求解 SSSP 问题图运算的复制数据算法*

杨敬安

（合肥工业大学人工智能研究所　合肥　230009）

摘要　本文首先提出求解 SSSP 问题图运算的数据并行算法及复制数据算法，并把复制数据技术成功地用于求解 SSSP 问题图运算证明算法的有效性，然后计算并讨论复制数据算法对数据并行算法的加速，最后指出复制数据技术不仅能用于图象的快速分析，而且也能广泛地用于解各种图运算问题.

关键词　SSSP 问题，复制数据算法，数据并行算法，横向优先搜索算法，加速比.

Dijkstra 等提出的线性横向优先搜索算法是找出图的任意已知对顶点间最短路径的著名的串行算法[1]，但此算法不易并行实现，使运行速度受限. 为提高运行速度，R. C. Paige 等提出计算最短路径并行算法[2]，Floyd 对此算法进行改进使计算最短路径能够并行实现，其速度和串行算法相比有显著提高.[3]然而，到目前为止，求出从单源到每个顶点的最短路径的各种算法多半是基于 Dijkstra 算法思想，根据找出通向那个顶点的所有路径的最小成本，迭代地修改从此源到其它顶点的距离，直到这一距离稳定为止，因而这些算法的运行速度也不能满足某些领域的使用要求. 由于复制数据(Replicated Data)技术能把操作并行与数据并行紧密结合，利用大的处理机数组有效地处理小的数据结构，这样可把求解问题的主操作分解为一组子操作，并把每个子操作分配给一个 copy，使这些子操作能用数据结构的各种 copies 同时实现，避免了迭代修改过程，大大加快求解 SSSP 问题的速度，性能价格比有明显提高. 但是利用复制数据技术解图运算问题不如计算图象象素数组那样有效，而且用 SIMD(single instruction multiple data)结构实现又比较困难.[4]所以为克服求解 SSSP (single source shortest path)问题所遇到的困难，本文提出利用联结机实现求解 SSSP 问题图运算的数据并行算法及复制数据算法，并且导出复制数据算法对数据并行算法的加速.

1　SSSP 问题的描述

SSSP 问题定义为找出自单源顶点开始到图的每个顶点的最小加权路径问题，这是应用很广的图运算问题. 例如，如果图的顶点表示城市，而边界权重表示城市间的运输费用，那么自源顶点 A 的单源最短路径给出从城市 A 到其它每个城市的最小运输费用.

图是抽象的数据结构，由某些顶点(结点)和连结顶点对的边组成，并且可以是单向的.

* 作者杨敬安，1943 年生，教授，博士导师，主要研究领域为图象理解，模式识别，计算机视觉，人工智能与机器人等. 本文通讯联系人，杨敬安，合肥 230009，合肥工业大学人工智能研究所

本文 1995－10－09 收到修改稿

在有向图中,边仅在一个方向连结.[5,6]这里我们讨论假定边以双向连结顶点,每个边有相关的权重,此权重可对用其连结那些顶点的费用进行编码.

邻近矩阵和邻近表是图的 2 种通用的计算机表示. 对有 n 个顶点的图,如果顶点 i 和 j 由边连结,邻近矩阵是使(i,j)项置 1 的 $n\times n$ 位数组,而另一个 $n\times n$ 数组存储与边有关的权重. 由于对每个顶点以其标记的某种顺序(譬如说上升顺序)连结的那些顶点列表,上述图的邻表由 n 个表组成,表的总长度为 $2e$,这里 e 是图中的边数. 另一组 n 个表存储与边有关的权,因此邻近表所需的存储量等于 $4e$,而邻近矩阵所需的存储量为 $2n^2$. 在稀疏连结图上,边的数目 e 小于 $O(n^2)$. 因此邻近表要求的存储量一般小于邻近矩阵所要求的存储量. 通常一对顶点间的连结可从邻近矩阵在 $O(1)$时间内确定,但是确定顶点度数需要时间 $O(n)$. 在最坏情况下,根据邻近表连结仅在 $O(n)$时间内就可确定,而顶点的度数可由邻近表在时间 $O(1)$内计算,这是由于大多数表的实现能够很有效地计算表长.

本文利用邻近表表示处理器数组上的图,每个顶点邻点表存入 PE(processing elements). 为了存取特定的邻点,此表必须被独立地存取在每个 PE 内. 例如,以最小的费用选取邻点,权重表需要用独立的分类运算进行分类. 为检查已知顶点是否是邻点,需要对邻点的表进行独立搜索. 许多这些运算能用寻址自主结构成功地实现,而且本文的求解 SSSP 问题的复制数据算法就是基于寻址自主结构实现图的运算.

2　解 SSSP 问题算法

如果把图顶点 i 的邻表存放于数组 $Adj[i]$,令 D 为图的任意顶点的最大度数,P 是边数,s 为源的顶点,那么本节给出求解 SSSP 问题的几种算法: ① 解 SSSP 问题的串行算法; ② 解 SSSP 问题的并行算法; ③ SSSP 问题的复制数据算法以及加速的计算.

2.1　算法

① 解 SSSP 问题的串行算法

```
Program Sequential−SSSP
d[i] = ω(s, i)
Repeat
  For i=1 to n do
    For j=1 to #neighbors of i do
      d[i]=min{d[i],d[Adj[i,j]]+ω(i,Adj[i,j])}
  Until the d−values are stable
End Sequential−SSSP
```

此算法的 2 个内循环在最坏情况下的时间复杂性为 $O(nD)$,外循环将迭代 P 次. 由于 P 能够大到 $n-1$,因此串行 SSSP 算法在最坏情况下时间复杂性是 $nPD=O(n^2D)$. 由于 P 在每次迭代顺序地通过每个邻域进行循环,所以此算法的并行型能够并行地处理每个顶点. 这里我们假定顶点 i 的邻近表被存储在数组 Adj_i 内第 i 个 PE 内,距离数组 d 被分配给 n 个 PEs,而 d_i 存储在第 i 个 PE 内.

② 解 SSSP 的并行算法

```
Program Parallel−SSSP
if processor−number is s
  d=0
else
  d = ∞
```

```
Repeat
  For i=1 to n in parallel do
    For j=1 to D do
      temp = receive d from PE⟨Adj_i[j]⟩
      d = min{d, temp + ω_i[j]}
    Until the d−values are stable
End parallel−SSSP
```

内循环由总的通讯、加法及最小测量值组成，此循环执行 D 次，因此其运行时间为 $O(D(t_c^g+Ct_i))$，式中 t_c^g 为总通讯时间，t_i 为取指令时间，而 C 是常数. 在超立方体和其它对数网络上执行时，总通讯时间 t_c^g 为 $O(\log^2 n)$，而在 $2D$ 和 $3D$ 网格上总通讯时间为 $O(n)$. 由于这个内循环将迭代 P 次，所以此并行单源最短路径算法在最坏情况下时间复杂性等于 $PD(t_c^g+Ct_i)=O(nD(t_c^g+Ct_i))$. 在超立方体上，此等式简化为 $O(n\cdot D\cdot \log^2 n)$.

解 SSSP 问题的复制数据算法使用图的 D 个 copies 以并行方式滚动循环，图的每个 copy 处理每个顶点的 D 个可能邻点之一，因此 1 次迭代后，对于每个顶点，计算了过每个邻点的新路径，每个顶点最小 D 值给出到达这个顶点的最新路径.

③ 解 SSSP 的复制数据算法

```
Program Replicated−SSSP
if processor−number is s
  d=0
else
  d=∞
Repeat
  For i=1 to n in parallel do
    For j=1 to D in parallel do
      temp=d value from PE⟨j, Adj_i[j]⟩
      temp=temp + ω_i[j]
    temp=spread(min_reduce(temp))
    d=min(d,temp)
  Until the d−values are stable
End Replicated−SSSP
```

以上算法使用 nD 个处理器. 内循环执行步骤要求寻址自主以便在所有 PEs 上同时执行. 此外，为迅速执行，从此相邻顶点取出距离数组 d 这一步要求通讯自主. 因此，复制数据 SSSP 算法需要通讯自主与寻址自主.

在复制数据算法中，内循环运行时间为 $O(t_c^g+t_{ScanMin}^D+Ct_i)$，式中 $t_{ScanMin}^D$ 是计算跨每个顶点的 D 个 copies 最小的扫描操作时间. 扫描操作在超立方体和其它对数网络上耗时为 $O(\log n)$，而在网格上耗时是 $O(n)$. 由于内循环将迭代 P 次，复制数据单源最短路径在最坏况下的时间复杂性是：$P(t_c^g+t_{ScanMin}^D+Ct_i)=O(P(t_c^g+t_{ScanMin}^D+Ct_i))$. 而在超立方体上，这可简化成 $O(n(\log^2 n+\log D))$.

2.2 导出加速的计算

复制数据技术的加速定义为数据并行算法与复制数据算法运行时间比. 根据分析可得出，加速由下式确定：

$$S=\frac{PD(t_c^g+Ct_i)}{P(t_c^g+t_{ScanMin}^D+Ct_i)}=\frac{D(C_1+C_2\log^2 n)}{(C_2\log^2 n+C_3+C_4\log D)} \quad (1)$$

加速主要依赖于图的最大度数. 由于每步使用了一般通讯运算，故也依赖于图的大小. 对特

定尺寸的图，n 为常数，这样根据数据复制技术获得的加速等式可简化为

$$S=\frac{K_1D}{(K_2+\log D)} \tag{2}$$

从直观上看，在算法使用的 copy 数目内，加速为线性. 然而，由于使加速亚线性到 $\log D$ 倍，故为计算最小路径，每次迭代都需付间接费用.

3　算法的实现

计算 SSSP 问题的数据并行算法和复制数据算法在联结机 CM－2 和 MP－1 上实现，本节给出通过实验获得的某些结果.

3.1　用联结机 CM－2 实现 SSSP 算法

对于复制数据算法，CM－2 形如二维数组，每行含图的一个 copy，顶点 i 被映射到每行的第 i 列内的 PE. 因二维数组的维数是 2 的幂，故每行及每列映射到互不相交的亚超立方体(Subhyper－cube)，这样就能有效地实现在图的每个 copy 内和在每个顶点的 copies 之间的运算. 第 i 列内的每个 PE 存储第 i 顶点的邻近表及权重表. 对于数据并行算法，CM－2 形如处理器的单个线性数组，顶点号被映射到第 i 个 PE 并含邻近表和权重表的第 i 行. 表 1 给出用 16K 处理器 CM－2 实现 2 种 SSSP 算法的结果，含有 128 个顶点的随机图和随机权用于完成这一实验. 任意顶点的最大度数和图中边的总数如表中所示，数据并行算法和复制数据算法以毫秒计时，复制数据算法对数据并行算法的加速在表的最后一列给出. 由实验得出，复制数据技术能产生惊人的加速. 图 1 给出根据等式(2)对各种 D 值在 $K_1=K_2=2.5$ 的情况下实验及理论的加速，图 2 给出加速与 copies 数目之间的关系.

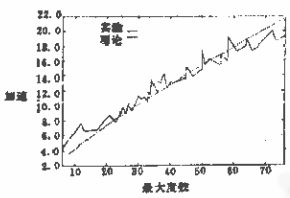

图 1 SSSP 算法在 CM－2 上运行的实验与理论加速比较

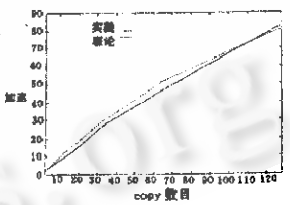

图 2 理论计算及实验获得的加速与 copies 数目间的关系曲线

3.2　用 MP－1 实现 SSSP 算法

实验用的 MP－1 由 16K 个 PEs 组成，连接成 128×128 网格. 对于复制数据算法，使用二维数组结构. 类似用 CM－2 解 SSSP 问题那样，图的每个 *copy* 映射成数组的行. 而对于数据并行算法，MP－1 可看成是 PEs 单个线性数组，顶点 i 映射到第 i 个 PE 上. 表 2 给出用 16K 的处理器的 MP－1 实现 2 种 SSSP 算法的结果，此实验用含 128 个顶点的随机图和随机权实现，并且该表给出任意顶点的最大度数(Maximum Degree)和边的总数. 数据并行算法和复制数据算法所耗时间用毫秒(*ms*)计算，而复制数据算法对数据并行算法的加速由表的最后一列给出.

表 1　用内存为 16K 的处理器的联结机 CM－2 解 SSSP 问题的计时(*ms*)

(128－顶点图，16K 个 PEs)

最大度数	边界数目	数据并行算法	复制数据算法	加速比
30	3338	511.28	45.23	11.30
34	2896	518.19	45.12	11.48
38	3658	618.22	43.53	14.20
45	3994	673.78	44.36	15.19
50	4656	728.17	49.22	14.79
52	4758	794.00	50.23	15.81
56	5098	803.95	52.41	15.34
58	5936	925.52	48.27	19.17
64	6314	1012.52	53.58	18.90
72	7468	1171.66	58.54	20.01
76	8176	1178.08	61.05	19.30

表 2　用内存为 8K 的处理器 MP－1 解 SSSP 问题的计时(ms)

(128－顶点图，16K 个 PEs)

最大度数	边界数目	数据并行算法	复制数据算法	加速比
30	3338	38.96	10.41	3.74
34	2896	37.96	10.52	3.61
38	3658	45.77	10.65	4.30
45	3994	52.86	10.99	4.81
50	4656	60.81	12.64	4.81
52	4758	62.93	13.44	4.68
56	5098	64.78	13.14	4.93
58	5936	78.79	13.49	5.04
64	6314	88.84	15.82	5.62
72	7468	113.93	17.21	6.62
76	8176	118.71	18.56	6.40

4 结 论

利用复制数据技术能够求解典型的 SSSP 图论运算问题，而且根据实验证明，与并行算法比较，获得的加速十分明显. 这不仅大大鼓舞人们直接利用复制数据技术解各种图论运算问题，而且可解许多其它问题，因此复制数据理论与技术的进一步研究十分重要，其应用前景也是无法估量的. 但是利用复制数据技术获得的加速能力依赖于互联网络怎样有效地支持研究问题所遇到的通讯种类，因此分析用 3 种互联网络实现各种数据复制算法将是我们未来研究的主要方向之一，特别是我们将着重研究在图象解释中，利用通讯自主识别邻近区域，以寻址自主产生这些邻近区域的子区域并进行有效的空间搜索.

本文完成于美国马里兰大学自动化研究中心与计算机科学系.

参考文献

1　Dijkstra E W. A note on two problems in connexion with graphs. Numerische Mithematik, 1959,1:269～271.

2　Paige R C, Kruskal C P. Parallel algorithms for shortest path problems. In: Proc. of the International Conference on Parallel Processing, 1985. 14～19.

3　Floyd R W. Algorithm 97: shortest path. Communications of the ACM, 1962,5:345～348.

4　Cypher R, Sanz J L C. SIMD architectures and algorithms for image processing and computer vision. IEEE Trans.

on Acoustics, Speech and Signal Processing, 1989,37:2158～2173.

5 Chan T F, Saad Y. Multigrid algorithms on the hypercube multiprocessor. IEEE Trans. on Computers, 1986,35: 969～977.

6 Preparata F P. Advances in computing research. Computational Geometry, JAI PRESS INC., 1983.

A STUDY ON THE REPLICATED DATA ALGORITHM FOR SOLVING THE SINGLE SOURCE SHORTEST PATH PROBLEM

Yang Jing'an

(*Institute of Artificial Intelligence Hefei University of Technology Hefei* 230009)

Abstract This paper first proposes a data parallel algorithm and a replicated data algorithm for the graph theoretic operation of computing the single source shortest paths and demonstrates the generality of the data replicated technique by applying it to the graph theoretic operation for solving SSSP problems, then computes and discusses the speedup of the replicated data algorithm over the data parallel algorithm, finally points out that the data replication technique is not only applicable to fast image analysis, but also is general enouth to be applicable to a large class of the graph theoretic operation problems.

Key words SSSP problems, the replicated data algorithm, data parallel algorithm, breadth—first—search—algorithm, speedup ratio.