广西科学院学报
Journal of Guangxi Academy of Sciences

2009,25(4):270~272,277
Vol. 25,No. 4 November 2009

多核机器上线程级并行加解密数据库数据方法*

A Thread-Level Parallel Encryption and Decryption Method for Database on Multi-core Machines

韦 伟,冯 佩,柯 琦,林 瑞,钟 诚
WEI Wei,FENG Pei,KE Qi,LIN Rui,ZHONG Cheng

(广西大学计算机与电子信息学院,广西南宁 530004)
(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004,China)

摘要:利用数据划分思想和多线程技术,提出一种加解密数据库数据的方法并用实验来检验方法的有效性。该方法使用3-DES密码算法,参考资源调度策略与划分机制,在多核计算机系统上对数据库中待加解密的数据进行划分,并利用多线程并行技术对数据进行加解密处理。实验结果表明,多核多线程并行方法能够显著地加快数据库数据加解密速度。

关键词:数据库 加密解密 多核计算机 多线程 并行算法
中图法分类号:TP309.7,TP338.6 **文献标识码**:A **文章编号**:1002-7378(2009)04-0270-03

Abstract:Based on data partitioning idea and multi-threading technology, a database encryption and decryption method is proposed and its effectiveness is verified by experiments. The data stored in database on a multi-core computer are firstly partitioned by applying resource scheduling strategy and partitioning mechanism. Then the data are encrypted/decrypted by parallel multiple threads and 3-DES cipher algorithm. The experimental results show that the presented method can accelerate significantly the database encryption and decryption operations by multi-core multi-threading techniques.

Key words:database,encryption and decryption,multi-core computers,multiple-threads,parallel algorithms

随着计算机、网络和数据库技术的广泛应用,对数据库中的敏感数据进行加密保护成为一项非常重要的工作[1]。在实际应用中经常需要对数据库中的海量数据进行处理并提供实时响应,这就需要高性能并行分布计算技术[2]的支撑。与传统的单核处理器相比,多核处理器提供了更强的并行处理能力[3],已经成为微处理器设计的主流方向。本文运用多核计算机系统的多线程并行技术[4],提出一种实现数据库数据加解密方法,并进行实验测试。

收稿日期:2009-10-10
作者简介:韦 伟(1985-),男,硕士研究生,主要从事网络与并行分布计算研究。
*广西高校优秀人才资助计划项目(RC2007004)资助。

1 数据库数据并行加解密方法

数据库数据并行加解密使用3-DES密码算法[5],参考资源调度策略与划分机制[6],在多核计算机系统上对数据库中待加解密的数据进行划分,并利用多线程并行技术对数据进行加解密处理,以提高大规模数据库数据加解密的响应速度。

由于许多数据库中的数据量通常很大,所以需要分多轮将数据读入内存以进行加解密处理。为此,使用数组作为数据的缓存,每一轮只从数据库中读取与所设置的缓存大小一致的部分数据进行并行加解密处理,经过多轮处理之后才完成对所有数据库数据的加解密。

1.1 数据库数据划分

根据所生成的并行线程的数目 Threads,对数

据库中的数据表的 m 行(记录)和 n 列(字段/属性)进行动态地划分,使其满足 Threads =m'×n',其中 m' 为分组数据的行(记录)数,n' 为分组数据的列(字段)数,1≤m' ≤m,1≤n'≤n。这样,就将数据表划分成了 Threads 块。然后,通过一个二维数组元素 A[i][j]来描述动态划分后的各数据块所对应的线程号 T(tr,tf),其中 i,j 为数据表的行和列号。

采用4个线程对数据表中16个数据项进行划分见图1。

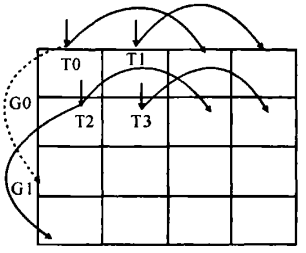

图1 2×2数据划分示例

1.2 线程级并行加密数据算法

输入:明文数组 pText,记录数 m,字段数 n,分组记录数 m',分组字段数 n',线程标识号(tr,tf)

输出:密文数组 cText

```
Begin
 for (int i= tr; i<m; i=i+m')
   { for (int j=tf; j<n; j=j+n')
     { 调用加密算法对数组单元 pText[i][j]进
       行加密,结果保存到数组单元 cText[i]
       [j]; } }
End.
```

类似地,可以给出多线程并行解密数据库数据的算法。

1.3 多线程并行的同步

考虑到异步并行的每个线程完成加解密的时间可能不同,因此必须使得当最后一个线程完成处理之后,才能读取数据表中下一数据块。这就需要实现多线程之间的同步。同步用于协调线程执行和管理共享数据[7]。本文采用路障机制[4]来同步各个异步执行的线程。在允许读取下一个待加密的数据块之前设置路障,这就保证所有线程在完成当前加密任务之后,才能越过该逻辑点继续执行。

使用 Java 语言[8]编程实现并行加解密。具体算法的调用通过 JDK 提供的 Cipher 类的对象来执行。从更优化程序执行性能的目的出发,使用数据复制的方法,创建一个类对象集合,为每个线程分配一个 Cipher 类对象变量,使得每个线程分别访问不同的类对象,从而避免因为隐式锁的存在而引起的线程串行等待。

2 性能测试

2.1 测试环境

四核计算机 Intel(R) Xeon(R) E5405 2.0GHz,其内存 2GB、2 级缓存 12MB,1 级缓存 128KB;Windows Sever 2003操作系统;SQL Sever 2000数据库;编程语言和环境为 Java 与 JDK 1.6.0_10。

2.2 测试结果及分析

在数据库中设计3张数据表:明文表、密文表和解密表,分别用于保存待加密的原始数据、加密后的结果数据以及解密后的结果数据。明文表有10个字段,字段的数据类型为 varchar,其长度为51个字符,数据项的内容是随机生成的定长字符串,每条记录数据规模约0.5KB。采用动态关闭处理器核的方法,用3-DES 密码算法对记录数目分别为100万、200万、300万、400万、500万和600万的数据进行加密。分别在线程数不同、处理器核数不同和记录数不同3种情况下测试新方法对加密时间的影响。分别运行1～4个处理器核时,多线程并行加密明文表中的数据所需的运行时间如图2～5所示。

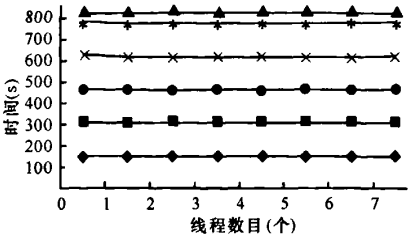

图2 运行1个处理器核时,线程数目与运行时间关系

-▲-:6.0×10⁶;-*-:5.0×10⁶;-×-:4.0×10⁶;-●-:3.0×10⁶;-■-:2.0×10⁶;-◆-:1.0×10⁵。

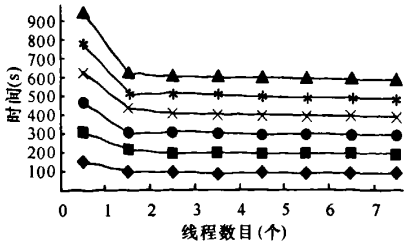

图3 运行2个处理器核时,线程数目与运行时间关系

-▲-:6.0×10⁶;-*-:5.0×10⁶;-×-:4.0×10⁶;-●-:3.0×10⁶;-■-:2.0×10⁶;-◆-:1.0×10⁵。

从图2可以看出,对于固定的记录数,增大线程数并不能显著地减少加密时间,相反还有可能会增

加耗时。这是因为多个线程分时共享一个处理器核造成竞争。并发执行的线程越多，线程启、停以及切换就越频繁，操作系统调度多个线程的开销也越大，相应抵消了多线程并行加密带来的效益。

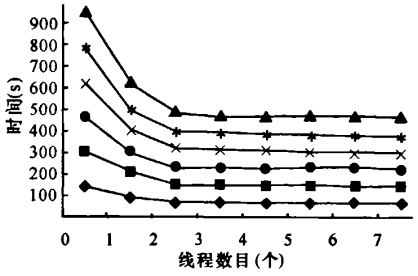

图4 运行3个处理器核时，线程数目与运行时间关系

-▲-：6.0×10⁶；—*—：5.0×10⁶；—×—：4.0×10⁶；-●-：3.0×10⁶；-■-：2.0×10⁶；-◆-：1.0×10⁶。

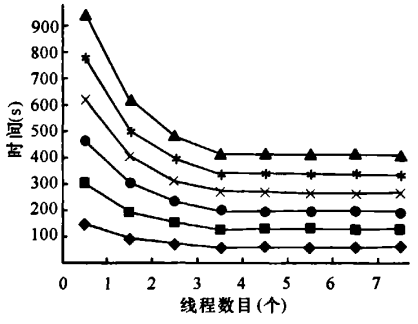

图5 运行4个处理器核时，线程数目与运行时间关系

-▲-：6.0×10⁶；—*—：5.0×10⁶；—×—：4.0×10⁶；-●-：3.0×10⁶；-■-：2.0×10⁶；-◆-：1.0×10⁶。

图3～5的结果表明，在线程数小于处理核数的范围内，增加线程数目能够明显地减少加密时间。因为每一个线程都能够被操作系统分配到一个处理器内核上运行，多线程之间对处理器核没有竞争，能够有效地并行加密，达到了较好的加速。

由图6可知，当固定记录数为4.0×10^6条，并且线程数不超过处理器核数时，对于相同的线程数，使用处理器核数越多，其加密所需的时间越少。但是，当线程数大于处理器核数时，4种情形下多线程并行加密所需的时间并没有明显的下降。

图7结果表明，当固定记录数为4.0×10^6条，并且使用4个处理器核，运行4个线程时，并行加密获得的加速最小，运行12个线程时，获得最大的加速。当使用3个处理器核，运行3个线程时，并行加密获得的加速最小，运行9个线程时，获得最大的加速。当使用2个处理器核，运行2个线程时，并行加密获得的加速最小，运行10个线程时，获得最大的加速。另一方面，对于相同的记录数、相同的线程数，当处理器核数增加时，多线程并行加密获得的加速比是逐渐增大的。

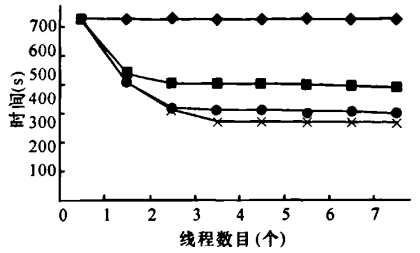

图6 不同处理器核数和线程数运行时的加密时间

-◆-：1核；-■-：2核；-●-：3核；-×-：4核。

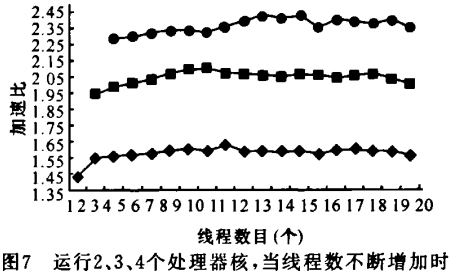

图7 运行2、3、4个处理器核，当线程数不断增加时并行加密的加速比

-◆-：2核；-■-：3核；-●-：4核。

3 结束语

本文利用数据划分思想和多线程技术，在多核计算机上实现线程级并行加解密数据库数据。实验结果表明：在多核机器上运用多线程并行方法能够显著地提高数据库加解密的速度。程序中创建的线程数受限于可用的处理器核数，对于不同处理器核数，应该设计最佳的线程数与之匹配，以使得加解密的加速比达到最大。创建过多的线程并不一定能够提高数据处理效率，因为并发执行的线程越多，线程启动、撤销以及切换就越频繁，操作系统调度多个线程的开销也会越大。下一步的工作将研究如何利用多核处理器共享二级缓存的大小，分配合适的记录数进行加解密，以减少数据在主存和二级缓存之间的交换。

参考文献：

[1] 钟诚，赵跃华. 信息安全概论[M]. 武汉：武汉理工大学出版社，2003.

[2] 陈国良. 并行算法的设计与分析(修订版)[M]. 北京：高等教育出版社，2002.

[3] 李晓明，王韬，刘东，等. 走进多核时代[J]. 计算机科学与探索，2008，2(6)：562-570.

(下转第277页)

efficient full-domain k-anonymity: proceedings of the 24th ACM SIGMOD International Conference on Management of Data[C]. New York: ACM Press, 2005:49-60.

[13] Iyengar V. Transforming data to satisfy privacy constraints: proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York: ACM Press, 2002:279-288.

[14] Wang K, Yu P, Chakraborty S. Bottom-up generalization: a data mining solution to privacy protection: proceedings of the 4th IEEE International Conference on Data Mining[C]. Washington DC: IEEE Computer Society, 2004:249-256.

[15] Fung B, Wang K, Yu P. Top-down specialization for information and privacy preservation: proceedings of the 21st International Conference on Data Engineering [C]. Washington DC: IEEE Computer Society, 2005, 205-216.

[16] Lefevre K, Dewitt D, Ramakrishnan R. Mondrian multidimensional k-anonymity: proceedings of the 22nd International Conference on Data Engineering [C]. Washington DC: IEEE Computer Society, 2006: 25-34.

[17] Ji-WonByun, Ashish Kanira, Elisa Bertino, et al. Efficient k-anonymity using clustering technique[R]. CERIAS Technical Report 2006-10, West Lafayette, Indiana: Purdue University, 2006.

[18] Jiuyong Li, Raymond Chi-Wing Wong, Adawai-Chee Fu, et al. Achieving k-anonymity by clustering in attribute hierachical structures: proceedings of the 8th International Conference on Data Warehousing and Knowledge Discovery[C]. Krakow: [s. n.], 2006:405-416.

[19] Gagan Aggarwal, Tomas Feder, Krishnaram Kenihapadi, et al. Achieving anonymity via clustering: proceedings of the 25th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems[C]. New York: ACM Press, 2006:153-162.

[20] Duncan G, Feinberg S E. Obtaining information while preserving privacy: a markov perturbation method for tabular data: proceedings of Joint Statistical Meetings [C]. Anaheim, CA, 1997.

[21] Xiao X, Tao Y. Anatomy: Simple and effective privacy preservation: proceedings of the 32nd International Conference on Very Large Databases [C]. Seoul, Korea: Sept, 2006:139-150.

[22] 杨晓春,王雅哲,王斌,等. 数据发布中面向多敏感属性的隐私保护方法[J]. 计算机学报, 2008, 31(4): 574-586.

[23] Wong R, Liu Y, Yin J, et al. (a,k)-anonymity based privacy preservation by loss join: proceedings of the Advances in Data and Web Management, Joint 9th Asia-Pacific Web Conference, and 8th International Conference on Web-Age Information Management, Huangshan, Anhui, China [C]. Lecture Notes in Computer Science 4505, Springer, 2007:733-744.

[24] 刘喻,吕大鹏,冯建华,等. 数据发布中的匿名化技术研究综述[J]. 计算机应用, 2007, 27(10): 2361-2364.

[25] Xiaokui Xiao, Yufei Tao. Personalized Privacy Preservation: proceedings of the ACM SIGMOD International Conference on Management of Data[C]. New York: ACM Press, 2006:229-240.

[26] 陈珂. 开放式环境下敏感数据安全的关键技术研究[D]. 杭州: 浙江大学, 2007.

[27] Xu J, Wangw, Pei J, et al. Utility-based anonymization using local recoding: proceedings of the 12th ACM SIGK DD international conference on Knowledge discovery and data mining [C]. New York: ACM Press, 2006:785-790.

(责任编辑:韦廷宗)

(上接第272页)

[4] Shameem Akhter, Jason Roberts. Multi-Core programming: increasing perform-ance through software multi-threading[M]. 北京: 电子工业出版社, 2007.

[5] 卢开澄. 计算机密码学[M]. 北京: 清华大学出版社, 2003.

[6] 王晶,樊晓桠,张盛兵,等. 多核多线程结构线程调度策略研究[J]. 计算机科学, 2007, 34(9): 256-258.

[7] EI-Moursy A, Garg R, Albonesi D H, et al. Compatible phase co-Scheduling on a CMP of multi-threaded processors: proceedings of IEEE 20th International Parallel and Distributed Processing Symposium, April 25-29, 2006[C]. Rhodes Island, Greece, pp, 10-22.

[8] Cay S Horstmann, Gary Cornell. JAVA 核心技术:第二卷[M]. 第7版. 北京: 机械工业出版社, 2006.

(责任编辑:尹 闯)