Itanium Performance Insights from the IMPACT Compiler


IMPACT Research Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
http://www.crhc.uiuc.edu/IMPACT/
Itanium Development History

Intel and Hewlett Packard

- Intel/HP alliance input from UIUC
- Intel begins prototype compiler based on IMPACT
- Intel/HP announce Itanium
- Intel commercially licenses IMPACT
- Intel donates prototype compiler and system to IMPACT Research
- Intel/HP publically release Itanium processor & compilers

- ‘87 IMPACT ILP compiler born
- Research Partnership with HP Labs: predication & speculation
- Advanced control flow, predication & speculation balancing research
- Focused Itanium development begins at IMPACT Research
- General spec. support add to linux kernel
- Results comparable to Intel ecc achieved

University of Illinois at Urbana-Champaign

Itanium Performance Insights from the IMPACT Compiler
Itanium Architecture Overview

- Itanium design goal: enhance scalability of parallelism by moving complex decisions to the compiler
  - Bundling: enables static scheduling by communicating instruction parallelism
  - Predication: allows compiler to optimize across multiple paths by providing an alternative to control flow
  - Speculation support: allows compiler to select specific instructions for early execution

Original

Control Speculation

Predication
Itanium Compilation Landscape

• Increased reliance on the compiler for performance
  – Explicit control of the architecture: realities of modern microarchitecture have become visible at software level
  – Particular problems: effects of runtime uncertainty
    • Control resolution, variable memory latency, etc.
  – Solutions from EPIC/VLIW research
    • Memory disambiguation, profiling
    • Static scheduling, control speculation, predication

<table>
<thead>
<tr>
<th></th>
<th>Applicability</th>
<th>Public/Proprietary</th>
<th>Peephole opti level</th>
<th>ILP opti level</th>
<th>Extensibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcc</td>
<td>Very High</td>
<td>Public</td>
<td>Low</td>
<td>Very Low</td>
<td>Low</td>
</tr>
<tr>
<td>ecc</td>
<td>High</td>
<td>Proprietary</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>IMPACT</td>
<td>Medium</td>
<td>Future Public</td>
<td>Medium</td>
<td>Very High</td>
<td>Very High</td>
</tr>
</tbody>
</table>
IMPACT Compiler

- IMPACT compiler supports ILP compilation and research
  - Extensible framework for easy implementation of new optimizations
  - Versatile and pervasive intermediate representation (IR)
  - Comprehensive predicate-aware dataflow and predicate analyses
  - Direct analysis from IR at most stages of compilation
  - Advanced interprocedural pointer analysis
SPECcpu2000 Ratio Comparison

Dual processor HP i2000 Itanium 800 MHz with 2MB L3 cache
linux kernel 2.4.7 with PMC patch, gcc 2.9.6, ecc 5.0.1 beta
(ecc measurements not official Intel results)
Cycle Accounting Breakdown

Measured machine execution cycles using performance monitoring counters

Itanium Performance Insights from the IMPACT Compiler
Performance of Memory Hierarchy

![Diagram showing the performance of memory hierarchy with various cache levels and accessing latencies.](image-url)
Approaches to Control Speculation

General Speculation

- Inexpensive
- Moderate
- Expensive
- Result

Sentinel Speculation

- Write Value Into Register
- Write NaT into Register
- Except on Consumption

Diagram flow:
1. Id.s → Check DTLB → Walk VHPT → Check Page Table → Load Page
2. Id.s → Check DTLB → Walk VHPT → Write Value Into Register → Valid
3. Id.s → Check DTLB → Walk VHPT → Write Value Into Register → Not Valid
4. Id.s → Check DTLB → Write NaT into Register → Not Valid
5. Id.s → Check DTLB → Write NaT into Register → Valid
6. Id.s → Check DTLB → Write NaT into Register → Not Valid

Main Points:
- Write Value Into Register
- Write NaT into Register
- Except on Consumption
Architecture and OS Support

• IPF allows OS to mix and match general and sentinel speculation models

• Linux kernel modifications for general speculation
  – Basic support (system-wide general speculation): modify Default Control Register (DCR) not to defer transparent exceptions
  – Basic support caused page fault on every speculative NULL pointer access
  – Solution: install page 0 translation as a NaT page
  – Advanced support (selective general speculation): Bit in binary header indicates presence of recovery code. If no recovery code, ED bits in page table entries are cleared, overriding the DCR setting, so that no serviceable faults are deferred
Cycle Accounting Breakdown Revisited

Measured machine execution cycles using performance monitoring counters

Itanium Performance Insights from the IMPACT Compiler
Hyperblock Transformation

Example from *164.gzip* deflate()

- Commonly executed code pulled together into a straight-line sequence
- Serial branch execution replaced with parallel predicate evaluation
- Compromise between ILP and code size
  - Bundling reduces explicit horizontal NOPs
  - Interlocking reduces explicit vertical NOPs
Branch Effects of ILP Transformations

- ILP compilation eliminates branches
  - Predication and branch combining
  - Loop unrolling
- Reduction in total dynamic branches
- Proportionally fewer taken branches
- Reduction in mispredicted branches
Comparison of Branch Behavior

![Graph showing comparison of branch behavior]

- Not-taken incorrect
- Taken incorrect
- Taken correct
- Not-taken correct

Branches

gcc  ecc  impact

164  175  181  186  197  254  255  256  300
Challenges and Future Directions

• Profile dependence and accuracy
  – Getting profile into build systems
  – Transparent runtime data collection and re-optimization, making more use of the Itanium performance monitoring counters
• Effectiveness of ILP compilation algorithms
  – New algorithms to increase both aggressiveness and stability of speculation and predication
  – Program level ILP transformations
• Memory subsystem improvements
  – More efficient pointer analysis algorithms
  – Memory data flow analysis and optimization
• More results are being derived for presentation at Microprocessor Forum
Acknowledgements

• Former IMPACT members
  – David August, Ben-Chung Cheng, Daniel Connors, Kevin Crozier, Brian Deitrich, John Gyllenhaal, Richard Hank, Teresa Johnson, Dan Lavery, Scott Mahlke, Le-Chun Wu

• Intel Itanium Team
  – Carole Dulong, John Crawford, Dan Lavery, Steve Skedzielewski, Jim Pierce

• HP Itanium Team
  – Richard Holman, Vatsa Santhanam, Carol Thompson, Richard Hank

• HP Labs PD Team
  – Bob Rau, Mike Schlansker, Vinod Kathail, Scott Mahlke

• HP Labs Linux Team
  – Brian Lynn, David Mosberger, Hans Boehm, Stephane Eranian

• HP Philanthropy - 9 i2000 Itanium machines, expedited
  – Karen Fontana, Tony Napolitan, Ralph Hyver, Chris Hsiung, Rob Bouzon

• Intel - 2 alpha/beta Itaniums
  – Carole Dulong, John Crawford