算法的框架也很契合 Blogel,因此,本文对 SPINAL 进行修改,将其移植到 Blogel 平台,从而得到一种改进算法 Blogel-SPINAL。

2 Blogel 平台

Blogel 平台具有以下特点:

(1) Blogel 遵循 BSP 模型^[21],即“计算-通信-同步”模型。Blogel 的整个计算过程由若干个顺序执行的“超级步”组成,每个“超级步”始于“计算”,每个进程独立进行计算;然后是“通信”,每个进程计算完可以向其他进程发送消息;最后是“同步”,同步完成后系统将从一个“超级步”迈向下一个“超级步”,直到达到算法的终止条件。

(2) Blogel 以块和顶点为中心。由于 Blogel 是分布式平台,因此执行算法时有很多个进程,被称作工作节点,工作节点是粗粒度的计算单元,在工作节点下还有块和顶点这 2 个细粒度的计算单元,它们拥有各自的属性,边的信息一般就存储在对应顶点的属性中。在预处理阶段,Blogel 将顶点划分成块,因此,块的属性中包括属于它的所有顶点的属性,再

将块分配到工作节点,形成了“顶点-块-工作节点”3 层结构。在每个“超级步”中,先是每个顶点独立地进行计算,然后是每个块独立地进行计算,顶点和块的计算函数由用户自定义,它们在计算时可以查看和修改本身的属性,并向其他顶点和块发送消息。这些消息经过同步后会在下一个“超级步”被对应的顶点和块接收。

(3) Blogel 的终止条件:Blogel 的块和顶点的属性中包含状态信息,即“活跃”状态与“不活跃”状态。在第 1 个“超级步”中,所有块和顶点都处于“活跃”状态。在每个“超级步”中,只有处于“活跃”状态的块和顶点才会进行计算,计算时可以调用 vote_to_halt 函数转换到“不活跃”状态。“不活跃”的块和顶点在收到消息后会重新转换到“活跃”状态。当所有块和顶点都处于“不活跃”状态后,整个计算结束。

3 SPINAL 算法

SPINAL 算法是一种对 2 个 PPI 网络进行全局比对的算法。设 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$ 是 2 个 PPI 网络,首先构造它们的积图 $G_{12} = (V_{12}, E_{12})$,积图的顶点 $\langle v_1, v_2 \rangle \in V_{12}$ 是由来自 2 个 PPI 网络的一对顶点组合而成,所以, $V_{12} = V_1 \times V_2$ 。这样,积图中的每个顶点对应于 2 个网络中的一对顶点,反之亦然;积图的边 $(\langle v_1, v_2 \rangle, \langle u_1, u_2 \rangle) \in E_{12}$ 是由来自 2 个 PPI 网络的一对边组合而成,即满足 $(v_1, u_1) \in E_1$ 且 $(v_2, u_2) \in E_2$ 。从积图中可以求出比对图 $A_{12} = (V_A, E_A)$,比对图的顶点保证不重叠,即 $\forall \langle v_1, v_2 \rangle = \langle u_1, u_2 \rangle \in V_A, v_1 = u_1$ 且 $v_2 = u_2$ 。比对图的边则包含 G_{12} 中所有由 V_A 中顶点组成的边。由于比对图的顶点不会出现重叠,因此一个比对图可以对应一个全局比对,而这个全局比对的得分定义为:

$$GNAS(A_{12}) = \alpha \times |E_A| + (1 - \alpha) \times \sum_{\langle v_1, v_2 \rangle \in V_A} seq(v_1, v_2) \quad (1)$$

其中, $seq(v_1, v_2)$ 是一个反映 v_1 和 v_2 对应蛋白质序列相似度的函数,可以用归一化的 BLAST bit-score 或 BLAST E-value 来度量。因此, $\alpha \in [0, 1]$ 是一个控制参数,用于控制拓扑结构相似度和蛋白质序列相似度占得分的比重。由于求出得分最大的比对图是 NP 难问题^[13],因此 SPINAL 仅求出一个得分较大的比对图,具体见算法 1。

算法 1 SPINAL 算法

输入 PPI 网络 $G_1 = (V_1, E_1)$ 和 $G_2 = (V_2, E_2)$, seq, α
输出 比对图 G_A
//计算估计值阶段
1. 计算 P 的初值
2. repeat

```

3.   P' ← P
4.   for each < v1, v2 > ∈ V12
5.       构造 < v1, v2 > 的邻居二分图
6.       求出邻居二分图的贪心匹配 C
7.       根据 P' 和 C 求出新的 P(v1, v2)
8.   until 足够次迭代
//求比对图阶段
9.   repeat
10.      取估计值最大的未比对上的顶点对加入 A12
11.      while VA 的邻居二分图不为空
12.          构造 VA 的邻居二分图
13.          求出邻居二分图的最大权匹配 C
14.          计算邻居二分图每组顶点对的实际得分, 对
C 进行修正
15.          将修正后 C 中对应的顶点对加入 VA, 连接
的边加入 EA
16.   until |VA| = min(|V1|, |V2|)

```

整个 SPINAL 算法可以分成以下 2 个阶段:

阶段 1 计算估计值阶段, 求出每个顶点的估计值, 即粗粒度阶段。

对积图中的每个顶点 $\langle v_1, v_2 \rangle$ 定义一个得分 $P(v_1, v_2)$ 作为估计值, 用来估计顶点 $\langle v_1, v_2 \rangle$ 加入比对图 A_{12} 后总得分增加的幅度, 从而求出每个顶点的估计值。为了计算这个估计值, 先定义顶点 $\langle v_1, v_2 \rangle$ 的邻居二分图, 用 $Nei_{G_1}(v_1)$ 和 $Nei_{G_2}(v_2)$ 分别表示 v_1 和 v_2 在图 G_1 和 G_2 中的邻居, 那么用 $Nei_{G_1}(v_1) \cup Nei_{G_2}(v_2)$ 作为二分图的顶点, 用 $P(u_1, u_2)$, $u_1 \in Nei_{G_1}(v_1)$, $u_2 \in Nei_{G_2}(v_2)$ 作为二分图中边的权重就可以构造 $\langle v_1, v_2 \rangle$ 的邻居二分图。设 C 为邻居二分图的最大全匹配, 则估计值 $P(v_1, v_2)$ 定义为:

$$P(v_1, v_2) = \alpha \times \frac{\sum_{(u_1, u_2) \in C} \frac{P(u_1, u_2)}{|Nei_{G_1}(u_1)| \times |Nei_{G_2}(u_2)|}}{\sqrt{C}} + (1 - \alpha) \times seq(v_1, v_2) \quad (2)$$

因此, 顶点 $\langle v_1, v_2 \rangle$ 的估计值要通过积图中邻居的估计值求出。

首先定义估计值 $P(v_1, v_2)$ 的初值:

$$P = \alpha \times |Nei_{G_1}(v_1) - Nei_{G_2}(v_2)| + (1 - \alpha) \times seq(v_1, v_2) \quad (3)$$

然后进行多次迭代, 用上一次迭代后的估计值求出新的估计值, 经过足够次迭代后估计值会收敛, 即使是规模较大的积图也会在 10 次 ~ 15 次迭代后收敛^[13]。另外, 由于贪心匹配和最大权匹配求出的估计值差别不大, 而贪心匹配比最大权匹配快很多, 因此可以选择通过贪心匹配来计算估计值^[13]。设 Δ_1 和 Δ_2 分别表示 2 个 PPI 网络顶点的最大度, 则此阶段每次迭代的理论时间复杂度为 $O(|V_1| |V_2| \Delta_1 \Delta_2 \lg(\Delta_1 \Delta_2))$ 。由于大部分 PPI 网络属于无标度 (scale-free) 网络^[22],

多数顶点的度都较小, 因此实际时间复杂度远小于理论值。阶段 1 的伪代码见算法 1 的第 1 行 ~ 第 8 行。

阶段 2 求比对图阶段, 利用估计值求出一个得分较高的比对图, 即细粒度阶段。

比对图 $A_{12} = (V_A, E_A)$ 初始为空, 然后逐渐向图中添加顶点和边, 每向 A_{12} 加入一个顶点, 也就分别比对上来自 G_1 和 G_2 中的一对顶点。

此阶段就是每次从积图中选择一个对应顶点对 v_1 和 v_2 都未被比对上, 且估计值最大的顶点 $\langle v_1, v_2 \rangle$ 加入比对图, 然后以这个顶点为起点向周围扩张, 将积图中更多的顶点加入比对图, 直到无法扩张时再选择新的起点开始扩张。最后当 $|V_A| = \min(|V_1|, |V_2|)$ 时, 即完成全局比对任务。

扩张部分通过重复以下 4 个步骤完成:

(1) 构造 V_A 的邻居二分图, 与单个顶点的邻居二分图类似。找到 V_A 中所有顶点在 G_1 和 G_2 中对应顶点的邻居, 去除其中已经比对上的, 将剩余的顶点作为邻居二分图的顶点, 而对应的估计值作为边权。

(2) 求出邻居二分图的最大权匹配 C 。

(3) P 始终只是估计值, 由于部分比对图已经求出, 因此可以求出二分图中每组顶点对的部分实际得分, 然后将每组不在匹配 C 中的顶点对实际得分与 C 中重叠的顶点对进行比较, 如果不在 C 中的顶点对实际得分较高, 则可通过交换对 C 进行修正。

(4) 将修正后的匹配 C 对应的积图顶点加入 V_A , 积图中相连的边也同时加入 E_A 。

假设 $N = \max(|V_1|, |V_2|)$, 则阶段 2 的理论时间复杂度为 $O(|V_1| |V_2| \Delta_1 \Delta_2 + N^2 \lg N)$, 同样由于 PPI 网络的无标度特性, 这个阶段的实际时间复杂度也远小于理论值。因为这个阶段时间开销很小, 多数情况不到 SPINAL 算法总开销的 5%^[13], 所以在扩张部分的第 (2) 步求最大权匹配, 以得到更好的匹配 C 。阶段 2 的伪代码见算法 1 的第 9 行 ~ 第 16 行。

4 Blogel-SPINAL 算法

4.1 算法描述

由于 Blogel 平台以每个块和顶点作为计算单元, 因此基于 Blogel 的算法需要给出每个块和顶点的属性以及如何利用这些属性计算出结果。

SPINAL 算法的计算估计值阶段首先将估计值进行初始化, 然后进行了多次迭代, 每次迭代对积图上每个顶点利用邻居的原估计值计算出新估计值, 因此, 基于 Blogel 的算法同样在积图上进行计算。

朴素的分布式算法为: 顶点的属性 $\langle v_1, v_2 \rangle$ 中

包含 $Nei_{G_1}(v_1)$, $Nei_{G_2}(v_2)$, $seq(v_1, v_2)$ 和 $P(v_1, v_2)$, 其中, $Nei_{G_1}(v_1)$, $Nei_{G_2}(v_2)$ 和 $seq(v_1, v_2)$ 直接从输入读取; $P(v_1, v_2)$ 的初值在读取邻居信息和相似度后直接计算得到, 积图上的邻居可通过 $Nei_{G_1}(v_1) \times Nei_{G_2}(v_2)$ 得到。在第 1 个“超级步”中, 每个顶点向所有邻居顶点发送消息告知估计值; 从第 2 个“超级步”开始, 每个顶点都可以从消息中获得所有邻居的原估计值, 再加上本身属性中的 seq , 所有顶点都可以独立计算出各自的新估计值, 然后向所有邻居发送消息告知估计值。经过足够次数的“超级步”, 估计值都得到收敛, 即可停止运算。如果所有顶点最终被均匀分配到所有 W 个工作节点, 则每个“超级步”的运算时间复杂度为 $O(|V_1| |V_2| \Delta_1 \Delta_2 \lg(\Delta_1 \Delta_2) / W)$, 通信时间复杂度为 $O(|V_1| |V_2| \Delta_1 \Delta_2)$ 。

朴素算法完全只进行顶点计算, 没有考虑块计算, 并且 SPINAL 算法的求比对图阶段要多次从积图中的一个顶点出发, 按照广度优先的层次结构进行扩张, 最后得到一个比对图, 不适合将计算拆分到每个顶点或是每个块进行独立计算, 无法移植到 Blogel 平台。为此, 本文提出 Blogel-SPINAL 算法。首先, 估计值的计算可以直接移到块计算函数中, 这样在同一个块的顶点的估计值可以直接获取, 不需要经过通信; 其次, 邻居信息也可以移到块属性中, 这样每个块中最多只需要存储 PPI 网络中每个顶点的邻居信息一份, 减少了冗余度; 最后, 计算完估计值以后, 不需要向所有邻居各发送一条消息, 只需要向所有包含邻居的块发送一条消息。包含邻居的块可以在 Blogel 的划分与分配阶段中计算得到, 并存储到每个顶点的属性中。设总块数为 B , 则改进后的运算时间复杂度不变, 通信时间复杂度降到 $O(|V_1| |V_2| \min(\Delta_1 \Delta_2, B))$ 。Blogel-SPINAL 只进行块计算, 其计算函数见算法 2。

算法 2 Blogel-SPINAL 求估计值阶段块计算函数

```
//对块 b 进行计算
//step_num 表示当前“超级步”的步数
1. if step_num > 1
    //每条消息中包含发送消息的顶点  $\langle u_1, u_2 \rangle$  及其
//估计值 p
2.   for each  $\{ \langle u_1, u_2 \rangle, p \} \in$  收到的消息
3.      $P'(\langle u_1, u_2 \rangle) \leftarrow p$ 
4.    $P'(\langle v_1, v_2 \rangle) \leftarrow P(\langle v_1, v_2 \rangle), \langle v_1, v_2 \rangle \in b$ 
5.   for each  $\langle v_1, v_2 \rangle \in b$ 
6.     构造  $\langle v_1, v_2 \rangle$  的邻居二分图
7.     求出邻居二分图的贪心匹配 C
8.     根据  $P'$  和 C 求出新的  $P(\langle v_1, v_2 \rangle)$ 
9.   if step_num 不到足够次数
```

```
10.   for each  $\langle v_1, v_2 \rangle \in b$ 
11.     for each  $c \in$  包含邻居的块
12.       向块 c 发送消息  $\{ \langle v_1, v_2 \rangle, P(\langle v_1, v_2 \rangle) \}$ 
13.   else
14.     vote_to_halt
```

Blogel-SPINAL 算法的整体流程如下:

(1) 预处理阶段。将积图 G_{12} 的顶点划分成块, 再将块分配到工作顶点, 同时对每个顶点计算出包含邻居的块。

(2) 计算估计值阶段。在 Blogel 平台下通过多个“超级步”计算出积图每个顶点的估计值, 见算法 2。

(3) 求比对图阶段。同 SPINAL 算法一样, 在集中式环境下求出比对图 A_{12} , 完成 2 个 PPI 网络 G_1 和 G_2 的全局比对, 见算法 1 的第 9 行 ~ 第 16 行。

4.2 划分与分配策略

Blogel-SPINAL 算法的预处理阶段是将积图的顶点先划分成块并分配到工作节点。划分与分配的策略会对实际的通信开销和工作节点负载均衡程度造成很大影响, 其中通信开销正比于发送消息的数目。而计算一个顶点 $\langle v_1, v_2 \rangle$ 的估计值时间复杂度为 $O(|Nei_{G_1}(v_1)| |Nei_{G_2}(v_2)| \lg(|Nei_{G_1}(v_1)| |Nei_{G_2}(v_2)|))$, 可以用这个上界来度量每个顶点的时间开销, 将划分到一个块所有顶点的计算时间开销相加, 就是这个块的计算时间开销; 将分配到一个工作节点的所有块的时间开销相加, 就是这个工作节点的负载压力, 在完美的负载均衡状态下, 所有工作节点负载压力一致。

策略中首先要考虑块的数目: 块的数目影响通信开销, 块越少, 每个块上的顶点越多, 对于每个顶点来说, 属于同一块的邻居也越多, 包含邻居的其他块也越少。所以, 块的数目越少越好, 而块的数目必须大于等于工作节点数目, 因此, 最好的选择是将顶点划分成同工作节点数目 W 相同的块数, 然后向每个工作节点分配一个块。

接下来就要考虑如何将顶点划分成块, 由于 PPI 网络大多是由一个大连通分量和数个小连通分量组成, 它们的积图也会由一个大连通分量和数个小连通分量组成, 因此可以先将大连通分量中的顶点划分成 W 块, 然后采用贪心策略将小连通分量并入 W 个大块: 将每个小连通分量划分到一个小块, 将小块按计算时间开销从大到小排序, 然后依次加入当前时间开销最少的大块。这样, 计算小连通分量上的顶点估计值时就不存在通信开销。

对于划分大连通分量, 一种比较简单的划分策略是 GVD 策略^[23], 其时间复杂度为 $O(|E_{12}|)$, 但 GVD 策略随机性很强, 划分的块太少很容易产生负载不均衡的现象。为了均衡负载, 可以用 GVD + 贪

心策略进行划分: 首先用 GVD 策略将顶点划分成 B 个小块; 然后将每个小块按负载从大到小排序; 最后依次将所有小块分配到当期负载最小的 W 个大块中。贪心策略的时间复杂度为 $O(B \lg B)$, 由于 $B \ll |E_{12}|$, 因此 GVD + 贪心策略的时间复杂度仍为 $O(|E_{12}|)$ 。

另一种划分策略是 METIS 策略^[24], 此策略的时间复杂度较高, 但可以使得划分后每个块的顶点权值之和相近且连接不同块中顶点的边权之和尽量少。将顶点的权重赋为计算时间开销, 将所有边的权重赋为 1, 则用 METIS 策略可以得到较好的划分结果。但是 METIS 策略的时间复杂度较高, 直接对积图进行划分耗时较大。所以, 可以采用 GVD + METIS 策略: 先用 GVD 策略将顶点划分成 B 个小块, 然后计算小块的权值和邻接关系, 再用 METIS 策略将小块分配到 W 个大块。同样由于 $B \ll |E_{12}|$, GVD + METIS 策略的时间复杂度也是 $O(|E_{12}|)$ 。

总体划分和分配策略就是先采用 GVD + 贪心或 GVD + METIS 策略将大连通分量划分成 W 个块, 然后再采用贪心策略将小连通分量并入 W 个块, 最后将 W 个块一对一地分配到 W 个工作节点上。

5 实验与结果分析

5.1 实验环境

集中式算法 SPINAL 的实验在 1 台 Intel(R) Core(TM)2 Duo CPU E7500@2.93 GHz CPU, 2 GB 可用内存并安装 Ubuntu 10.10 的机器上进行。分布式算法 Blogel-SPINAL 的实验在数台同样配置的机器组成的集群上进行。Blogel 平台将机器分为 1 个主节点(Master)和数个次节点(Slave), 由于机器都是双核, 因此在每个次节点上运行 2 个工作节点, 主节点由于任务较重, 仅运行 1 个工作节点。

5.2 数据集

实验数据来自 IsoBase^[25], 数据集中包含 C. elegans,

D. melanogaster, H. Sapiens, M. musculus 和 S. cerevisiae 这 5 个物种的 PPI 网络以及蛋白质之间的 BLAST bit-score, 用 ce, dm, hs, mm 和 sc 分别表示这 5 个物种。去除重边之后各个 PPI 网络的规模见表 1。全局网络比对的实验在 mm-dm, mm-hs, mm-sc, ce-dm, ce-hs 和 ce-sc 这 6 对 PPI 网络上进行。

表 1 PPI 网络规模

物种	顶点数	边数	最大连通分量顶点数	连通分量数
mm	623	575	167	150
ce	2 995	4 855	2 722	123
dm	7 396	24 995	7 284	57
hs	10 403	54 911	10 153	169
sc	5 524	82 794	5 523	2

5.3 实验结果

本文进行了比较实验和可扩展性实验, 对 Blogel-SPINAL 多个方面进行测试。

5.3.1 比较实验

本实验比较 SPINAL 算法和采用 2 种划分策略的 Blogel-SPINAL 算法总耗时以及 Blogel-SPINAL 算法的各阶段耗时。在进行比较实验时, Blogel 平台下使用了 5 台机器运行 9 个工作节点。由于 GVD 策略是一种随机策略, 因此在每组数据上运行了 5 次, 取平均值计算耗时, 3 个算法的总耗时与求比对图阶段的耗时见表 2。可以看出, 采用 2 种策略的 Blogel-SPINAL 算法在所有数据上都比 SPINAL 算法快, 这证明了 Blogel-SPINAL 算法确实能有效地提升全局网络比对的速度。另外, 算法中求比对图阶段的耗时占总耗时的比例都相当低, 而且对于最大连通分量的顶点数占总顶点数比重越大的网络, 这个阶段的耗时占总耗时的比例越低, 所以, 将这个阶段放在集中式环境下运行并不会成为 Blogel-SPINAL 算法的瓶颈。

表 2 总耗时与求比对图阶段耗时

PPI 网络对	总耗时			求比对图阶段耗时
	SPINAL 算法	GVD + 贪心策略	GVD + METIS 策略	
mm-dm	83.89	51.52	50.59	18.81
mm-hs	166.72	86.13	82.43	27.27
mm-sc	183.33	77.12	69.98	7.09
ce-dm	589.10	379.18	371.86	7.42
ce-hs	1233.50	726.86	717.13	11.67
ce-sc	1 716.96	861.23	835.25	6.96

采用 2 种策略的 Blogel-SPINAL 算法在前 2 个阶段的耗时情况比较如表 3 所示。可以看出, 采用 GVD + METIS 策略的预处理阶段耗时比 GVD + 贪心策略稍长一点, 这是因为 GVD + METIS 策略在用 GVD 策略划分出小块以后还要花一点时间计算小块的权值和邻接关系; 而计算估计值阶段的通信开销明显 GVD

+ METIS 策略少很多, 这是因为 METIS 策略会尽量减少连接不同块的边, 比起 GVD + 贪心策略自然有明显优势; 2 种策略下计算估计值阶段的运算开销差不多, 这是因为 2 种策略都注意负载均衡。因此, 综合来看, 采用 GVD + METIS 策略的 Blogel-SPINAL 算法速度更快。

表 3 Blogel-SPINAL 算法各阶段耗时

s

PPI 网络对	预处理阶段耗时		计算估计值阶段通信耗时		计算估计值阶段运算耗时	
	GVD + 贪心策略	GVD + METIS 策略	GVD + 贪心策略	GVD + METIS 策略	GVD + 贪心策略	GVD + METIS 策略
mm-dm	14.24	15.26	5.70	3.07	12.78	13.45
mm-hs	20.61	21.82	8.82	3.77	29.43	29.57
mm-sc	11.13	11.66	7.39	3.24	51.51	47.99
ce-dm	127.49	140.74	78.47	62.38	165.80	161.32
ce-hs	186.77	210.47	129.96	93.03	398.47	401.97
ce-sc	110.19	127.41	95.91	55.49	648.16	645.39

5.3.2 可扩展性实验

可扩展性实验测试采用 GVD + METIS 策略的 Blogel-SPINAL 算法在工作节点数目不同情况下的耗时变化。由于求比对图阶段的耗时不受工作节点数目影响,因此只考虑前两个阶段的耗时变化,实验结果如图 1 所示。可以看出,当工作节点数目为 3 时,Blogel-SPINAL 算法的耗时已经小于 SPINAL 算法。随着工作节点数目的增加,预处理阶段的耗时稍有增长,而计算估计值阶段的耗时则大幅度减少,因此,前两个阶段的总耗时也一直在减少,可见 Blogel-SPINAL 算法的扩展性较好。

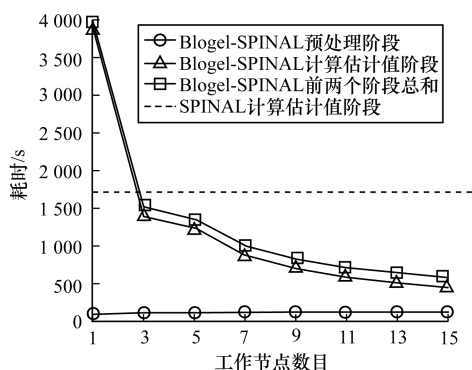


图 1 不同数目工作节点下的耗时比较

但随着工作节点的增多,预处理阶段耗时占总耗时的比重越来越大,逐渐成为整个算法速度上的瓶颈,因此,下一步还需要对划分和分配策略加以改进,使整个算法总耗时进一步降低。

6 结束语

本文对集中式全局网络比对算法 SPINAL 进行修改,将其移植到分布式平台 Blogel 上,给出了 Blogel-SPINAL 算法,该算法可得到与 SPINAL 算法相当的比对结果,同时还大幅提升了比对速度。

Blogel-SPINAL 算法可以分成预处理、计算估计值和求比对图 3 个阶段,其中计算估计值阶段耗时较大,可以通过增加 Blogel 系统的工作节点来降低此阶段的耗时;而求比对图阶段耗时较少,对总耗时影响不大;但预处理阶段的耗时随着工作节点增多

反而会缓慢增长,因此,下一步将对预处理阶段加以改进,以降低算法的总耗时。

参考文献

- [1] von Mering C, Krause R, Snel B, et al. Comparative Assessment of Large-scale Data Sets of Protein-protein Interactions[J]. Nature, 2002, 417(6887): 399-403.
- [2] Fields S, Song O. A Novel Genetic System to Detect Protein-protein Interactions[J]. Nature, 1989, 340(6230): 245-246.
- [3] Liang Zhi, Xu Meng, Teng Maikun, et al. Comparison of Protein Interaction Networks Reveals Species Conservation and Divergence[J]. BMC Bioinformatics, 2006, 7(1): 457.
- [4] Koldř M, Lässig M, Berg J. From Protein Interactions to Functional Annotation: Graph Alignment in Herpes[J]. BMC Systems Biology, 2008, 2(1): 90.
- [5] Sharan R, Suthram S, Kelley R M, et al. Conserved Patterns of Protein Interaction in Multiple Species[J]. Proceedings of the National Academy of Sciences, 2005, 102(6): 1974-1979.
- [6] Singh R, Xu Jinbo, Berger B. Global Alignment of Multiple Protein Interaction Networks with Application to Functional Orthology Detection[J]. Proceedings of the National Academy of Science, 2008, 105(35): 12763-12768.
- [7] 郭杏莉, 高琳, 陈新. 生物网络比对的模型与算法[J]. 软件学报, 2010, 21(9): 2089-2106.
- [8] Flannick J, Novak A, Srinivasan B S, et al. Graemlin: General and Robust Alignment of Multiple Large Interaction Networks[J]. Genome Research, 2006, 16(9): 1169-1181.
- [9] Milenković T, Ng W L, Hayes W, et al. Optimal Network Alignment with Graphlet Degree Vectors[J]. Cancer Informatics, 2010, 9: 121-137.
- [10] Kuchaiev O, Milenković T, Memišević V, et al. Topological Network Alignment Uncovers Biological Function and Phylogeny[J]. Journal of the Royal Society Interface, 2010, 7(50): 1341-1354.
- [11] Kuchaiev O, Pržulj N. Integrative Network Alignment Reveals Large Regions of Global Network Similarity in Yeast and Human[J]. Bioinformatics, 2011, 27(10): 1390-1396.
- [12] Memišević V, Pržulj N. C-GRAAL: Common-neighbors-based Global GRAPh ALignment of Biological Networks[J]. Integrative Biology, 2012, 4(7): 734-743.

(下转第 220 页)

溯树的基础上引入用户请求输入闭包,能够有效判断用户请求的可满足性,而且在对可满足的用户请求进行处理的过程中,能够缩小搜索空间并合理裁剪回溯树。仿真实例结果表明,该算法将搜索空间限制在用户请求输入闭包中,从而处理用户请求时,可降低搜索空间、处理时间及成本,简化回溯树的构建过程,提高服务组合效率。今后的研究工作包括:(1)采用更有效的方式描述 Web 服务的语义信息,根据语义信息合理构建本体概念树;(2)在服务组合方法中,将 Web 服务的非功能属性如 QoS、负载均衡^[17]等问题纳入考虑范围,增强组合结果的可用性。

参考文献

- [1] Cardoso J, Sheth A. Semantic E-workflow Composition[J]. Journal of Intelligent Information System, 2003, 12(3): 191-225.
- [2] 胡海洋, 李 刚, 韩燕波. 一种面向业务用户的大粒度服务组合方法[J]. 计算机学报, 2005, 28(4): 694-703.
- [3] Rao Jinghai, Kungas P, Matskin M. Logic-based Web Services Composition: From Service Description to Process Model [C]//Proceedings of International Conference on Web Services. Washington D. C., USA: IEEE Computer Society, 2004: 446-453.
- [4] Zhao Haibo, Doshi P. A Hierarchical Framework for Composing Nested Web Processes [C]//Proceedings of International Conference on Service-oriented Computing. Berlin, Germany: Springer, 2006: 116-128.
- [5] 邱莉榕, 史忠植, 林 芬, 等. 基于主体的语义 Web 服务自动组合研究[J]. 计算机研究与发展, 2007, 44(4): 643-650.
- [6] 邓水光, 吴 健, 李 莹, 等. 基于回溯树的 Web 服务自动组合[J]. 软件学报, 2007, 18(8): 1896-1910.
- [7] 李瑞宁, 周竹荣. 基于完备回溯树的语义 Web 服务自动组合[J]. 计算机应用, 2008, 28(6): 1427-1430.
- [8] 周竹荣. 基于语义的教学网格门户研究[D]. 重庆: 西南大学, 2008.
- [9] 晏丽萍. Web 服务自动组合[D]. 重庆: 西南大学, 2012.
- [10] 罗晓娟, 牛秦洲, 叶恒舟. 一种优化的基于回溯树的 Web 服务自动组合方法[J]. 大众科技, 2010, (4): 28-31.
- [11] 张忠平, 赵海亮, 张志惠. 基于本体的概念相似度计算[J]. 计算机工程, 2009, 35(7): 17-19.
- [12] Li Yuhua, Bandar Z. An Approach for Measuring Semantic Similarity Between Words Using Multiple Information Sources[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(4): 871-882.
- [13] 徐完平. 基于语义 Web 服务自动组合的研究[D]. 武汉: 湖北工业大学, 2010.
- [14] 邓水光. Web 服务自动组合与形式化验证的研究[D]. 杭州: 浙江大学, 2007.
- [15] 刘家茂, 顾 宁, 施伯乐. 基于 Mediator 的 Web Services 无回溯反向链动态合成[J]. 计算机研究与发展, 2005, 42(7): 1153-1158.
- [16] 李利波, 刘明利. 一种改进的无回溯反向 Web 服务动态组合方法[C]//2011 年全国通信安全学术会议论文集. 北京: [出版者不详], 2011: 10-15.
- [17] 余建涛, 胡宏林, 金盛峣, 等. 移动负载均衡与移动鲁棒性优化的冲突协调[J]. 计算机工程, 2012, 38(1): 37-41.

编辑 陆燕菲

(上接第 214 页)

- [13] Aladağ A E, Erten C. SPINAL: Scalable Protein Interaction Network Alignment [J]. Bioinformatics, 2013, 29(7): 917-924.
- [14] Klau G W. A New Graph-based Method for Pairwise Global Network Alignment [J]. BMC Bioinformatics, 2009, 10(S1).
- [15] Chindelevitch L, Liao C S, Berger B. Local Optimization for Global Alignment of Protein Interaction Networks[M]//Altman R. Pacific Symposium on Biocomputing 2010 (BIOCOMPUTING 2010). [S. l.]: World Scientific Publishing Company, 2010: 123-132.
- [16] Uetz P, Giot L, Cagney G, et al. A Comprehensive Analysis of Protein-protein Interactions in Saccharomyces Cerevisiae[J]. Nature, 2000, 403(6770): 623-627.
- [17] Rigaut G, Shevchenko A, Rutz B, et al. A Generic Protein Purification Method for Protein Complex Characterization and Proteome Exploration[J]. Nature Biotechnology, 1999, 17(10): 1030-1032.
- [18] Ho Y, Gruhler A, Heilbut A, et al. Systematic Identification of Protein Complexes in Saccharomyces Cerevisiae by Mass Spectrometry [J]. Nature, 2002, 415(6868): 180-183.
- [19] Zhu Heng, Bilgin M, Bangham R, et al. Global Analysis of Protein Activities Using Proteome Chips[J]. Science, 2001, 293(5537): 2101-2105.
- [20] Yan Da, Cheng J, Lu Yi, et al. Blogel: A Block-centric Framework for Distributed Computation on Real-world Graphs [J]. Proceedings of the VLDB Endowment, 2014, 7(14): 1981-1992.
- [21] 任年海. 一个有效的并行模型——BSP 并行模型[J]. 计算机与现代化, 2006, (3): 34-36.
- [22] 李 敏, 陈建二, 王建新. 基于复杂网络理论的 PPI 网络拓扑分析[J]. 计算机工程与应用, 2008, 44(8): 20-22.
- [23] Erwig M, Hagen F. The Graph Voronoi Diagram with Applications[J]. Networks, 2000, 36(3): 156-163.
- [24] Karypis G, Kumar V. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs[J]. SIAM Journal on Scientific Computing, 1998, 20(1): 359-392.
- [25] Park D, Singh R, Baym M, et al. IsoBase: A Database of Functionally Related Proteins Across PPI Networks[J]. Nucleic Acids Research, 2011, 39(S1): 295-300.

编辑 金胡考