

Intel Visual Fortran 应用程序开发

周振红 徐进军 毕苏萍
苗 丽 王宗敏

编 著

- ☐ Intel Visual Fortran Compiler 9.0 for Windows
- ☐ Visual Studio.NET (Visual C++.NET)
- ☐ Fortran QuickWin
- ☐ Fortran Windows
- ☐ Fortran Dynamic-link Library
- ☐ Creating Multithread Applications
- ☐ Programming with Mixed Languages
- ☐ Using Dialogs and Controls
- ☒ Intel Fortran Module Wizard
- ☐ Porting Applications from CVF to IVF

 黄河水利出版社

Intel Visual Fortran

应用程序开发

周振红 徐进军 毕苏萍

编著

苗 丽 王宗敏

黄河水利出版社

内 容 提 要

本书针对 Compaq Visual Fortran(CVF) 6.6 的后继编译器 Intel Visual Fortran(IVF) 9.0, 系统介绍 Visual Studio.NET 环境下各种 Fortran 应用程序的开发, 重点讲解 Fortran QuickWin、Fortran Windows、动态链接库及多线程应用程序的开发, 以及对话框和控件、自动化服务器和 ActiveX 控件的使用, 并全面、深入地探讨 Intel Fortran 与 Visual C++/Basic.NET 的混合编程。另外, 本书对 Visual Studio.NET 开发环境、Intel Fortran 9.0 编译器以及 CVF 向 IVF 的转换等也进行了简要介绍。

本书实例丰富、注重实用, 面向具有 Fortran 90/95 基础的中、高级读者, 适合作为理工科高年级本科生、研究生的教学及教学参考用书, 也可作为科学与工程计算领域研究、开发参考用书。

图书在版编目(CIP)数据

Intel Visual Fortran 应用程序开发 / 周振红等编著.
郑州: 黄河水利出版社, 2006.9
ISBN 7-80734-129-7

I . I . . . II . 周 . . . III . FORTRAN 语言-程序设计-
高等学校-教材 IV . TP312

中国版本图书馆 CIP 数据核字 (2006) 第 105517 号

组稿编辑: 岳德军 0371-66022217

出 版 社: 黄河水利出版社

地址: 河南省郑州市金水路 11 号

邮政编码: 450003

发行单位: 黄河水利出版社

发行部电话: 0371-66026940

传真: 0371-66022620

E-mail: hhslebs@126.com

承印单位: 黄河水利委员会印刷厂

开本: 787 mm × 1 092 mm 1 / 16

印张: 22.75

字数: 480 千字

印数: 1—2 000

版次: 2006 年 9 月第 1 版

印次: 2006 年 9 月第 1 次印刷

书号: ISBN 7-80734-129-7 / TP · 26

定价: 38.00 元

前言

Microsoft.NET Framework 是微软 2002 年开始推出的新一代计算平台,它简化了在高度分布式 Internet 环境中的应用程序开发。.NET Framework 以公共语言运行库和.NET Framework 类库为基础平台,它为 Visual Basic.NET、Visual C++.NET 和 Visual C#.NET (统称 Visual Studio.NET) 提供了统一的开发环境,遗憾的是该环境不包括针对科学与工程计算的 Visual Fortran.NET 开发工具。

尽管 Fortran 语言标准发展很快,但支持它的编译器通常要滞后一到两年的时间。2004 年 5 月,在 ISO、IEC 的联合工作组 JTC1/SC22/WG5 以及美国 Fortran 委员会 NCITS/J3 的共同努力下发布了 Fortran 2003 标准;直到 2005 年 6 月,Intel 公司才正式发布 Intel Visual Fortran (IVF) Compiler 9.0,该编译器除完全支持 Fortran 90/95 标准外,还对 Fortran 2003 的部分功能给予支持,且其中的 IA-32 编译器能以插件方式集成到 Visual Studio.NET 开发环境、共享 Visual C++.NET 中的工具和运行库。

此前, Fortran 90/95 的 Win32 开发环境多采用 Compaq Visual Fortran (CVF) 6.x,但在 2005 年 CVF 开发团队加盟到 Intel 公司,HP 宣布其 CVF 6.6 截至 2005 年 12 月 31 日、IVF 9.0 将作为其新一代后继编译器。此编译器将 CVF 前端与英特尔处理器后端相结合,拥有 CVF 丰富的语言功能和英特尔处理器的代码生成及优化功能,使运行在英特尔平台上的程序性能得到大幅度提高。所以,从 CVF 6.x(Visual C++ 6.0)转移到 IVF 9.0(Visual C++.NET 2002/2003)已是势在必行。

从对 Fortran 90/95 语言标准支持方面看,IVF 9.0 是 CVF 6.6 的升级版本,但在许多方面做出了改进,且所有的系统库都重新加以实现;若从开发环境方面看,两者却差异很大,比如,Visual Studio.NET 解决方案资源管理器中的一个工程只能由一种语言的源程序组成,而 Visual Fortran 6.6(Visual C++ 6.0)项目空间中的一个工程可由 Fortran 和 C++(C)两种语言的源程序组成;况且,Intel Fortran 9.0 的联机文档大部分借用了 Visual Fortran 6.6 的,当中有些部分并不合适,缺乏完整的开发实例,致使其推广应用受到一定程度的限制。有鉴于此,我们在对 Intel Fortran 9.0 全面认识、深入探究的基础上编撰了本书,书中内容涵盖了 Visual Studio.NET 环境下各种 Fortran 应用程序的开发,甚至包括 Intel Fortran 与 Visual C++/Basic.NET 的混合编程。

本书第 1 章、第 9 章、第 10 章由周振红(郑州大学)、王宗敏(郑州大学)共同编

写,第2章至第4章由徐进军(武汉大学)编写,第5章、第6章由毕苏萍(郑州大学)编写,第7章、第8章由苗丽(郑州大学)编写,全书由周振红统编。

本书的出版得到了郑州大学水利工程一级学科(博士点)建设资金资助。

感谢读者选择使用本书,欢迎您对本书内容提出批评和建议,我们不胜感激。

作者的联系方式为,电子邮件: zzh6374@yahoo.com.cn; 通信地址: (450002) 郑州大学环境与水利学院,周振红。

作 者

2006年6月

导 读

第 1 章 IVF 编译器及 CVF 向 IVF 的转换。介绍 IVF 9.0 的主要功能及特点、IVF 9.0 软件包的安装, 以及 CVF 向 IVF 的转换。

第 2 章 .NET 开发 Fortran 应用程序。介绍 .NET 开发环境的设置、各种类型的 Intel Fortran 应用程序及其在 .NET 环境下的创建。

第 3 章 QuickWin 绘图和输出文本。介绍常用的 Fortran QuickWin 应用程序的子窗口操作、绘图选项的设置以及图形和文本(包括字符文本和字体文本)的输出。

本章示例程序 15 个。

第 4 章 QuickWin 界面定制。内容包括对 QuickWin 框架窗口的菜单、图标和标题进行定制, 字符串及提示文字的本地化, 以及子窗口客户区的鼠标输入。

本章示例程序 14 个。

第 5 章 使用对话框和控件。内容包括对话框程序的界面设计、代码编写, 以及 Windows 常用控件的使用。

本章示例程序 3 个。

第 6 章 设计 Windows 应用程序。简要介绍 Windows 应用程序的结构, 在此基础上讲述无模式对话框、单文档(SDI)和多文档(MDI)应用程序的设计, 并进一步探讨 SDI 程序中使用无模式对话框、设计菜单和绘制图形的问题。

本章示例程序 3 个。

第 7 章 使用 COM 组件。先引入相关的 COM 组件技术, 包括组件对象模型(COM)、自动化和 ActiveX 控件技术, 然后讲解如何利用 Intel Fortran 模块向导来操作自动化服务器和 ActiveX 控件。

本章示例程序 3 个。

第 8 章 创建多线程应用程序。先给出进程、线程及其优先级的基本概念, 然后讲述线程的创建、终止等基本操作, 使用临界段、互斥体、信号计数器和事件对象的线程同步以及线程的局部存储, 多进程应用程序的创建, 并给出多线程操作的相关例程。

本章示例程序 2 个。

第 9 章 创建和使用动态链接库。先介绍动态链接库(DLL)的基本概念及其组成等, 然后重点讲解如何在调用程序和 DLL 被调用程序(两者位于同一进程)间共享数据、例程及对话框资源, 如何跨进程共享 DLL 中的 Fortran 共用区数据。在 DLL

的使用上，则给出了隐式和显式两种链接方式。

本章示例程序 7 个。

第 10 章 Intel Fortran 与 Visual C++/Basic.NET 的混合编程。先简要介绍混合编程的优越性、可行性、调用约定的基本概念及.NET 环境下的混合编程实现方式，然后重点讲解如何在 Intel Fortran 与 Visual C++/.NET 间协调它们所使用的调用约定、命名约定，如何传递数据，如何匹配各种数据类型；针对 Intel Fortran 与 Visual Basic.NET 间的混合编程，则重点讲解如何传递 Intel Fortran 中的字符串、数组及派生类型三种特殊数据。最后，指出了基于静态库进行 Intel Fortran/Visual C++/.NET 编译和链接需要注意的特殊方面。

本章示例程序 21 个。

目 录

前 言

导 读

第 1 章 IVF 编译器及 CVF 向 IVF 的转换	(1)
1.1 IVF 编译器 9.0 Windows 版	(1)
1.1.1 标准版与专业版	(1)
1.1.2 优化及其他功能	(2)
1.1.3 兼容性	(6)
1.2 IVF 9.0 的安装	(7)
1.2.1 系统要求	(7)
1.2.2 安装支持软件	(9)
1.2.3 安装 IVF 9.0	(9)
1.2.4 示例开发环境	(13)
1.3 CVF 转换到 IVF	(13)
1.3.1 IVF 与 CVF 的兼容性	(13)
1.3.2 移植 CVF 工程	(14)
1.3.3 IVF 引入的变化	(16)
小 结	(19)
第 2 章 .NET 开发 Fortran 应用程序	(21)
2.1 设置 .NET 开发环境	(21)
2.1.1 编译器版本	(21)
2.1.2 搜索目录	(21)
2.1.3 自定义源文件扩展名	(22)
2.2 Fortran 应用程序类型	(23)
2.2.1 控制台应用程序	(24)
2.2.2 标准图形应用程序	(24)
2.2.3 QuickWin 应用程序	(26)
2.2.4 Windows 应用程序	(27)
2.2.5 静态链接库	(28)
2.2.6 动态链接库	(29)

2.3 创建工程	(31)
2.3.1 添加文件	(32)
2.3.2 选择配置	(32)
2.3.3 设置工程	(33)
2.3.4 生成可执行文件	(34)
2.3.5 宏指令	(35)
小 结	(37)
第 3 章 QuickWin 绘图和输出文本	(38)
3.1 操作子窗口	(38)
3.1.1 创建子窗口	(38)
3.1.2 焦点窗口与活动窗口	(40)
3.1.3 设置子窗口属性	(41)
3.1.4 控制窗口大小和位置	(43)
3.1.5 保持子窗口显示	(44)
3.2 设置图形输出选项	(46)
3.2.1 控制光标显示	(46)
3.2.2 设置绘图坐标系	(47)
3.2.3 设置颜色	(50)
3.2.4 设置图形属性	(51)
3.3 输出图形和文本	(56)
3.3.1 绘制图形	(56)
3.3.2 输出字符文本	(57)
3.3.3 输出字体文本	(60)
小 结	(74)
第 4 章 QuickWin 界面定制	(75)
4.1 菜单操作	(75)
4.1.1 设置初始菜单	(75)
4.1.2 删除、插入和追加菜单	(78)
4.1.3 修改菜单	(83)
4.1.4 调整子窗口列表的菜单位置	(85)
4.1.5 模拟鼠标选取菜单项的操作	(86)
4.2 本地化	(87)
4.3 信息提示	(89)
4.3.1 使用信息对话框	(89)

4.3.2 定制 About 对话框	(91)
4.4 图标和标题	(93)
4.4.1 定制图标	(93)
4.4.2 定制标题	(94)
4.5 鼠标输入	(99)
4.5.1 事件方法	(100)
4.5.2 阻塞方法	(103)
小 结	(105)
第 5 章 使用对话框和控件	(106)
5.1 设计对话框	(106)
5.1.1 插入对话框资源	(107)
5.1.2 布置对话框	(108)
5.1.3 设置对话框及其控件属性	(109)
5.1.4 保存对话框资源	(111)
5.2 编写对话框程序	(113)
5.2.1 引用对话框资源	(113)
5.2.2 操作对话框及其控件	(114)
5.2.3 编写控件回调例程	(116)
5.2.4 控制对话框的退出	(120)
5.3 使用控件	(121)
5.3.1 控件索引	(121)
5.3.2 使用控件	(125)
小 结	(144)
第 6 章 设计 Windows 应用程序	(145)
6.1 Windows 应用程序的结构	(145)
6.1.1 主函数	(145)
6.1.2 窗口函数	(149)
6.1.3 Windows 系统、主函数、窗口函数之间的关系	(150)
6.1.4 Fortran Windows 程序的结构	(153)
6.2 无模式对话框应用程序	(154)
6.3 SDI 和 MDI Windows 程序	(166)
6.3.1 SDI 和 MDI 界面	(166)
6.3.2 使用无模式对话框	(168)
6.3.3 使用菜单	(177)

6.3.4 绘制图形	(184)
小 结	(188)
第 7 章 使用 COM 组件	(189)
7.1 相关的 COM 技术	(189)
7.1.1 组件对象模型(COM)	(189)
7.1.2 自动化	(200)
7.1.3 ActiveX 控件	(201)
7.2 Intel Fortran 模块向导	(201)
7.3 使用自动化服务器	(203)
7.4 使用 Activex 控件	(213)
小 结	(224)
第 8 章 创建多线程应用程序	(226)
8.1 进程、线程及其优先级	(226)
8.1.1 进程、线程的基本概念	(226)
8.1.2 进程和线程的优先级	(226)
8.2 线程操作	(228)
8.2.1 线程的创建	(228)
8.2.2 线程的终止	(228)
8.2.3 线程的挂起和恢复	(229)
8.3 线程同步	(235)
8.3.1 临界段	(235)
8.3.2 互斥体	(236)
8.3.3 信号计数器	(237)
8.3.4 事件对象	(237)
8.4 线程局部存储(TLS)	(238)
8.5 进程操作	(238)
8.6 多线程操作的相关例程	(253)
小 结	(255)
第 9 章 创建和使用动态链接库	(256)
9.1 动态链接库概述	(256)
9.1.1 动态链接库的基本概念	(256)
9.1.2 动态链接库的组成	(257)
9.2 动态链接库的创建	(259)
9.2.1 导出声明	(259)

9.2.2	共享 DLL 中的例程	(260)
9.2.3	共享 DLL 中的数据	(264)
9.2.4	跨进程共享 DLL 中的共用区数据	(267)
9.2.5	共享 DLL 中的对话框资源	(271)
9.3	动态链接库的使用	(289)
9.3.1	隐式链接	(289)
9.3.2	显式链接	(289)
小 结		(293)
第 10 章	Intel Fortran 与 Visual C++/Basic.NET 的混合编程	(295)
10.1	混合编程概述	(295)
10.1.1	混合编程的优势	(295)
10.1.2	混合编程的前提条件	(296)
10.1.3	调用约定	(296)
10.1.4	.NET 下的混合编程实现方式	(297)
10.2	协调调用约定	(298)
10.2.1	属性编译指令和调用约定	(298)
10.2.2	堆栈清理及可选参数传递	(300)
10.2.3	匹配 Intel Fortran 与 C/C++ 的调用约定	(303)
10.3	协调命名约定	(304)
10.3.1	Intel Fortran 和 C/C++ 的命名约定	(305)
10.3.2	协调目标实体名的大、小写	(306)
10.3.3	Intel Fortran 模块命名规则	(308)
10.4	传递数据	(309)
10.4.1	例程参数	(309)
10.4.2	模 块	(310)
10.4.3	外部变量	(314)
10.5	匹配数据类型	(319)
10.5.1	数值型、复数型及逻辑型	(320)
10.5.2	字符串	(322)
10.5.3	指 针	(330)
10.5.4	数 组	(333)
10.5.5	派生类型	(338)
10.6	Intel Fortran/Visual C++ .NET 的编译和链接	(341)
10.6.1	运行库的一致性	(341)

10.6.2 项目依赖项	(343)
10.6.3 Intel Fortran 支持库	(344)
小 结	(347)
参考文献	(349)

第 1 章 IVF 编译器及 CVF 向 IVF 的转换

2005 年 6 月 14 日,英特尔公司正式发布了 Intel Visual Fortran(IVF)Compiler 9.0 for Windows, 以作为 Compaq Visual Fortran(CVF)6.6 的后继编译器。此编译器将 CVF 前端与针对英特尔处理器系列的 IVF 编译器后端相结合, 拥有 CVF 丰富的语言功能与 IVF 编译器的代码生成及优化功能。此编译器包含标准版与专业版, 专业版包含 Visual Numerics 公司的 IMSL Fortran 函数库 5.0。

本章主要内容:

- IVF 编译器的优化功能
- IVF 9.0 的安装
- CVF 向 IVF 的转换

1.1 IVF 编译器 9.0 Windows 版

使用 Intel Visual Fortran(IVF) 9.0 编译器(Windows 版), 可以使开发的应用程序在现有的英特尔处理器与体系结构、多核心英特尔处理器及最新英特尔处理器上获得前所未有的执行性能。

1.1.1 标准版与专业版

IVF 编译器 Windows 版有“标准版”与“专业版”两个版本。“标准版”包含英特尔编译器、英特尔数组可视化器、英特尔调试器以及其他一些功能。“专业版”拥有“标准版”的所有功能, 此外, 还包含 Visual Numerics 的“IMSL Fortran 函数库 5.0”。

“IMSL Fortran 函数库 5.0”将世界知名的 IMSL F90 函数库、并行处理功能及

IMSL Fortran 77 函数库集成到一个紧密相关的软件包中。它包含强大而灵活的新接口模块,可以使用所有高级 Fortran 语法与可选参数,同时仍提供真正的向后兼容性。这些接口模块可降低发生错误的可能性,简化并加速编码工作,同时还可提高代码质量。同时,还可以促进开发更简洁的 Fortran 应用程序,为熟练的程序员提供深入到底层并取得完全控制的能力。“IMSL Fortran 函数库 5.0”包含许多新的、有价值的数值算法,并扩展了对 SMP 并行处理的支持,在此基础上,可以开发出复杂的数值分析应用程序。

1.1.2 优化及其他功能

IVF 编译器开发的应用程序,能以较高的速度运行在英特尔 IA-32 处理器、英特尔 EM64T 处理器及安腾 2 处理器上。主要优化功能有:支持 Pentium 4 和 Pentium M 的第三代数据流单指令多数据扩展指令集、安腾 2 的软件管道、过程间优化(IPO)、档案导引优化(PGO)等,通过自动并行和 OpenMP 技术支持多线程开发。具体包括以下几方面。

1.1.2.1 经过优化的浮点指令吞吐能力

在 IA-32 系统上,IVF 编译器使用堆栈来高效执行浮点(FP)指令。由于重叠指令的计算结果可放入任何堆栈寄存器,应用程序性能因此得以提高。安腾 2 处理器提供可直接寻址的浮点寄存器(支持经管道化处理的浮点循环),与采用传统体系结构的处理器相比,它需要的加载与存储操作次数大为减少。这样,运行应用程序时的执行速度就会快许多。

1.1.2.2 过程间优化(IPO)

对于包含许多常用中、小函数的程序,特别是循环内包含调用的程序,IPO 可以极大地提高应用程序性能。

1.1.2.3 档案导引优化(PGO)

PGO 编译过程可以使 IVF 编译器更好地利用处理器微体系结构,更有效地使用指令调度与高速缓存,并可以更好地执行分支预测。它通过重新组织代码布局、缩短代码长度并减少分支预测失误来减少指令缓存反覆,从而提高应用程序性能。

1.1.2.4 仅限 IA-32 体系结构

(1)支持第三代数据流单指令多数据扩展指令集。32 位功能支持“第三代数据流

单指令多数据扩展指令集”，使得奔腾 4 处理器引入的“英特尔 NetBurst”微体系结构与众不同。此编译器也继续支持传统的性能改善功能，如 MMX 技术。“第三代数据流单指令多数据扩展指令集”超越了改善应用程序多媒体或图形组件性能这一初衷，它包含了更强劲的性能，可以满足进行浮点与双精度计算的需要。这些新指令可以通过许多途径来支持，其中包括内嵌 ASM、编译器内建函数、类库、矢量器以及英特尔性能库。

(2)自动矢量器。提供可以自动对代码进行并行化处理的矢量器，以最大限度利用处理器的潜在能力。此矢量器包含齐全的文档，其中的示例显示了如何提高应用程序的执行速度。矢量器还提供多项功能，包括支持先进的动态数据调整策略，其中包含可以生成平衡负载的循环剥离技术，以及可以匹配整个缓存线预取情况的循环展开技术。

(3)对各代英特尔处理器的运行时支持——处理器调度。32 位功能提供了一个称为处理器调度的选项，可用于构建针对特定的某代英特尔处理器的应用程序。调度选项现在可以针对多个特定目标，可以开发针对最新英特尔处理器——奔腾 4 处理器的应用程序，同时还可以确保可执行文件能在以前的 IA-32 体系结构上正常运行。在一个可执行文件中，就可以使开发的应用程序既能利用最新处理器的性能优势，同时也可以基于以前英特尔体系结构的系统上保持很高的性能。

1.1.2.5 仅限英特尔安腾 2 微体系结构

(1)断定。先前的体系结构通过分支指令来实现条件执行，针对安腾 2 处理器的 IVF 编译器通过断定指令来实现条件执行。通过采用断定这项重要的优化措施，可以从程序序列中完全排除分支，从而形成更大的基本指令块，并消除相关的预测失误所造成的损失，这两项都有助于改善应用程序性能。由于使用断定之后存在的分支更少，控制流发生改变的概率也更小，因此动态指令获取的效率会更高。

(2)改进的分支预测。利用分支预测功能，处理器可以确定分支指令后面最可能的代码路径。处理器根据断定获取并开始执行最可能的代码路径上的指令。沿错误路径执行的指令的结果必须丢弃，并且必须获取并执行正确路径中的指令，所以分支预测失误会导致执行延迟。在安腾 2 微体系结构上，IVF 编译器与处理器可以互通分支信息，借此减少分支预测失误。它还可以使编译的代码能够利用运行时信息管理处理器硬件。这两项功能对断定起着补充作用，提供了多项性能优势：分支预测失误更少的应用程序运行速度更快，由仍可能出现的分支预测失误引起的性能开销得到降低，应用程序的缓存失误次数也更少。为确保应用程序在所有情况下均能正确运行，需要时会执行代码恢复过程。

(3)推测。开发人员可以利用推测技术在实际需要之前执行一些操作(如消耗很

高的加载指令)来提高性能。为确保代码正确,编译器会根据需要执行代码恢复过程,以确保在先前的推测发生失误时仍能正确执行所有受影响的操作。

(4)软件管道。通过支持软件管道,减少处理循环所需的时钟周期数。它会试图将每个循环分解成几个迭代阶段,每个阶段包含几条指令,并将这些迭代阶段搭接起来。软件管道技术允许同时存在多个循环迭代,并且不展开整个循环,所以在基于安腾 2 的系统上运行的应用程序中,它可以执行单周期、整循环计算。尽管并非所有循环都能得益于软件管道技术,但对于能够获益的那些循环而言,通过结合软件管道与断定、推测技术,可以显著减少代码扩展、路径长度及分支预测失误,从而实现应用程序性能上的飞跃。

1.1.2.6 支持英特尔扩展内存 64 位技术(EM64T)

包括使用英特尔 EM64T 开发高性能应用程序时所需的各项功能。

1.1.2.7 多线程应用程序支持

(1)OpenMP 支持。OpenMP 是可移植多线程应用程序开发领域的行业标准,在细粒度(循环级别)与粗粒度(函数级别)线程技术上具有很高的效率。对于将串行应用程序转换成并行应用程序,OpenMP 指令是一种容易使用且作用强大的手段,它具有使应用程序因为在多核心与对称多处理器系统上并行执行而获得大幅性能提升的潜力。IVF 编译器支持 OpenMP Fortran 2.0 版 API 规范(Workshare OpenMP 指令除外),并且能够为共享内存并行编程执行代码转换。利用 OpenMP Fortran 2.0 版,IVF 编译器可以支持多线程应用程序的开发与调试。

(2)自动并行。包含自动将循环线程化的自动并行功能,提供一个简便的方法以充分利用并行机制的优势,来提高应用程序在多处理器系统上的性能。它会检测能够安全地并行执行的循环,并自动为这些循环生成多线程代码。开发人员不需要处理迭代划分、数据共享、线程调度及同步等低级别的细节。它还具有从多处理器系统与支持“超线程技术”的系统获取性能提升的优势。优化功能的改善包括将并行循环矢量化能力,以及新增的“基于软件的推测性提前计算”功能,此功能可以改善应用程序在含“超线程技术”的英特尔处理器上的性能。

1.1.2.8 自动接口检查

调用例程时,对于难以诊断的应用程序错误,同一来源导致的错误的参数列表并不一致。Fortran 90 引入了“显式接口”,使得编译器能够检查正确的调用,但是过去已经写好的和以后将编写的许多程序都没有提供这些额外的信息。IVF 编译器 9.0 可以自动检查参数列表的一致性,甚至可以跨越多个源文件来进行这项检查。