Stephen Blair-Chappell and Andrew Stokes, 罗秋明, 孔畅, 刘成健等译,Intel Parallel Studio环境下的并行程序设计. 清华大学出版社, 2013

## 1、Intel的并行模型

Cilk Plus Threading Building Block (TBB) C++

行业相关的库：Intel Performance Primitives (IPP)

Intel Math Kernel Library (MKL)，包含BLAS,LAPACK,SCALEPACK等

现有标准：Intel MPI, OpenMP, Coarray Fortran, OpenCL

## 2、Parallel Studio XE概览

OpenMP: Intel编译器参数， /Qopenmp

Parallel Studio XE： Windows Linux

Parallel Studio: Windows

第2篇 Parallel Studio XE教程

## 4、生成优质的代码

使用Windows版本的编译器选项，可使用option-mapping工具来找到Linux中等价的选项。

map\_opts -tl -lc -opts /Oy-

-t 用于设置目标OS，1(linux), w(windows)

-l 用于设置语言类型，c或者f(fortran)

代码优化的七个步骤：

1、不使用优化技术构建应用程序

/Od (-O0)：阻止编译器的优化行为

构建和运行程序：

nmake clean

nmake TARGET=intel.oopt CFLAGS=/Od

2、使用通用优化

4中粗粒度的优化开关选项：

/O1 优化代码长度与运行速度

/O2 优化运行速度

/O3

/Ox 完全优化

需要尝试，不一定优化级别越高就越好越快

用/Qopt-report 产生优化报告：/Qopt-report:0 (1, 2, 3)

3、使用处理器相关的优化

自动向量优化时Intel编译器的优化方法。是使用了CPU的单指令多数据流(SIMD)指令来加快执行时间。SSE2 SSE3 SSE4.1 -> AVX

自动向量化原则

启用自动向量化： /Qvec-

强化自动向量化： /Qx<architecture>

<architecture>可以使用SSE2 SSE3 SS4.1 SSE4.2 AVX

使用/Qx选项来加强自动向量化会引起两个问题：（1）程序将不能在非Intel处理器上运行；如AMD设备上运行，应当使用/arch选项，而不是/Qx

（2）在不支持所使用的选项的某一代CPU上运行优化代码，也不能运行。

**非Inel CPU的情况：**

使用/arch: <architecture>

<architecture>可以是ia32 SSE2 SSE3 SS4.1

判断自动向量化是否生效： /Qvec-report: n

n=1 仅报告成功的向量化的循环

n=2 同时报告成功和不成功的向量化的循环

。。。 n=5

使用向量化来构建程序：

nmake clean

nmake CFLAGS="/Qvec-" TARGET=novec

运行 novec.exe

4、增加过程间优化

过程间优化(interprocedural Optimization, IPO)

/Qipo

5、性能测评指导的优化

/Qprof-gen

/Qprof-use

构建在多种CPU上运行的应用程序：

/Qax (注意不是/Qx)

例子：/QaxSSE3

/QaxAVX /arch:SSE3

## 5、并行程序的调优

使用Amplifier XE命令行测出基准

amplxe-cl -collect concurrency 888.exe

可看到内容：平均并发度、运行时间、程序的等待时间

生成热点报告：

amplxe-cl -report hotspots -group-by openmp-task

识别并发热点：

amplxe-gui r000cc