

面向多核架构的操作系统可扩展性研究

Research on Scalability of Operating Systems on Multi-Core Platforms

姓 名：崔 岩

导 师：史元春 教 授

陈 渝 副教授

学 号：2007310406

提纲

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排



提纲

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- 选题背景
 - 多核挑战
 - 可扩展性瓶颈
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排



多核挑战

- 选题背景

- 1. 多核挑战

- 2. 可扩展性瓶颈

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

Multi-Core Hardware

Power
Wall

ILP
Wall

Memory
Wall

- 工业界

- 普遍采用
- Intel/AMD/IBM/Sun/...

- 涉及领域

- 无处不在
- 服务器/PC/笔记本/嵌入式系统/...



多核挑战

- 选题背景

- 1. 多核挑战

- 2. 可扩展性瓶颈

- 相关研究

- 研究内容

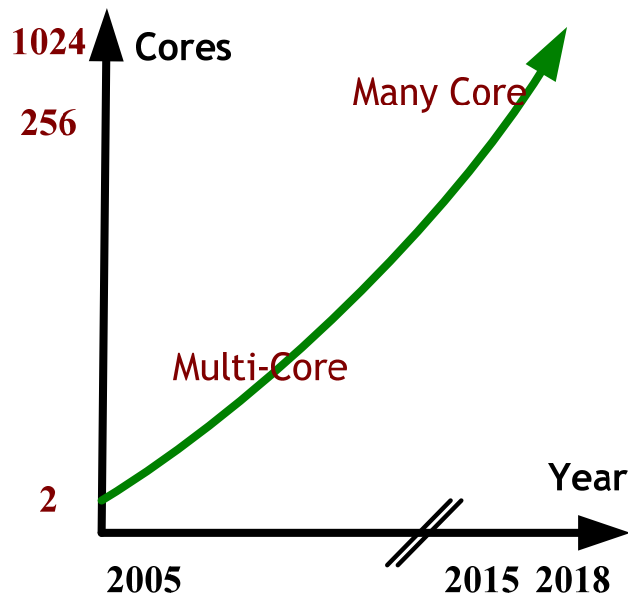
- 现有研究成果

- 研究进度安排

- 多核(CMP) V.S. 对称多处理器(SMP)

- 核的数目更多

- SMP: 低端(2CPU), 中级(4~8CPU), 高端(>16CPU)
- CMP: 4~8 cores 多核系统, **1000+ cores (<10年)**
E.g. Intel's 80 cores chip & Tiler's 64 core chip



多核挑战

- 选题背景

1. 多核挑战

2. 可扩展性瓶颈

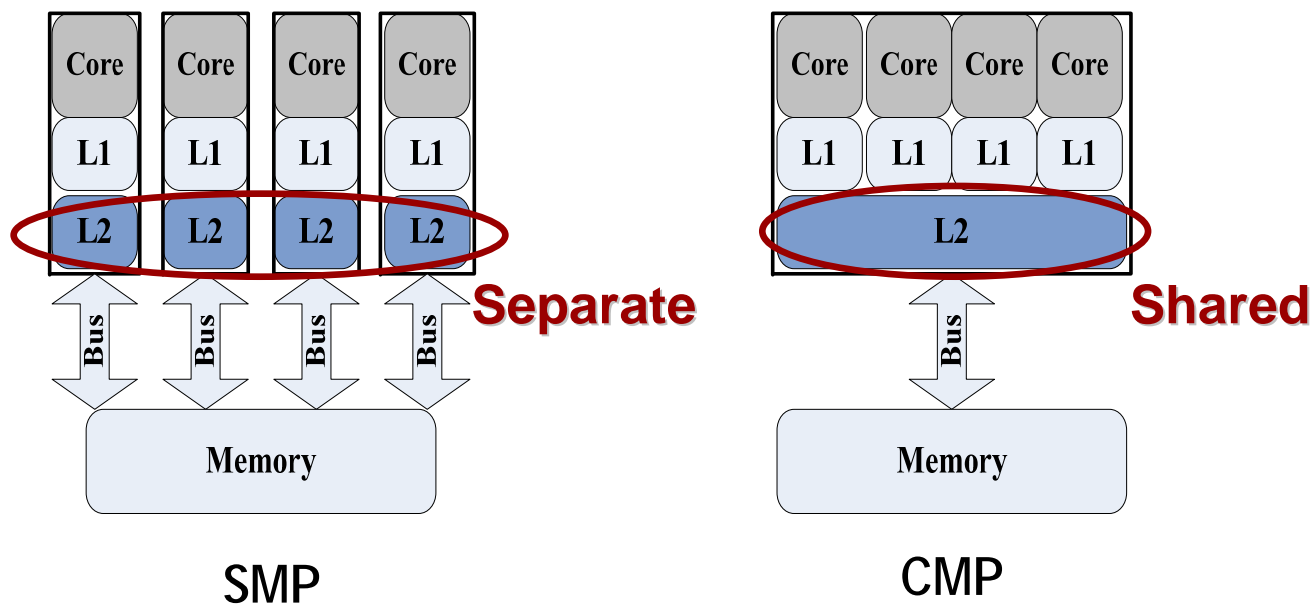
- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

- 多核(CMP) V.S. 对称多处理器(SMP)
 - 硬件资源共享(e.g., 最后级缓存)



多核挑战

- 选题背景

1. 多核挑战

2. 可扩展性瓶颈

- 相关研究

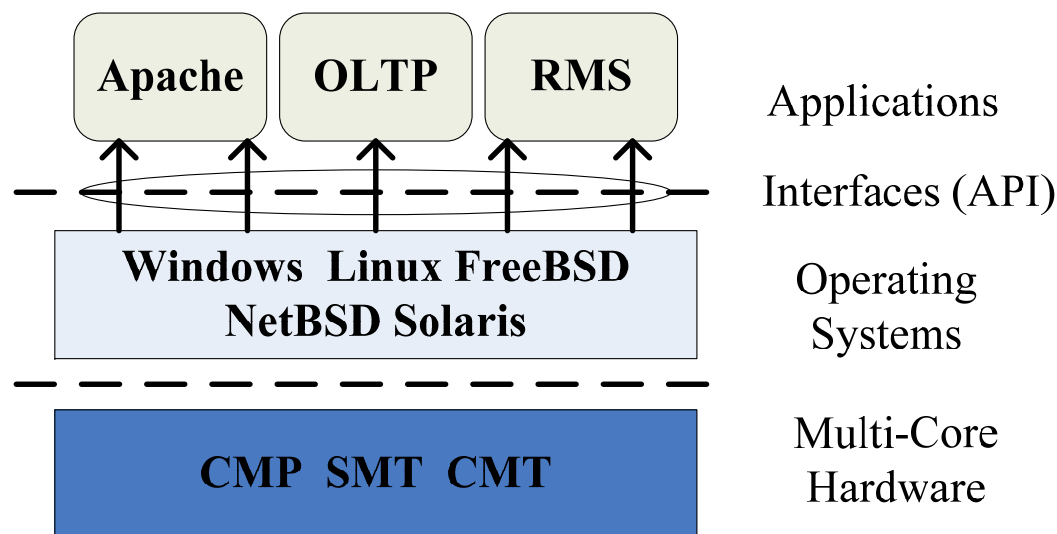
- 研究内容

- 现有研究成果

- 研究进度安排

- 操作系统的可扩展性

- 性能是否随着核的数目线性变化



可扩展性瓶颈

- 选题背景

- 1. 多核挑战

- 2. 可扩展性瓶颈

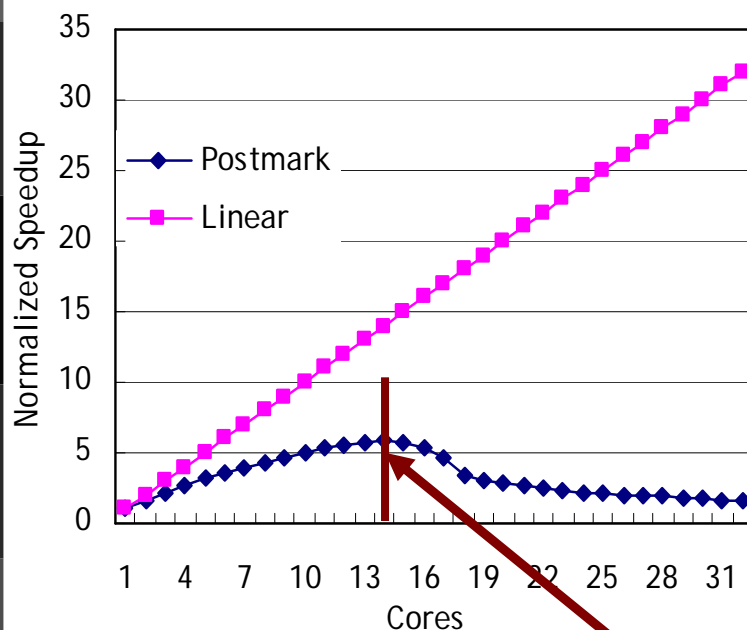
- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

内核锁(kernel lock)竞争



操作系统: Linux

实验平台: AMD 32 cores

测试程序: Parallel Postmark
(文件服务器模拟)

可扩展性瓶颈: 保护文件描述符表的自旋锁竞争^[1]

Speedup Decrease!

[1]. Yan Cui, Yu Chen, Yuanchun Shi, Qingbo Wu. Parallel Scalability Comparison of Commodity Operating Systems for Large-Scale Multi-Cores. In Proceedings of WIOSCA 2009.



可扩展性瓶颈

- 选题背景

- 1. 多核挑战

- 2. 可扩展性瓶颈

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

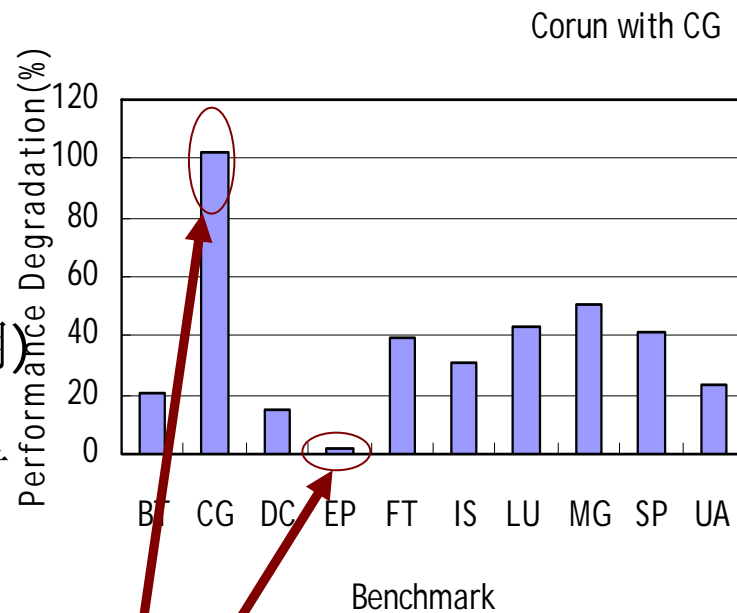
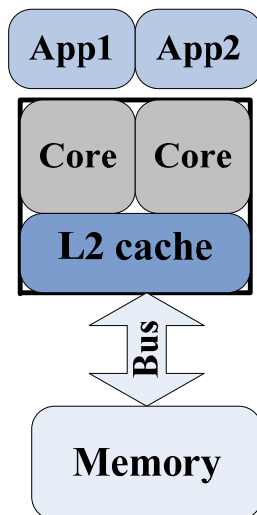
- 共享缓存竞争

操作系统: Linux

实验平台: Intel 8 cores

测试程序: NPB (科学计算应用)

可扩展性瓶颈: 共享缓存竞争



Effect of Cache Contention: 2%~102%



可扩展性瓶颈

- 选题背景

- 1. 多核挑战

- 2. 可扩展性瓶颈

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

- 共享是影响操作系统可扩展性的最根本原因

- 共享有多种体现:

内核数据的共享→锁竞争

硬件缓存的共享→缓存竞争

地址总线的共享→总线竞争^[24]

- 共享的两面性

好处: 方便、快速的通信

坏处: 引入额外数据交互, 降低可扩展性



提纲

- 选题背景

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

- 选题背景

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排



相关研究

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- 内核数据共享控制
 - 锁: 细粒度锁、自适应锁、可扩展锁 (MCS lock)
 - 复制: local run queue、slab allocator、Tornado (K42)
 - 应用控制共享: Corey
- 硬件资源共享控制(侧重在缓存竞争)
 - 资源划分: 硬件划分、软件划分
 - 调度: CASC



相关研究(1)

- 选题背景

- 相关研究

- 1. 数据共享控制

- 2. 缓存共享控制

- 研究内容

- 现有研究成果

- 研究进度安排

- 内核数据共享控制—锁
 - 细粒度锁：分解临界区
耗时，困难，易错 (死锁)^[21]
 - 自适应锁：自适应的睡眠和忙等
无法增强临界区数据的局部性(locality)^[7]
 - 可扩展锁 (e.g., MCS lock^[3]): 锁等待者等在不同的变量上
开销较大^[5]; 无法增强临界区数据的局部性^[7]



相关研究(2)

- 选题背景

- 相关研究

1. 数据共享控制

2. 缓存共享控制

- 研究内容

- 现有研究成果

- 研究进度安排

- 内核数据结构共享控制—复制

- 每个执行序列拥有一份共享数据拷贝，提升局部性

Linux local run queue^[26]、slab allocator、Tornado cluster object (OSDI 1999)^[25]

合并时有开销；共享数据冗余



相关研究(3)

- 选题背景

- 相关研究

1. 数据共享控制

2. 缓存共享控制

- 研究内容

- 现有研究成果

- 研究进度安排

- 内核数据结构共享控制—应用控制共享
 - 最小化共享数据冗余; 应用程序控制必要的数据共享 Corey (OSDI 2008)^[5]



相关研究(4)

- 选题背景

- 相关研究

1. 数据共享控制

2. 缓存共享控制

- 研究内容

- 现有研究成果

- 研究进度安排

- 硬件缓存共享控制—划分

- 硬件划分：基于扩展的LRU算法^{[9][10][23]}
需要特殊硬件的支持；缓存划分的粒度过大(way)
- 软件划分：基于页着色算法^{[11][13][22]}
性能提升依赖于程序组合；在线获取局部性信息引入较大开销



相关研究(5)

- 选题背景

- 相关研究

1. 数据共享控制

2. 缓存共享控制

- 研究内容

- 现有研究成果

- 研究进度安排

- 硬件缓存共享控制—调度

- 调度器选择合理的程序搭配(CASC)^{[13][14]}

没有硬件层的资源划分策略，性能提升有限(e.g., 流式应用和缓存敏感型应用)



提纲

- 选题背景

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

- 难点及预期创新点

- 现有研究成果

- 研究进度安排



研究目标

- 选题背景
- 相关研究
- 研究内容

1. 研究目标

2. 研究方案

3. 难点和创新点

- 现有研究成果
- 研究进度安排

● 前提

- 只考虑具有**硬件缓存一致性**(cache coherence)的多核系统

● 研究目标

- 研究**共享控制**机制, 有效提升操作系统在多核平台上的可扩展性



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 关键问题

- 共享控制关键技术
- 基于共享控制的研究型操作系统

- 指导原则

- 向操作系统的核心决策模块提供充分的信息 (应用程序, 多核硬件, 操作系统)



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存

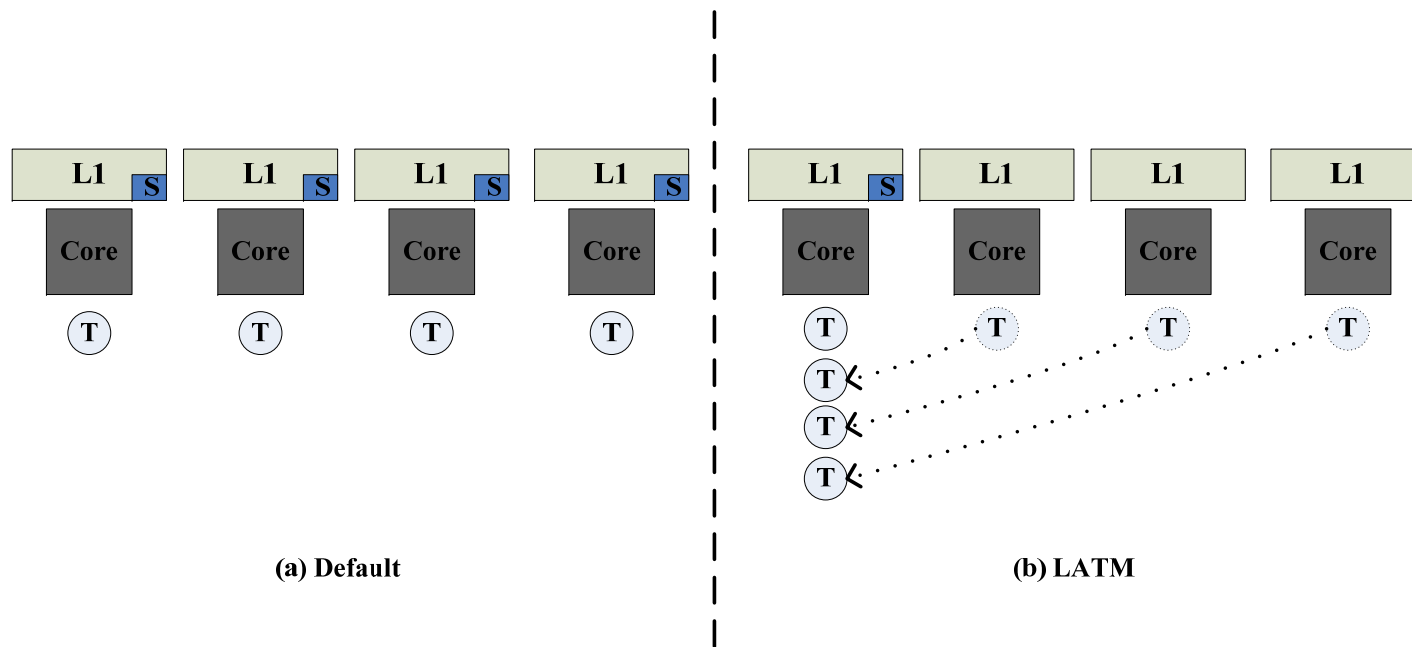
系统原型设计

- 难点和创新点

- 现有研究成果

- 研究进度安排

策略：锁感知的线程迁移 (LATM)^[27]



[27].Yan Cui, Yu Chen, Yuanchun Shi. Improving Kernel Scalability by Lock Aware Thread Migration. In Proceedings of Parallel Architectures and Compilation Technologies (PACT 2009).



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存
系统原型设计

- 难点和创新点

- 现有研究成果

- 研究进度安排

● 优点:

特点 技术	增强锁 局部性	增强临界 区局部性	减少数据冗余	降低冲突概率
细粒度锁	否	否	否	是
自适应锁	否	否	否	是
可扩展锁	是	否	否	是
数据复制	是	是	否	是
线程迁移	是	是	是	是



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 挑战:

- 线程迁移的时机

- 使用多少核处理迁移的线程

- 应用程序的行为随时间动态变化



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存

系统原型设计

- 难点和创新点

- 现有研究成果

- 研究进度安排

- 线程迁移的时机:

- 每个临界区前后

开销大: 净开销 8.1~10.5us (32 cores); 多个线程同时迁移 锁竞争; Postmark 性能降低90%

- 根据竞争激烈程度

计算 $E_j = \frac{\sum_i L_{ij}}{T_j}$ 若 $E_j \geq r$ 则该线程需要迁移

线程在第j个时间片的用锁程度

第j个时间片长度

第j个时间片第i个锁的时间

用户控制阈值



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存
系统原型设计

- 难点和创新点

- 现有研究成果

- 研究进度安排

- 使用多少核处理迁移线程：

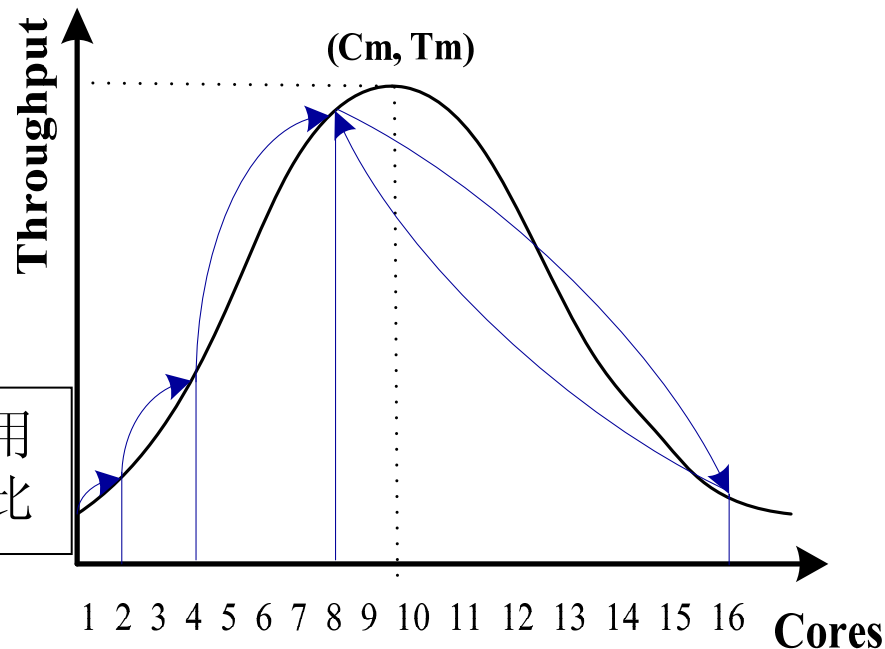
- 建模

$$\text{Max } n(1 - p(n))$$

$$\text{s.t. } n > 0, 0 < p(n) < 1$$

处理迁移
线程核数

线程的平均用
锁时间百分比



- 难点 测量误差



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 应用程序的行为随时间动态变化
 - 用锁行为变化 & 进程加入和退出
 - 固定间隔重新迁移
间隔选取：经验值(定长)；根据应用相关信息(变长)



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存

系统原型设计

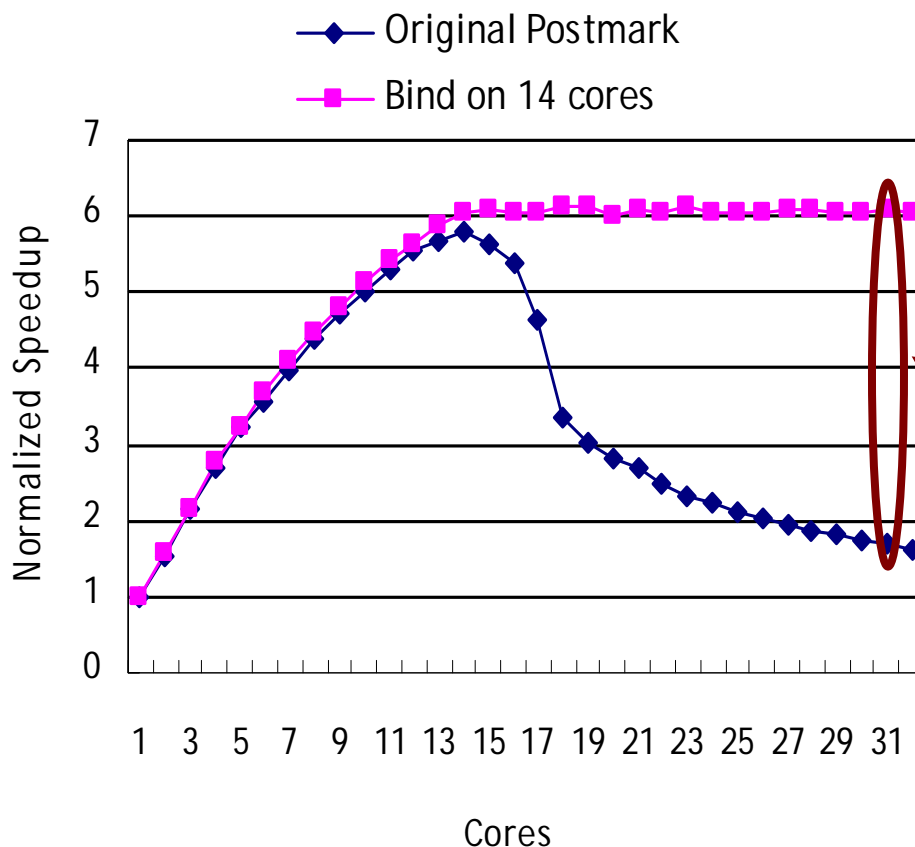
- 难点和创新点

- 现有研究成果

- 研究进度安排

● 潜力

- 设置亲和性(affinity)验证

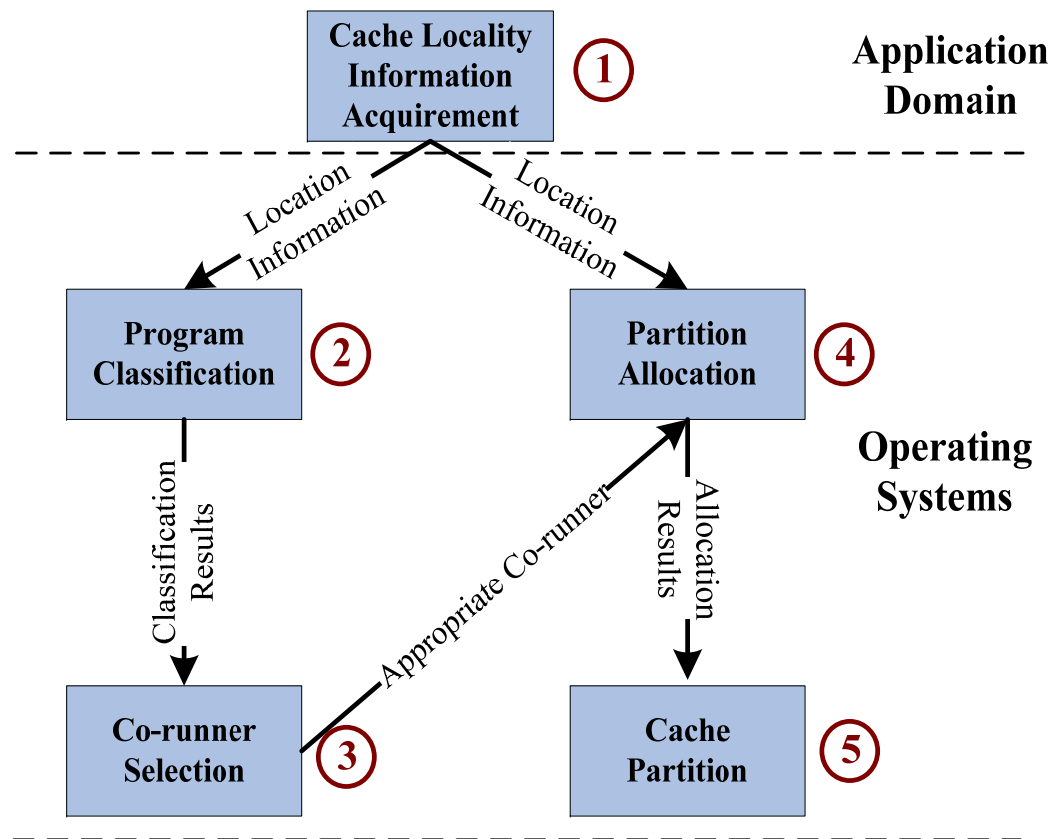


线程迁移可以达到的性能潜力



研究方案

策略：软件控制的共享缓存管理(SCSC)



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案
线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 优点:

1. 局部性信息获取自然 (应用提供 v.s. 操作检测^[13] v.s. 特殊硬件获得^[9])
2. 性能提升空间大 (调度器辅助缓存划分)
3. 不需改动现有硬件



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案
线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 挑战:

1. 局部性信息表示、获取和传递

2. 缓存划分方法

3. 调度策略

4. 局部性信息随时间变化



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存

系统原型设计

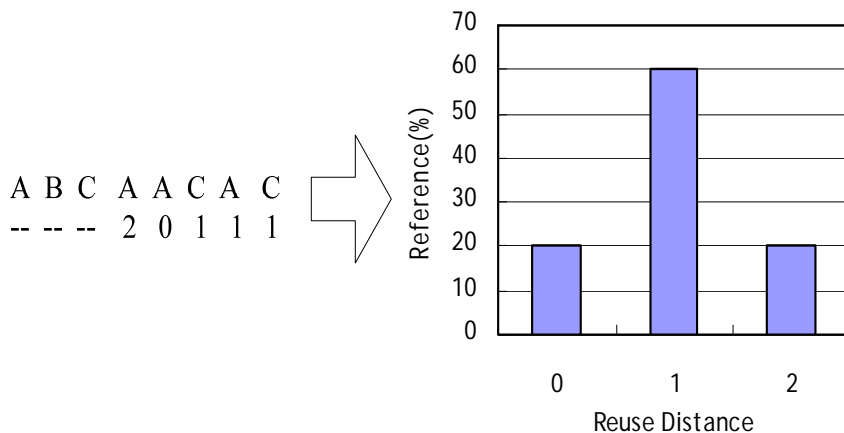
- 难点和创新点

- 现有研究成果

- 研究进度安排

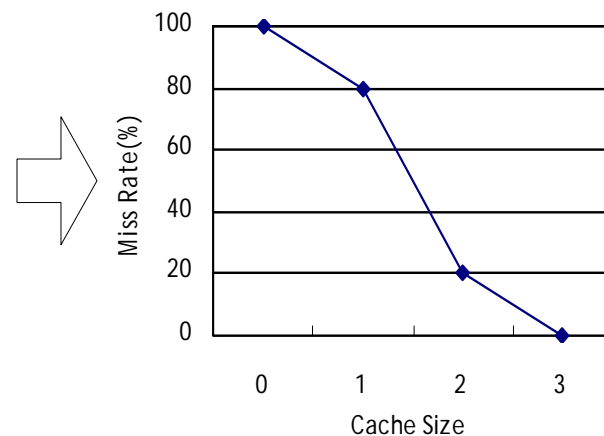
- 局部性信息表示:

- 重用距离 (reuse distance), MRC图



(a) Data Trace And Reuse Distance

(b) Histogram



(c) Miss Rate Curve



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案
线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 局部性信息获取和传递:
 - PIN注入工具 → 重用距离
 - 扩展二进制文件格式(e.g., 增加 section; 利用预留空间捎带等)



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存

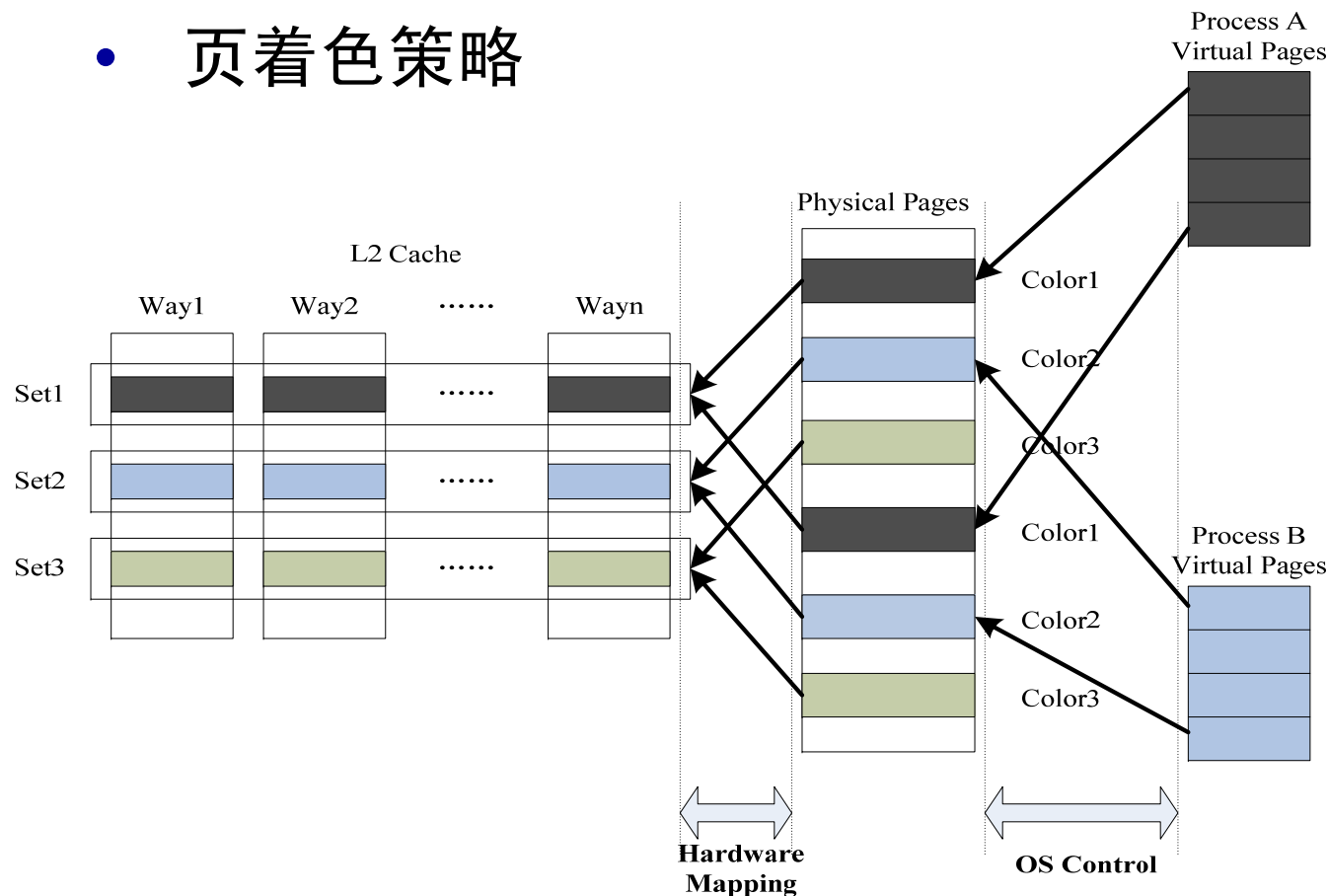
系统原型设计

- 难点和创新点

- 现有研究成果

- 研究进度安排

- 缓存划分
- 页着色策略



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 调度辅助缓存划分
 - 分类(MRC)

$$\text{计算 } D_i = \frac{(MRC_i(\min) - MRC_i(\max))}{MRC_i(\min)}$$

若 $D_i > r_1$ 则归类为缓存敏感型(sensitive)

若 $D_i < r_2$ 则归类为不敏感型(insensitive)

否则为比较敏感型(moderate)



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案
线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 调度辅助缓存划分

- 根据页着色策略实验的结果

- 初步结果 (SS & SI)

- 调度策略实现 (可扩展性):
gang scheduling; 处理器着色



研究方案

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 应用程序的局部性可能呈周期性^[28]
 - 周期检测算法
复杂; 有的程序呈非周期变化
 - 每隔固定指令数重新计算MRC



研究方案

- 选题背景

- 相关研究

- 研究内容

- 研究目标

- 研究方案

线程迁移

软件控制缓存

系统原型设计

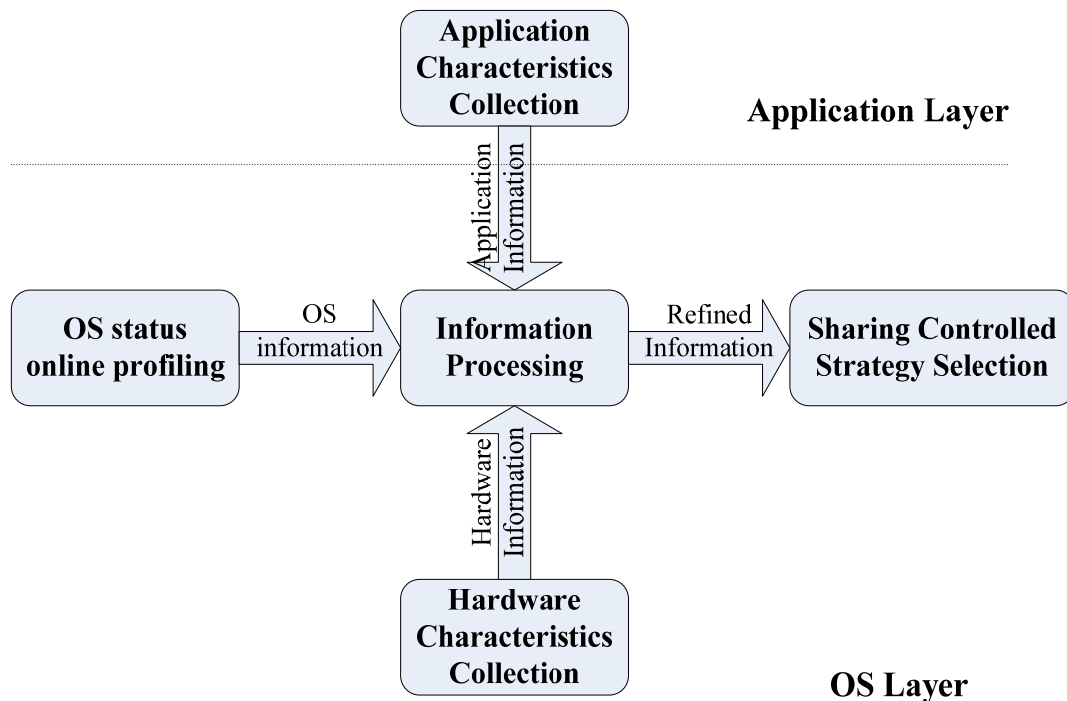
- 难点和创新点

- 现有研究成果

- 研究进度安排

融合多项共享控制技术

- 应用特征收集模块(e.g., 缓存特征, 系统调用频率)
- 硬件特征收集模块(e.g., 互联结构, 计算单元异构性)
- 在线监测模块 采集操作系统状态信息
- 共享控制策略选择模块



回顾

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

线程迁移

软件控制缓存

系统原型设计

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 要解决的关键问题

- 操作系统中的共享控制问题

- 提出的方案

- 锁感知的线程迁移策略(内核数据共享)
- 软件控制的硬件缓存管理策略(硬件缓存共享)
- 基于共享控制的操作系统原型



难点

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 线程迁移的动态性和负载均衡 (load balancing) 的延迟导致线程持锁时间的测量有误差
- 用于缓存划分的页着色策略引入了物理页复制的时间开销
- 程序行为的动态变化增加了系统设计的难度



可能的创新点

- 选题背景

- 相关研究

- 研究内容

1. 研究目标

2. 研究方案

3. 难点和创新点

- 现有研究成果

- 研究进度安排

- 针对内核数据结构的共享控制问题，提出了锁感知线程迁移策略并且在实际的多核系统上进行实现
- 为解决多核平台上的缓存竞争问题，提出软件控制的共享缓存管理策略及实现
- 融合多项共享控制技术，研究开发基于共享控制的操作系统原型



现有研究成果—论文

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- [1] Yan Cui, Yu Chen, Yuanchun Shi, Qingbo Wu. Parallel Scalability Comparison of Commodity Operating Systems on Large Scale Multi-cores. In Proceedings of the workshop on the interaction between Operating Systems and Computer Architecture, in conjunction with ISCA-36 (WIOSCA 2009).
- [2] Yan Cui, Yu Chen, Yuanchun Shi. OSMARK: A Benchmark suite for Understanding Parallel Scalability of Operating Systems on Multi-cores. In Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009). (EI source)
- [3] Yan Cui, Yu Chen, Yuanchun Shi. Improving Kernel Scalability by Lock-Aware Thread Migration. In Extended Abstract of Parallel Architectures and Compilation Techniques (PACT 2009). (EI source)
- [4] 崔岩, 史元春, 陈渝. 位置传感器网络中移动目标定位精度的视觉测量方法. 第三届和谐人机环境联合学术会议 (HHME 2007).
- [5] Yan Cui, Yu Chen, Yuanchun Shi. Scaling OLTP Applications on Multi-core Platforms. In Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS 2009). (Submitted) (EI source)
- [6] Wei Jiang, Yisu Zhou, Yan Cui, Wei Feng, Yu Chen, Yuanchun Shi, Qingbo Wu, "CFS Optimizations to KVM Threads on Multi-Core Environment", The Fifteenth International Conference on Parallel and Distributed Systems (ICPADS'09)(EI source)
- [7] Shen Wang, Yu Chen, Wei Jiang, Peng Li, Ting Dai, and Yan Cui. Fairness and Interactivity of Three CPU schedulers in Linux. In Proceedings of 15th IEEE International Conference on Embedded and Real-Time Computing System and Applications (RTCSA 2009) (EI source)



课题相关项目

- 选题背景

- 相关研究

- 研究内容

- 现有研究成果

- 研究进度安排

- Intel's Research Foundation: “Key System Software Technologies for Using Shared Last Level Cache on Many Cores”
- IBM's Research Foundation: “Key System Software Technologies for Using Many Cores without shared cache”
- 核高基 国家重大专项: “国产操作系统参考实现” (参与申请)
- 核高基 国家重大专项: “面向新型网络应用模式的网络化操作系统” (参与申请)



研究进度安排

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- 2009年10月-2010年4月
设计锁感知的线程迁移策略, 在Linux上完成原型系统并且验证可扩展性
- 2010年5月-2010年12月
设计软件控制的缓存管理策略, 在Linux上完成原型系统并且验证可扩展性
- 2011年1月-6月
整合各项关键技术, 设计共享控制的操作系统原型
- 2011年7月-12月
系统性能测试
- 2012年1月-4月
撰写博士学位论文
- 2012年6月
博士学位论文答辩



主要参考文献(1)

• 选题背景

• 相关研究

• 研究内容

• 现有研究成果

• 研究进度安排

- [1]. Yan Cui, Yu Chen, Yuanchun Shi, Qingbo Wu. Parallel Scalability Comparison of Commodity Operating Systems on Large-scale Multi-Cores. In Proceedings of WIOSCA 2009.
- [2]. Massalin, H., And Pu, C. A Lock-free multi-processor OS kernel. In Technical Report No. CUCS-005-91(1991).
- [3]. Mellor-Crummey, J., And M.L.Scott. Algorithms for scalable synchronization on shared-memory multiprocessors. ACM Trans. Comput. Syst, 9 1(1991), 21-56.
- [4]. Silas Boyd-Wickizer, Robert Morris, M.F.K. Reinventing scheduling for multicore systems. In Proceedings of the 12th Workshop on Hot Topics in Operating Systems. (HotOS 2009)
- [5]. Boyd-Wickizer, S., Chen, H., Chen, R., Mao, Y., Kashoek, F., Morris, R., Pesterev, A., Stein, L., Wu, M., Dai, Y., Zhang, Y., And Zhang, Z. Corey: An Operating Systems for many cores. In Proceedings of the 8th USENIX Symposium on Operating Systems Design and Implementation (2008).
- [6]. Sridharan, S., Keck, B., Murphy, R., Chandra, S., And Kogge, P. Thread migration to improve synchronization performance. In Workshop on Operating System Interference in High Performance Applications(2006).
- [7]. Suleman, M., A., Mutlu, O., Qureshi, M.K., And Patt, Y.N. Accelerating critical section execution with asymmetric multi-core architectures. In the Proceedings of International Conference on Architectural Support for Programming Language and Operating Systems(2009), pp. 253-264.
- [8]. TAM,D., AZIMI,R., And Stumm, M. Thread clustering: Sharing-aware scheduling on SMP-CMP-SMT Multiprocessors. In proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems (New York, NY, USA, 2007), ACM, pp. 47-58.
- [9]. G.E.Suh, L.Rudolph, and S.Devadas. Dynamic partitioning of shared cache memory. The Journal of SuperComputing, 28(1):7-26, 2004



主要参考文献(2)

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- [10]. M.K.Qureshi and Y.N.Patt. Utility-based cache partitioning: A low-overhead, high-performance, runtime mechanism to partition shared caches. In Proc. MICRO'06, pages 423-432, 2006.
- [11]. D.Tam, R.Azimi, L.Soares, and M.Stumm. Managing shared L2 caches on multicore systems in software. In WIOSCA 2007.
- [12]. E.Berg and E.Hagersten, Fast Data-Locality profiling of native execution. In SIGMETRICS, 2005.
- [13]. Jiang Lin, Qingda Lu, Xiaodong Zhang, and P. Sadayappan. Gaining Insights into Multicore Cache Partitioning : Bridging the Gap Between Simulation and Real Systems.
- [14]. Alexandra Fedorova, Margo Seltzer, Michael D.Smith and Christopher Small. CASC: A Cache-Aware Scheduling Algorithm for Multithreaded Chip Multiprocessors.
- [15]. Alexandra Fedorova, Margo Seltzer, Christopher Small and Daniel Nussbaum. Performance of Multithreaded Chip Multiprocessors and Implications for Operating System Design.
- [16]. Yuanlian Jiang, Xipeng Shen, Chen Jie, Rahul Tripathi. Analysis and Approximation of Optimal Co-Scheduling on Chip Multi-processors. In Proceedings of 17th international conference on Parallel architectures and compilation technologies.
- [17]. Mohan Rajagopalan, Brian T. Lewis, Todd A. Anderson. Thread Scheduling for Multi-Core Platforms. In Proceedings of 11th USENIX workshop on Hot Topics in operating systems.
- [18]. Gough, C. SIDDHA, S., And Chen, K. Kernel scalability-expending the horizon beyond fine grain locks. In Proceedings of the Linux Symposium 2007, pp. 153-165.
- [19]. Knauerhase, R., Brett, P., Hohlt, B., Li, T., And Hann, S. Using OS observations to improve performance in multicore systems. In Proceedings of the IEEE Micro(2008), pp54-66.
- [20]. Magnussen, P., Landin, A., And Hagersten, E. Queue locks on cache coherent multi-processors. In Proceedings 8th International Symposium on Parallel Processing (IPPS)(1994), pp. 165-171.



主要参考文献(3)

- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

- [21]. WENTZLAFF, D., And AGARWAL, A. The case for a factored operating systems (fos). In MIT-CSAIL-TR-2008-060 (2008).
- [22]. Xiao Zhang, Sandhya Dwarkadas, Kai Shen. Towards Practical Page Coloring-based Multi-Core cache Management. In Proceedings of the 4th ACM European conference on Computer Systems.
- [23]. G.E.Suh, S.Devadas, and L.Rudolph. A new memory monitoring scheme for memory-aware scheduling and partitioning. In Proc. HPCA 2002, pages 117-128.
- [24]. B.Veal and A.Foong. Performance Scalability of a Multi-core Web Server. In Proceedings of the 3rd ACM/IEEE Symposium on Architecture for networking and communications systems, pages 57-66, New York, NY, USA, 2007. ACM.
- [25]. B.Gamsa, O.Krieger, J.Appavoo, and M.Stumm. Tornado: Maximizing locality and concurrency in a shared memory multiprocessor operating system. In Proceedings of the 3rd USENIX Symposium on Operating Systems Design and Implementation, pages 87-100, Feb 1999.
- [26]. J.Aas. Understanding the Linux 2.6.8.1 CPU scheduler, Feb 2005.
- [27]. Yan Cui, Yu Chen, Yuanchun Shi. Improving Kernel Scalability by Lock Aware Thread Migration. In Extended Abstract of Parallel Architectures and Compilation Technologies (PACT 2009).
- [28]. Timothy Sherwood, Erez Perelman, and Brad Calder. Basic Block Distribution Analysis to Find Periodic Behavior and Simulation Points in Applications. In Proceedings of International Conference on Parallel Architectures and Compilation Technologies (PACT 2001).



- 选题背景
- 相关研究
- 研究内容
- 现有研究成果
- 研究进度安排

致谢

感谢参加评审选题报告的
各位老师和同学!

欢迎大家提问!

