一种基于综合历史信息的SMT结构分支预测算法*)

王 晶 樊晓桠 叶 曾
(西北工业大学计算机学院 西安 710072)

摘 要 在SMT结构中,可以同时从多个线程中取指。当可取指线程个数较少时,分支预测的重要性与在超标量处理器中的相比有增无减,因为SMT结构中转移误预测的代价更大了。影响分支预测准确率的关键因素是历史信息的组织方式和更新方式。本文仿真分析了这些因素对分支预测准确率的影响,提出了一种基于综合历史信息的分支预测算法——IHBP,把全局信息和局部信息结合在一起预测转移,解决了SMT结构中分支预测信息过时、混乱等问题,使得预测的准确率更具备鲁棒性。仿真结果表明:在8线程结构中,该算法与目前国际普遍采用的Gshare算法和Pag算法相比,分支预测准确率分别提高了8.5%和2.3%。

关键词 线程级并行,同时多线程,分支预测

An Intergrated History Information Branch Prediction Policy for Simultaneous Multi-threading Architecture

WANG Jing FAN Xiao-Ya YE Zeng
(School of Computer, Northwestern Polytechnical University, Xi'an 710072)

Abstract By converting thread-level parallelism to instruction level parallelism, Simultaneous Multithreaded (SMT) processors are emerging as effective ways to utilize the resources of modern superscalar architectures. However, branch prediction is still very important because of high wrong branch prediction penalty. The organization and modification method of branch history information are believed main factors affect the accuracy of branch predciton. In this paper, the various factors are simulated on 1,2,4 and 8 threads condition, where the branch prediction accuracy is compared and analyzed. And then, a new branch prediction method, called IHBP (Integrated History Information Branch Prediction), is proposed. It integrates both global history information and local history information to index Pattern History Table (PHT). The simulation results show that this new branch prediction method can improve branch prediction accuracy by 8.5% and 2.3% over per-thread Gshare and Pag method respectively for 8 threads. This speedup is enhanced by an advantage of overcoming the adverse influence of outdated and scrambled branch history and to make the prediction more stable and more accurate.

Keywords Tread level parallel, Simultaneous multi-threading, Branch prediction

1 引言

同时多线程结构[1]允许在每个周期将来自多个相互独立线程的多条指令发射到功能单元,在一定程度上缓解了分支预测的压力。但是,当可取指线程个数较少时,分支预测的重要性与在超标量处理器中的相比有增无减,因为SMT结构中转移误预测的代价更大了。

由于多个活动线程可以并行地乱序取指和执行,它们可以来自多个独立的程序段,也可以是一个并行程序的不同部分。当多个线程并行执行时,这些线程中的指令可能是乱序读取的。由于分支预测是在流水线的取指阶段进行的,因此分支预测的执行也可能是乱序的。也就是说,某个转移指令的预测可能提前执行,预测时使用的历史信息不是按照程序顺序记录的,而与具体的体系结构相关。多线程的乱序取指可能导致分支预测也是乱序的,预测时使用的历史信息相对单线程结构可能是:不充分的(包含的更新比较少)、不连续的(可能丢失一些更新)、过时的(不包含最近的更新)、混乱的(更新顺序是不正确的)、不正确的历史(包含错误的更新)[2]。若分支预测遇到上面的几种历史信息,预测准确率就会受到影响。因为这些历史引入了一些不确定因素,例如遇到指令Cache缺失的线程的转移指令比命中的取指要晚很多。

本文仿真分析了分支预测的历史信息的组织方式及更新方式对分支预测准确率的影响,提出了一种新的基于综合历史信息的分支预测算法(Integrated History Branch Prediction,IHBP),把全局信息和局部信息结合在一起预测转移,解决了SMT处理器中分支预测信息过时、混乱等问题,使得预测的准确率更具鲁棒性。

2 相关研究

在最初的同时多线程处理器结构中,大都沿用超标量处理器的分支预测技术[3]。近几年来,有研究者专门针对SMT处理器的特点,研究改进分支预测算法,使其预测准确率与超标量处理器中的准确率相当。

在超标量处理器结构的分支预测算法方面,Smith[4]提出的预测算法利用了2位饱和计数器表来跟踪转移可能发生的方向。每个分支用地址映象到一个计数器。如果对应的计数

*)本文得到国家自然科学基金(60573143)和新世纪优秀人才支持计划资助。王 晶 博士,研究方向:计算机系统结构;樊晓桠 教授,博士生导师,研究方向:计算机系统结构。

器的最高位被置位，就预测发生，否则预测不发生。最后基于分支转移的结果对这些计数器进行更新。当一个分支转移发生时，对应的2位计数器加1，否则计数器值减1。文[5]提出了两级分支预测技术，两级都用全局型表。该预测器使用两级历史信息进行预测：第一级历史记录之前 n 个转移指令的结果；针对第一级历史信息的每个模式，第二级历史信息对其最可能的分支方向进行跟踪。两级分支预测使用转移历史表(BHT -Branch History Table)，记录全局或者局部最近的 n 个分支转移的结果；使用1个或更多的2位饱和计数器组，称为模式历史表(PHT-Pattern History Table)，来跟踪分支最可能转移的方向。根据索引方式的不同，两级预测器分为9类，Gag方案最易于实现。Gselect[6]方案通过拼接地址的低位和历史寄存器来索引模式历史表。Gshare[7]使用一个全局的转移历史寄存器(BHR -Branch History Register)，将BHR的值和转移指令的地址进行异或操作，来索引第二级的模式历史表。Sechrest等人[8]提出了静态确定PHT值的PSg方法，对短的分支历史，静态确定的表和一个自适应的PHT性能相当。因为PHT的内容被静态确定，PSg方法在没有受益于自适应带来的好处的同时，也不需要PHT的预热时间，且实现起来简单。混合分支预测技术可以充分利用每种预测方法的长处，如文[7]选择通过2位计数器选择两种分支预测技术的一种、文[9]通过分支分类来构造混合分支预测技术，文[10]提出了AVG预测技术能够精确地预测循环分支。另外，文[11,12]等在上述分支预测算法上进行了一些改进。

在SMT结构的分支预测方面，Evers 等[13]提出了由多个预测器构成的混合分支预测技术，在该预测方案中包含了2bC、Gshare、Pshare、Gas、AVG以及Always Taken等多个预测器，分别用于不同的时机。Jayanth[2]在Pag预测器的基础上，用归纳和相关两种技术来提高单程序多线程处理器中分支预测的准确率。仿真结果显示，该Hybrid预测器获得的预测准确率和单线程处理器相当。

3 分支预测准确率的影响因素

影响分支预测准确率的关键因素是历史信息的组织方式和更新方式。为了分析这些因素对SMT分支预测的影响，我们通过仿真分析了不同的分支预测算法，并对影响分支预测准确率的几个因素进行分析比较。本文采用的仿真器工具是Linux版本的SMTSIM[1]仿真器。该仿真器能够用来对SMT处理器系统结构以及程序性能分析等进行建模，执行Alpha目标码，采用ICOUNT线程取指策略，仿真参数设定同文[1]。仿真程序来自SPEC2000 Benchmarks。每个测试程序都跳过最初的 3×10^8 条指令，用以“预热”仿真器，然后统计 $T\times3\times10^8$ 条指令的执行情况，其中 T 为仿真的线程个数。

3.1 共享全局信息和每线程全局信息

仿真的对象是常用的GSHARE预测器和每线程独立全局历史信息的GSHARE_PT预测器。这两种预测器的唯一区别是：GSHARE预测器的一级历史信息是所有线程共享；GSHARE_PT预测器则是按照线程独立记录的。

单线程时基于全局信息的GSHARE预测器的预测准确率如图1所示，单线程时预测的准确率avg为85%。不同线程个数时，两种预测器的平均准确率如图2所示。由于多线程执行可以有多种组合，所以对每种线程个数，都采用多种线程组合进行仿真，最后取平均值进行比较。

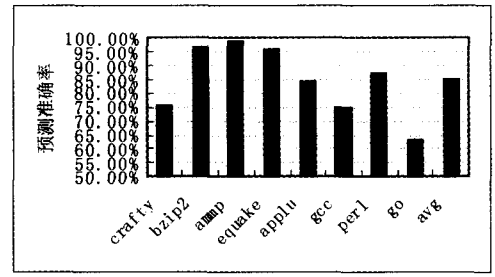


图1 单线程时GSHARE预测器的预测准确率

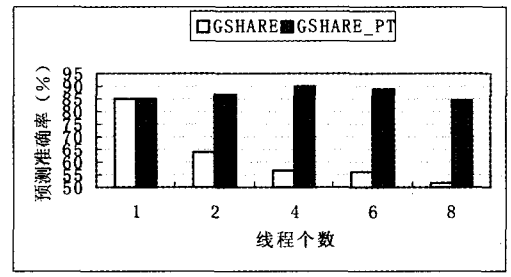


图2 不同线程个数时的预测准确率

随着线程个数的增加，基于共享信息的预测器GSHARE的预测准确率逐渐降低。到8线程时，准确率只有51.8%。而基于每线程私有信息的GSHARE_PT预测器，准确率均高于单线程时的准确率。在4线程时准确率可达到90%，比单线程的准确率高了5.2%。该结果说明：不同线程的转移信息导致共享的信息中有太多的交叉混乱，而独立的线程历史信息能更有效地反映转移的行为。

3.2 推测更新和精确更新

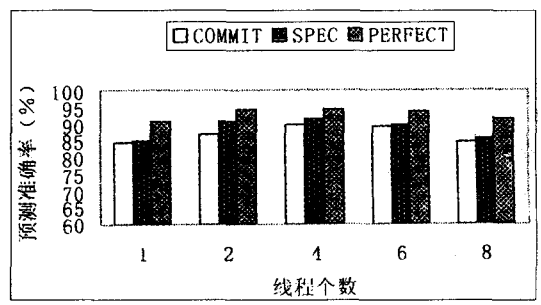


图3 信息更新方式对预测准确率的影响

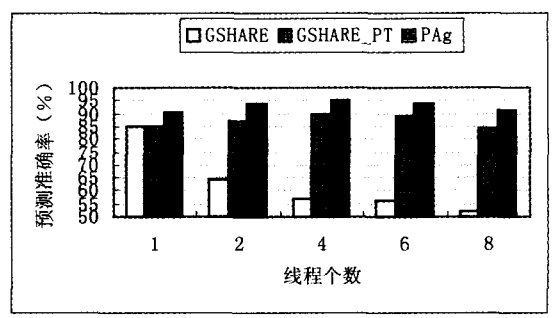


图4 基于全局/局部历史信息的预测准确率

仿真的对象是GSHARE_PT预测器，采用三种不同的信息更新方式：精确更新(COMMIT，提交时更新)、推测更新(SPEC，转移历史信息在分支预测时刻更新)和PERFECT更新(信息的更新既实时又准确，是一种理想的更新方式，这里

用于与其它两种更新方式做比较)。三种方式对预测准确率的影响如图 3 所示。从图中可以看出:(1)无论线程个数为多少,推测更新所能获得的准确率均高于精确更新,平均提高 2.1%;(2)当线程个数为 4 时,推测更新的效果最明显,预测的准确率与 PERFECT 更新几乎相当,分支预测的准确率较高,推测更新的信息大部分都正确。

3.3 全局历史信息和每地址历史信息

仿真的对象是 GSHARE、GSHARE_PT 和 PAg 三种预测器,它们均采用精确更新方式。在不同线程个数时,基于全局/局部历史信息的两种预测器的准确率如图 4 所示。从图中可以看出:(1)基于共享 BHR 的 GSHARE 预测器的准确率在线程个数较多时,受到的影响最大,由单线程时的84.7%降低到 8 线程时的 51.8%。而基于每线程独立 BHR 的 GSHARE_PT 预测器的准确率比 GSHARE 要稳定得多。(2)基于局部历史信息 BHT 的 PAg 预测器在任何情况下,均比 GSHARE 或者 GSHARE_PT 预测器要高,分别提高 48.3%和 6.9%。这也说明基于全局历史信息的分支预测更容易受到多线程取指的影响,而基于每地址的信息受到的影响要小。

4 IHBP 分支预测算法

在 SMT 结构中,由于多个线程的乱序交叉,使得历史信息可能更混乱。IHBP 分支预测算法采用了新的记录和使用历史信息的方法,其结构如图 5 所示。为了能够更好地适应不同转移指令的特点,IHBP 预测算法把全局历史信息和局部历史信息结合起来进行预测。

在 IHBP 分支预测算法中,转移指令首先根据所在的线程号把对应的全局转移历史信息选择出来,然后把选中的转移历史信息和按照地址读出的每地址的转移历史信息拼接起来作为 PHT 的索引,查找 PHT 中的饱和计数器的值,给出转移的预测结果。

尽管 IHBP 结合了两种历史信息进行预测,它与混合预测算法是不同的。混合预测方案是用两种历史信息分别进行预测,然后选择一个预测器的结果作为最后的结果,IHBP 则同时使用两种历史信息进行预测。与包含 Gag 和 Pag 的 Hybird 预测器相比,IHBP 预测算法只需要一个 PHT,而且不需要专门的选择器来选择预测结果,大大节省了面积和功耗。同时,这种拼接方式也减少了 PHT 冲突,使得某个转移指令对应的历史模式更唯一,进而提高预测的准确率。

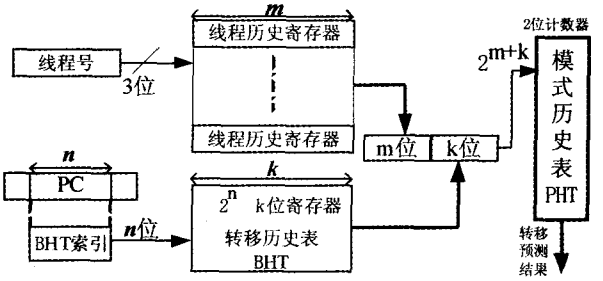
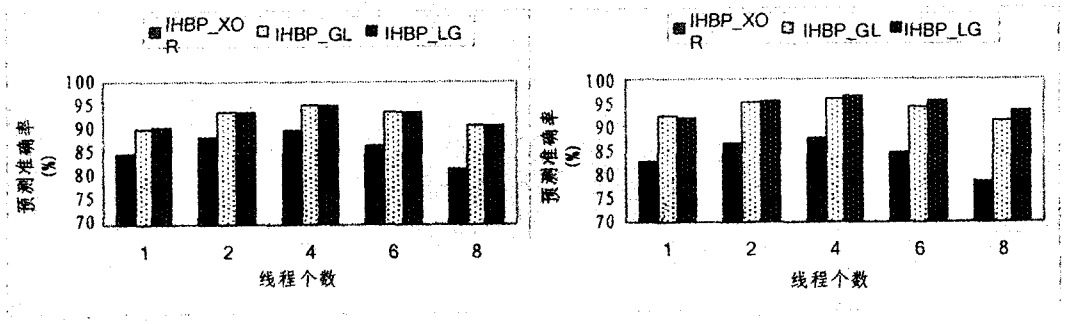


图 5 IHBP 分支预测算法结构

IHBP 分支预测方案使用全局(每线程的 BHR)和局部(每地址的 BHT)两种历史信息进行预测,两种历史信息通过一种哈希函数结合在一起,作为二级模式历史表 PHT 的索引。我们仿真了异或、全局在前局部在后拼接(每线程的 BHR 与每地址的 BHT 拼接)、局部在前全局在后拼接(每地址的 BHT 与每线程的 BHR 拼接)等三种方式,分别称为 IHBP_XOR 预测器、IHBP_GL 预测器和 IHBP_LG 预测器。并且用 PName. m. n 表示不同拼接位数的组合,其中 Pname 表示预测器的名称,有 IHBP_GL 和 IHBP_LG 两种;m 表示第一类历史信息的位数;n 表示第二类历史信息的位数。仿真结果表明 IHBP_GL. 6. 6 和 IHBP_LG. 6. 6 相对于其它拼接位数的组合准确率更高,因此后面的仿真均采用这种位数组合方式。

图 6(a)和(b)分别是历史信息采用精确更新和推测更新时 IHBP 三种索引方式对应的预测准确率。可以看出,无论是哪种更新方式,拼接方式比异或要好。不同的拼接方法对应的准确率差不多,全局在前局部在后拼接比局部在前全局在后拼接略好一些,我们在后面的仿真中均采用该方式(IHBP_GL. 6. 6)。



(a)精确更新　　(b)推测更新

图 6 三种哈希函数对预测准确率的影响

5 性能评价

图 7 和图 8 比较了 IHBP 预测器与其它几种分支预测器的预测准确率,并统计了不同预测器所能获得的 IPC。从图 8 可以看出,IHBP 预测算法在任何线程个数时,都比其它几种预测器的准确率要高。与 GSHARE 相比,IHBP 的预测准确率平均提高了 50.8%;与 GSHARE_PT 相比,预测准确率平均提高 6.6%;与 Pag 相比,预测准确率平均提高了 1.7%。IHBP 预测器的平均准确率可达到 94.5%。另外,当线程个数增加时,IHBP 预测算法的准确率也在增加,说明 IHBP 使用历史信息的方式是适合多线程处理器中的指令流特点的。IHBP 的 IPC 相对于 GSHARE、GSHARE_PT 和 Pag 分别平均提高了 34%、11.6%和 7.3%。

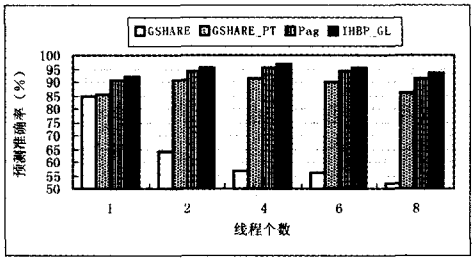


图7 IHBP与其它预测器准确率的比较

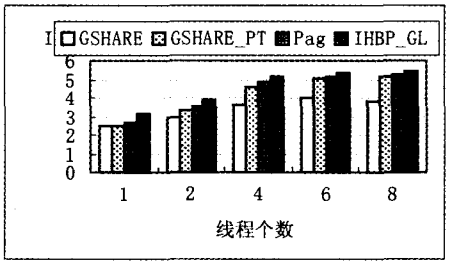


图8 几种预测器对IPC的影响

结论 本文详细分析了转移历史信息的组织方式和更新方式对分支预测准确率的影响。在此基础上，提出了IHBP分支预测算法，把全局信息和局部信息结合在一起来预测转移，同时发挥了两种转移历史信息的作用，采用推测更新的方式充分利用及时信息，解决了SMT处理器中分支预测信息过时、混乱等问题，使得预测的准确率更稳定。仿真结果表明：在8线程结构中，与目前国际普遍采用的每线程Gshare算法和Pag算法相比，分支预测准确率分别提高了8.5%和2.3%。

参考文献

1 Tullsen D M, Eggers S J, Levy H M. Simultaneous Multithreading: Maximizing On-Chip Parallelism. In: Proceedings of the International Symposium on Computer Architecture, 1995. 392～403
2 Gummaraju J, Franklin M. Branch Prediction in Multi-threaded Processors. IEEE, 2000
3 Evers M, Yeh T Y. Understanding branches and designing branch predictors for high-performance microprocessors. Proceedings of the IEEE, 2001, 89(11): 1610～1620
4 Smith J E. A Study of Branch Prediction Strategies. In: 8th Intl Symp on Computer Architecture, Minneapolis, MN, ACM, IEEE. May 1981. 135～148
5 Yeh T Y, Patt Y N. Alternative Implementations of Two-level Adaptive Branch Prediction. In: Proc. 19th Int'l Symp on Computer Architecture, 1992. 124～134
6 Pan S T, So K, Rahmeh J T. Improving the Accuracy of Dynamic Branch Prediction Using Branch Correlation. In: Proc. Architectural Support for Programming Languages and Operating Systems (ASPLOS-V), October 1992. 76～84
7 McFarling C. Combining Branch Predictors: [Technical Note]. TN-86. WRL June 1993
8 Sechrest S, Lee C C, Mudge T. The Role of Adaptivity in Two Level Adaptive Branch Prediction. In: 28th ACM/IEEE International Symposium on Microarchitecture_ Nov. 1995
9 Chang P Y, Hao E, Yeh T Y, et al. Branch Classication: a New Mechanism for Improving Branch Predictor Performance. In: 27th ACM/IEEE International Symposium on Microarchitecture. Nov. 1994
10 Chang P, Banerjee U. Profile-guided Multi-heuristic Branch Prediction. In: Proceedings of the International Conference on Parallel Processing, July 1995
11 Evers M, Chang P, Patt Y. Using Hybrid Branch Predictors to Improve Branch Prediction Accuracy in the Presence of Context Switches. In: International Symposium on Computer Architecture, May 1996. 3～11
12 Falc'on A, Santana O J, Medina P, et al. Studying New Ways for Improving Adaptive History Length Branch Predictors. In: Proceedings of the 4th International Symposium on High Performance Computing(ISHPC-IV), May 2002
13 陈跃跃，周兴铭．一种精确的分支预测微处理器模型．计算机研究与发展，2003，40(5)：741～745

(上接第237页)

第3步：同理，在子分布域S左中，进行双域双向水平倾角最小化圈绕寻找凸壳：

i. 双向"圈绕寻找下一新顶点"，作凸壳最新顶点寻找处理：

①"构造当前次新顶点的水平射线"处理：在子分布域$S_{左}$中分别过凸壳A向、B向当前新顶点$Q_{左下k}$、$Q_{左上k}$($1\leqslant k\leqslant n$)，分别作平行X轴负方向的同向顶点射线(简称正向射线)$Q_{左下k}L_{左下1}$和$Q_{左上k}L_{左上1}$($1\leqslant k\leqslant n$)；

②"寻找最小水平倾角点"处理：

在子分布域S左中，B向找出对当前次新顶点$Q_{左下k}$的水平射线$Q_{左下k}L_{左下k}$(为始边的)的水平倾角最小点$P_{左下j}$(即满足$\angle P_{左下j}Q_{左下j}L_{左下j}=\min\{\angle P_{左下k}Q_{左下j}L_{左下j}\mid P_{左下k}\in S\}$)(注：若有多个最小水平倾角点，则只取最后一个最小水平倾角点，即离当前次新顶点$Q_{左下j}$最远的那个最小水平倾角点$P_{左下j}$)；并且A向找出对当前次新顶点$Q_{左上k}$的水平射线$Q_{左上k}L_{左上k}$(为始边的)的水平倾角最小点$P_{左上i}$(即满足$\angle P_{左上i}Q_{左上i}L_{左上i}=\min\{\angle P_{左上k}Q_{左上i}L_{左上i}\mid P_{左上k}\in S\}$)(注：若有多个最小水平倾角点，则只取最后一个最小水平倾角点，即离当前次新顶点$Q_{左上k}$最远的那个最小水平倾角点$P_{左上i}$)；

③"标记当前最新顶点"处理：把当前所得水平倾角最小点$P_{左下i}$、$P_{左上i}$，分别按顺次保存为最新顶点$Q_{左下k}$、$Q_{左上k}$($1\leqslant k\leqslant n$)。

ii. "分布域极小化"，即当对前(已得各顶点所构成的)子凸壳$S_{左}$内点做类似子凸壳$S_{右}$中内点的处理。

最后，顺序把所得各顶点，依次两两连接而得到的凸多边形Q，必定是所求二维有限点集S的凸壳Q。

不难看出：本凸壳新算法，经极小化处理后的新分布域S中，其无效内点个数一般都会快速减少，故已可大幅提高凸壳Q生成速度。

结论 本文提出的双域双向水平倾角最小化圈绕，实现了小分布域化的凸壳顶点生成。它不仅克服了卷包裹凸壳算法等的近似或低效等弱点，改善了凸壳算法的时间、空间复杂度与效率，且颇易于改造为并行化算法。因此，它将有效提高二维凸壳生成速度，可进一步改进和提高二维凸壳在图像处理、文字分解、模式识别、物体分类、计算图形、指纹识别、遥测遥控、地物辨识、地质勘探、空天利用等的应用水平和工作效率。

参考文献

1 周启海．论二维点集或线段集凸壳生成算法改进与优化的同构化方向[J]．计算机科学，2007(7)
2 周启海．简论二维点集凸壳研究的意义、现状与创新[C]．见：第三届全国几何设计与计算学术会议论文集，北京：电子工业出版社，2007
3 周启海，杨祥茂，吴红玉．单域单向水平倾角最小化圈绕凸壳新算法[J]．西华大学学报(自科版)，2006(2)
4 周启海，吴红玉，黄涛．单域双向水平倾角最小化圈绕凸壳新算法[J]．计算机科学，2007(8)
5 周启海，黄涛，吴红玉，张元新．基于最大基线倾角智能逼近的凸壳新算法[J]．计算机科学，2007(9)
6 黄涛，周启海．双域单向水平倾角最小化圈绕凸壳新算法[J]．计算机科学，2007(12)
7 Chand D, Kapur S. An algorithm for convex polytope[J]. ACM, 1970, 17: 78～86