



Accelerating Linux/IPF GCC Performance – *Project “Osprey”*

Shin-Ming Liu, Project Manager,
Java Compilers & Tools Lab, HP
April 2006



Overview

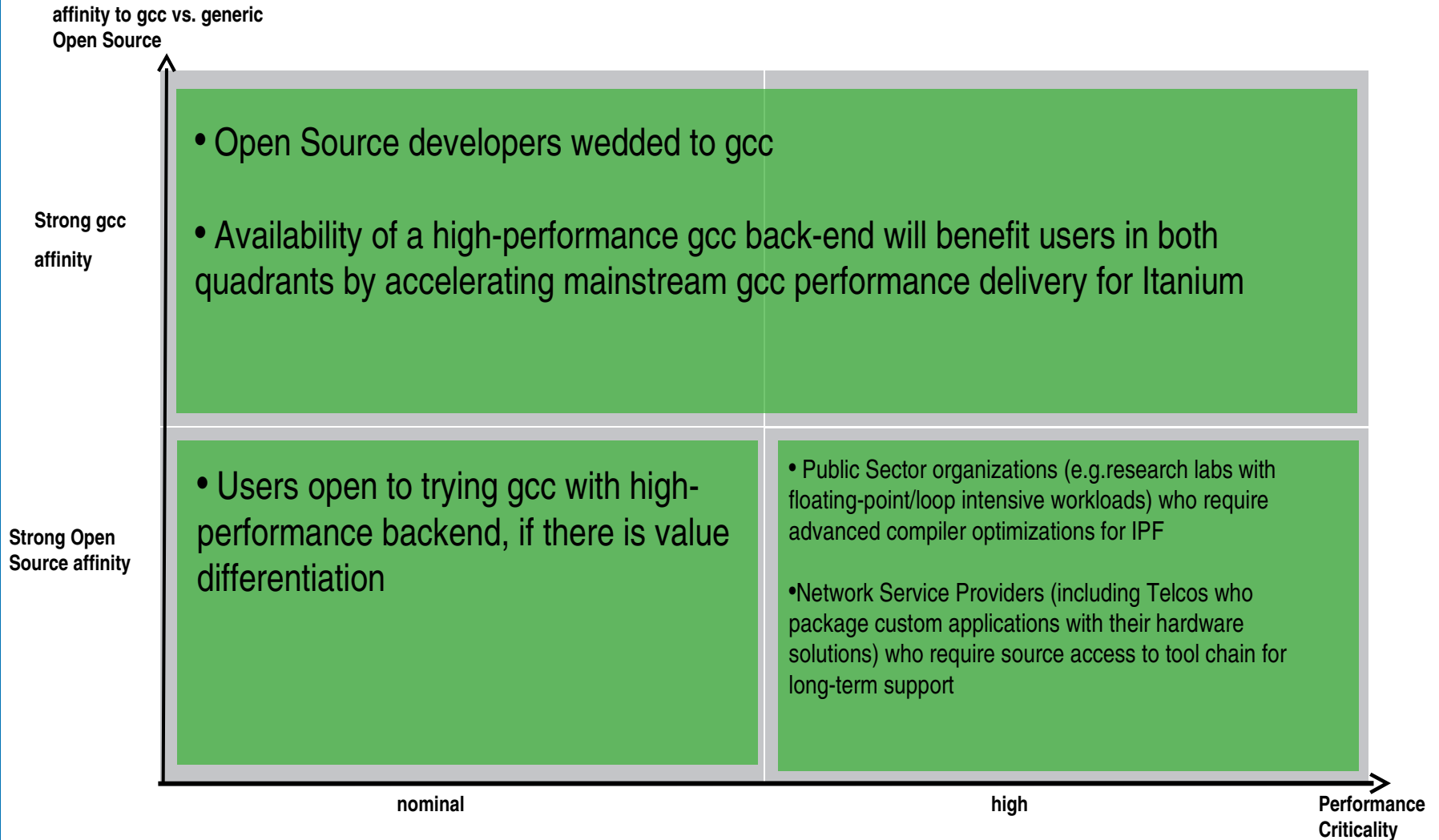
- General Background
 - Project Osprey Objectives
 - Current Status & Project Plan

GCC Performance Outlook for Itanium in Next Two Years



- Competitive positioning of Itanium processors requires high-performance compilers including gcc
 - Key gcc tree-SSA infrastructure work will benefit Itanium eventually. However, dramatic gcc performance improvement not in the cards near term
 - Competitive situation:
 - Sub-par gcc performance affects Linux/Itanium adoption
 - Strong gcc user preference for open source apps, other target markets
 - GNU f95 has long way to go performance-wise
 - Current gcc performance engineering efforts are vital (gcc 4.0, Gelato-sponsored efforts), however ...
- ***... the need for improved gcc performance for Itanium is urgent.***

Who will be best served by improved gcc performance for Itanium?



A new, complementary solution to accelerate gcc performance delivery



Project Osprey:

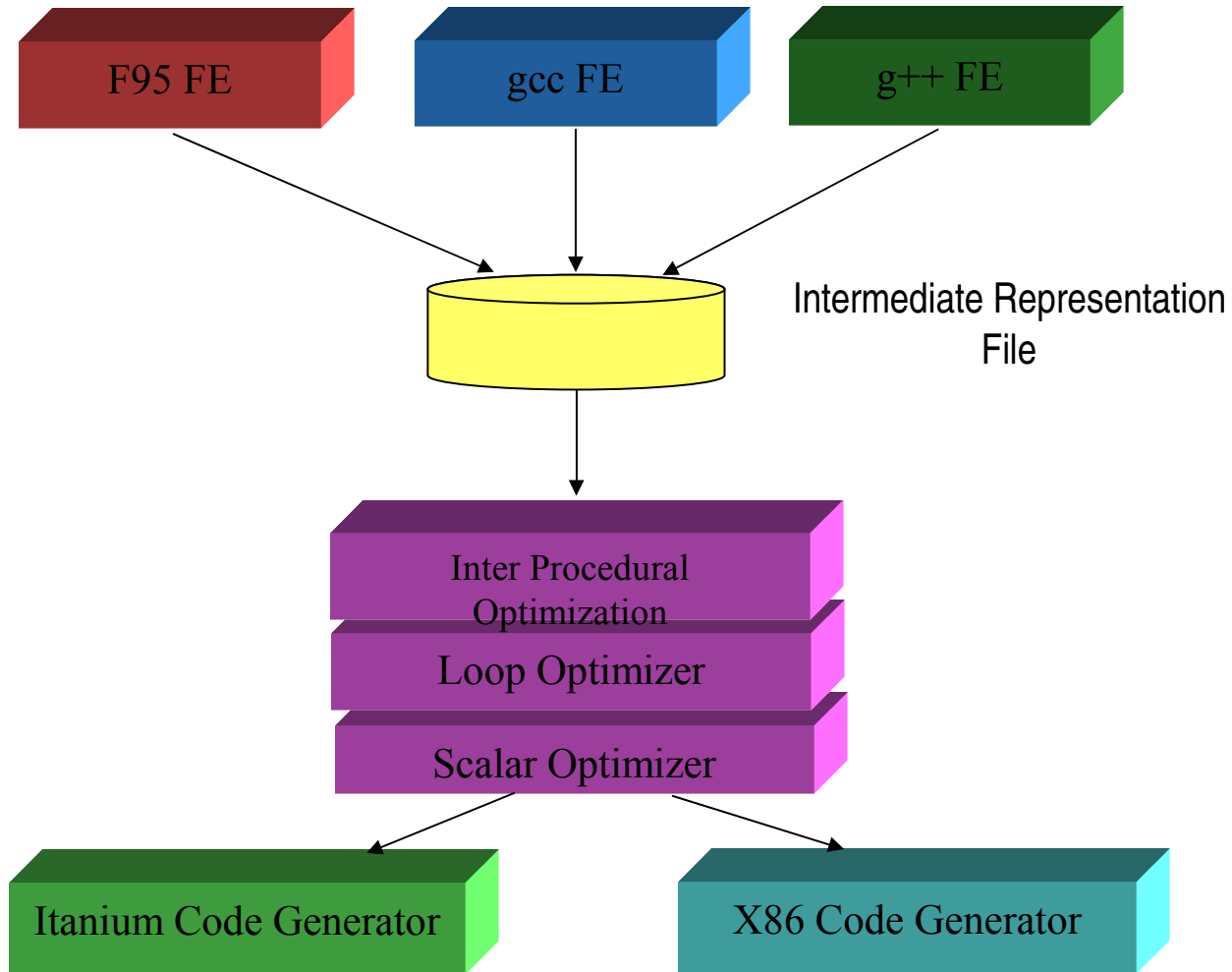
Leverage Open64 compiler technology to quickly deliver gnu-compatible high-performance, open-source, production-quality compiler suite for Itanium

Open64 Compiler Technology: Historical Background



- Derived from SGI MIPSpro compiler suite
- SGI did the initial port to Itanium
- SGI open sourced the MIPSpro compilers in Summer 2000 as the “Pro64” compiler suite
 - Gnu-front-end based C, C++ compilers
 - Cray-FORTRAN front-end based F95 compiler
- University of Delaware took over maintenance of “Pro64” compiler suite
 - New distribution re-named to be “Open64” compiler suite

Historical Background (cont'd)



Historical Background (cont'd)

- Intel enhanced the common Open64 compiler code Generator with advanced optimizations for Itanium
 - Itanium-enhanced Open64 compiler suite released as the “Open Research Compiler” (ORC) today to facilitate Itanium-related research efforts in academia
- Pathscale re-targeted Open64 compilers for x86-64 and seeded new repository to enable further development by open source compiler developers
- Thus, Open64 compiler technology ...
 - is gnu-compatible since it uses the gcc, g++ front-ends
 - incorporates back-end support for Itanium
 - is based on mature compiler technology developed by SGI with wide production exposure

Overview

- ✓ General Background
- Project Osprey Objectives
- Current Status & Project Plan

Project Osprey Strategic Objectives

- Significantly improve Linux/IPF GCC performance for the 2006-2007 timeframe
 - Reduce the gap between gcc on Itanium versus x86
 - Meet the needs of performance sensitive gcc users/apps
 - Facilitate advanced compiler research to complement Itanium roadmap
- Mate Open64 back-end with gcc front-end through persistent intermediate representation
 - Stay clear of GPL vs. BSD licensing & copyright ownership issues
 - Make it available as an alternative gcc back-end for IPF
- Establish & maintain healthy partnership with Open64 & gcc developer community
 - Help accelerate mainstream gcc performance engineering efforts
 - Provide a proof point for gcc developers on how the GCC infrastructure can be evolved to support Cross Module Link-Time Optimizations

Project Osprey – Non-goals

- Check-in Open64 back-end sources into gcc mainline sources
- Replacement of mainstream gcc in Linux/IPF distributions
- Osprey compiler support for x86-64 platform
 - Rely on Open64 development community for x86-64 Osprey compiler support

Project Osprey

2006 Engineering Goals



- Merge latest GCC4.x front-end sources into Open64
- Incorporate latest ORC Itanium Code Generator
- Implement IA64 C++ ABI compatibility
- Ensure gcc command option compatibility
- Incorporate recent Itanium research contributions in the areas of software pipelining, data prefetch for indirect pointer references
- Drive Compiler Validation Process for Quality
 - Create an Itanium-targeted branch off the latest GCC 4.x sources
 - Integrate in back-end components from Open64 compiler SUBVERSION repository
 - Form extended engineering team, by engaging key academic and research institutions

Overview

- ✓ General Background
- ✓ Project Osprey Objectives
- Current Status & Project Plan

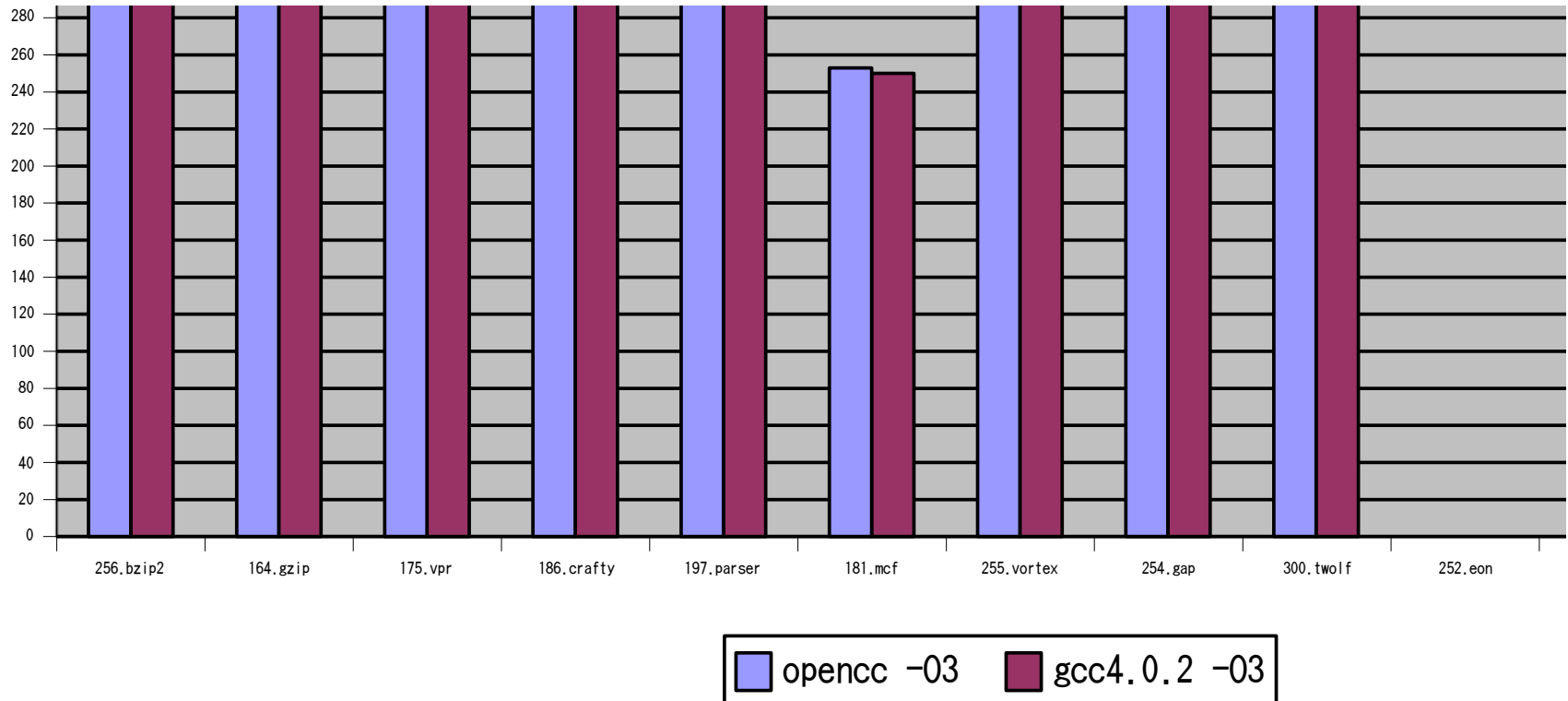
Current Activities

- *Tsinghua University*
 - *C++ ABI compatibility*
 - *Command Option compatibility*
 - *ASM statement support*
 - *Build Linux kernel to drive quality*
 - *Performance Analysis and Optimization Tuning on CERN loop*
- *ICT*
 - *F95 testing*
 - *SPEC2000fp performance analysis and optimization tuning*
- *University of Delaware*
 - *Hosts Osprey source repository and Bug tracking system*
 - *Source Merge with latest GCC front end*
 - *Optimization enhancements identified in Benchmark analysis*
- *HP*
 - *Autoconf*
 - *Quality Drive*

Key Project Milestones

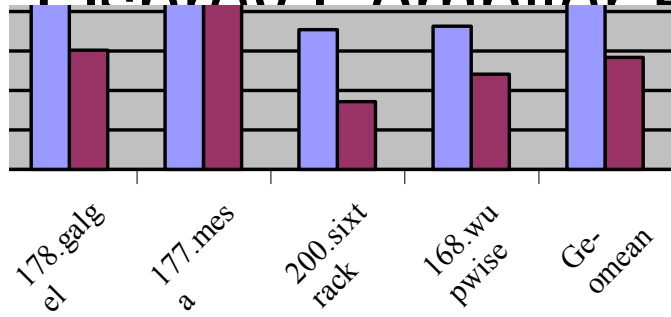
- *11/22/05: Subversion for Open64 main trunk*
- *12/20/05: Subversion with Pathscale merged*
- *3/18/06: Achieve first level C Robustness goal*
- 4/24/06: Gelato ICE project update
 - Preliminary performance assessment
- 5/30/06: Complete C/F95 quality goal
- 6/30/06: Achieve C++ Robustness goal
- 8/31/06: Achieve C++ Performance goal
- 9/30/06: First Osprey Compiler Release

Osprey Compiler Performance Status



- Osprey C compiler is 10% faster than gcc 4.0.2 on SPECint2000 benchmarks

Osprey Compiler Performance Status



- Osprey C/F95 compiler is 60% faster than gnu 4.0.2 on SPECfp2000 benchmarks

Call for Participation

- *Performance analysis of SPEC2006 benchmarks*
- *Performance analysis and tuning for Performance Sensitive Applications*
- *Automatic and Semi-automatic Parallelization based on Alternative Backend*
- *High Performance C/C++/F95 Open-MP with OpenMP library tuned for Itanium*
- *High Performance GCJ Compiler – hook up GCJ front end with the Open64 based Alternative Backend*



i n v e n t