

并行程序设计简介

使用Intel TBB线程库

李博群

深圳大学，计算数学

a14331990@163.com

5月29日，2014年

提纲

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

- 1 首页
- 2 提纲
- 3 前言
- 4 Intel TBB线程库简介
- 5 TBB的任务调度模型
- 6 TBB提供的常用并行策略
- 7 矩阵运算
- 8 格雷码解相位
- 9 参考文献
- 10 思考
- 11 结束

前言

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

设计并行程序的目的：

- 1. 节省时间和金钱；
- 2. 解决大规模或更复杂的问题；
- 3. 更好地利用并行硬件。

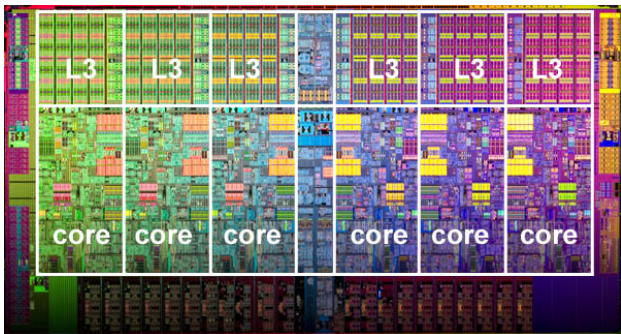


Figure: Intel Xeon六核处理器

Intel TBB线程库简介

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

Intel TBB线程库(Threading Building Blocks)是由Intel开发的一个适用于多核CPU并程序设计的C++模板库。它包含大量的适用于并程序设计的数据结构与算法，将并行计算单元抽象为任务，将任务分配到各个CPU核心上，并动态平衡各个CPU核心上的负载。这使得程序员无需手动操作线程，简化了并程序设计流程。

任务调度模型

并行程序设计 简介

李博群

TBB的任务调度模型

TBB应用任务窃取(task stealing)算法来实现多核负载平衡使得多核利用率以及性能得到提高。

- 1. 首先，所有任务被均匀分配到各个CPU核心上；
- 2. 当某个CPU核心完成了工作时，它将从其他仍在工作的CPU核心窃取任务。

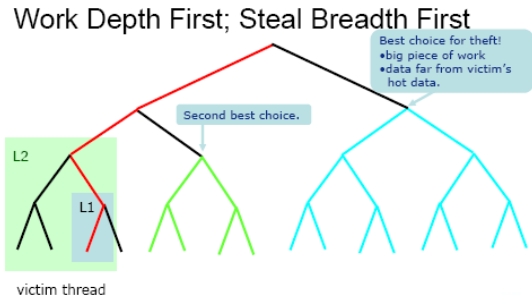


Figure: 任务窃取示意图

TBB提供的常用并行策略

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

**TBB提供的常
用并行策略**

矩阵运算

格雷码解相位

参考文献

思考

结束

TBB提供的常用并行算法框架有：

- 1. `parallel_for`，用于并行化for循环；
- 2. `parallel_reduce`，用于并行化分治算法。

parallel_for, 用法与实例

`parallel_for`递归地将for循环的范围划分为小范围，在每个小范围上执行计算，而小范围上的计算是通过重载操作符()来实现的。

实例：将两个矩阵相加，代码片段-小范围的计算

```
// Block operation
void TBBMatrix_Multiply_Parallel::operator()(const blocked_range2d<size_t> r) const
{
    TBBMatrix& A=my_A;
    TBBMatrix& B=my_B;
    TBBMatrix& C=my_C;
    for (size_t i = r.rows().begin(); i != r.rows().end(); ++i)
    {
        for (size_t j = r.cols().begin(); j != r.cols().end(); ++j)
        {
            double sum=0;
            for(int k=0;k<K;++k)
            {
                sum+=A.Get(i,k)*B.Get(k,j);
            }
            C.Set(sum,i,j);
        }
    }
}
```

Figure: 矩阵相加代码-小范围的计算

parallel_for, 用法与实例-续

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

实例：将两个矩阵相加，代码片段-初始任务划分

```
// Parallel matrix multiplication, 2d grain size: 10,10, or auto grain size
void Multiply_Parallel(TBBMatrix& C, TBBMatrix& A, TBBMatrix& B)
{
    int M=A.GetRowNum();
    int N=B.GetColNum();
    ...
    static affinity_partitioner ap;
    parallel_for( blocked_range2d<size_t>(0, M, 0, N),
        //parallel_for( blocked_range2d<size_t>(0, M, 10, 0, N, 10),
        ...      TBBMatrix_Multiply_Parallel(C,A,B), ap);
}
```

Figure: 矩阵相加代码-初始任务划分

parallel_reduce, 用法与实例

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

parallel_reduce递归地将计算任务划分为小任务，每个小任务的计算是通过重载操作符()来实现的，最后合并所有小任务的计算结果

实例：将1000个数相加，代码片段-小计算任务与计算结果合并

```
void operator()( const blocked_range<float*>& r ) {  
    float temp = value;  
    for( float* a=r.begin(); a!=r.end(); ++a ) {  
        temp += *a;  
    }  
    value = temp;  
}  
void join( Sum& rhs ) {value += rhs.value;}
```

Figure: 1000个数相加代码-小计算任务与计算结果合并

parallel_reduce, 用法与实例-续

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

实例：将1000个数相加，代码片段-初始任务划分

```
float ParallelSum( float array[], size_t n ) {  
    Sum total;  
    parallel_reduce( blocked_range<float*>( array, array+n ),  
                    total );  
    return total.value;  
}
```

Figure: 1000个数相加代码-初始任务划分

TBBMatrix

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

- 基于Intel TBB线程库的通用矩阵库
- 串行的矩阵加法和乘法
- 并行的矩阵加法和乘法

TBBMatrix的性能测试

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

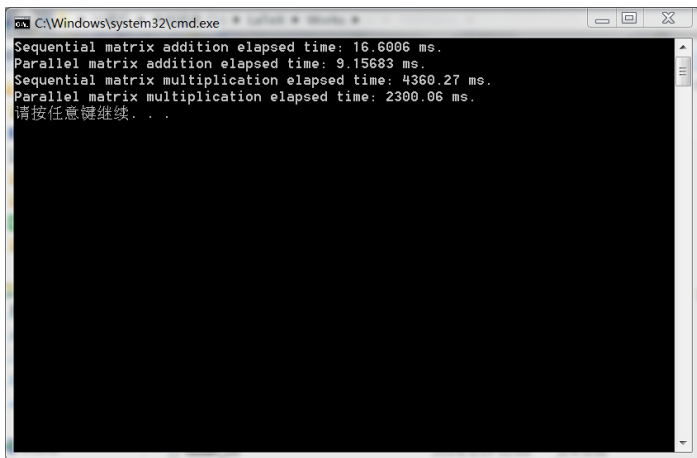
矩阵运算

格雷码解相位

参考文献

思考

结束



```
C:\Windows\system32\cmd.exe
Sequential matrix addition elapsed time: 16.6006 ms.
Parallel matrix addition elapsed time: 9.15683 ms.
Sequential matrix multiplication elapsed time: 4360.27 ms.
Parallel matrix multiplication elapsed time: 2300.06 ms.
请按任意键继续...
```

Figure: TBBMatrix的性能，测试计算机的处理器为“Intel Pentium Dual-Core E5500”

TBB_OPhase_graycode

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

- 基于Intel TBB线程库的格雷码解相位代码
- 串行的格雷码解相位代码
- 并行的格雷码解相位代码

TBB_OPhase_graycode的性能测试

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

```
E:\Microsoft\Projects\OPhase_graycode\Release>OPhase_graycode.exe  
'getPhase_graycode' time: 614.461ms  
'getPhase_graycode_Parallel' time: 226.009ms  
231.807  
231.807  
Same result!
```

Figure: TBB_OPhase_graycode的性能，测试计算机的处理器为“Intel Pentium Dual-Core E5500”

参考文献

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

- Intel Threading Building Blocks User Guide
- Intel Threading Building Blocks Reference Manual
- The Art of Concurrency, Clay Breshears, O'Reilly, 2009

思考

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

并行程序设计是一项较复杂的任务，可以分解为以下几个阶段

- 1. 设计一个有合理效率的串程序，注意这一点，模块化设计可以为以后的并行化做准备；合理效率是必要的，并行化一个效率很低的算法没有实际意义；
- 2. 通过分析找出关乎程序整体性能的关键模块；
- 3. 对关键模块采用恰当的并行策略加以并行化；
- 4. 调试，优化以及维护。

并行程序设计
简介

李博群

首页

提纲

前言

Intel TBB线程
库简介

TBB的任务调
度模型

TBB提供的常
用并行策略

矩阵运算

格雷码解相位

参考文献

思考

结束

谢谢!