CERN openlab/gelato Project

IPF compiler tests

Work done by Michal Kapalka (Summer Student 2003)
Stockholm, October 2003 – Sverre Jarp
IT Division – CERN

IPF C++ Compiler Status

S.Jarp

- **Intel's compilers:**
  - **Version 7.0 and 7.1 at production level**
  - **Version 8.0 beta**
    - **not covered in detail due to NDA**
- **GNU:**
  - **gcc 3.2, gcc 3.2.3, and 3.3 installed and tested**
  - **gcc 3.4 not yet ready**
- **ORC compiler (V2.0):**
  - **Not been able to guarantee correctness**
    - **With small test programs**
    - **Large benchmarks not tested**
- **Open Impact: No C++ support (yet)**

Processor hardware: Madison at 1.5 GHz

CERN Applications

S.Jarp
CERN

- **Very large C++ programs**
  - **Approximately 1 MLOC each**
  - **ROOT: Data Analysis framework**
    - **Also used by other sciences + fields such as financial analysis**
  - **GEANT4: Detector simulation program**
    - **Simulates the interaction of moving particles with matter**
    - **Also used in space exploration and life sciences**

Bench-1: ROOT tests/gcc

S.Jarp
CERN

Benchmark "stress" (Bigger is better)	O2	O3	O3, fast-math	O3, fast-math, IDML(*)
gcc 3.2	669	690	690	697
gcc 3.2.3	669	694	693	701 (105% of O2 result)
gcc 3.3	654	670	669	Failed at the time

- Best results with gcc 3.2.3
  - But results pretty equal
- gcc 3.3 has a drop in performance

inline-divide-minimum-latency

Bench-1: ROOT tests/ecc

S.Jarp
CERN

Benchmark "stress" (Bigger is better)	O2	O3	O2, IPO	O3, IPO	O2, ftz, IPO, PGO(*)
Intel 7.0	751	718	813	768	867
Intel 7.1	752	735	810	788	887 (118% of O2 result)
Intel 8.0 beta	Better	Better	Better	Better	Better

- **Good news with 8.0:**
  - **O3, IPO, PGO work together**
- **Intel compiler well ahead of gcc**
  - **Nice improvement with PGO**

prof_gen/prof_use + prof_dir

Bench-2: GEANT4 test40/gcc

S.Jarp
CERN

Real-time in seconds (Smaller is better)	O2	O3	O3, fast-math	O3, fast-math, IDML	O3, fast-math, IDML, PGO
gcc 3.2	142	141	126	121	118
gcc 3.2.3	143	142	128	122	119
gcc 3.3	140	138	124	119	116 (121%)

- **Best results with gcc 3.3**
  - **Nice improvements with higher optimisation:**
    - **PGO works because single directory is being used**

profile-arcs + branch-probabilities

Bench-2: GEANT4 test40/ecc

S.Jarp
CERN

Real-time in seconds (Smaller is better)	O2	O3	O2, IPO	O3, IPO	O2, ftz, PGO
Intel 7.0	144	128	147	132	Failed at the time
Intel 7.1	143	128	135	131	121 (118%)
Intel 8.0	Better	Better	No result at the time	No result at the time	Better

- Intel 7.0/7.1 behind gcc
- Intel 8.0 beta shows improvement
  - And has edged ahead

Some words on Intel v8.0 beta

S.Jarp
CERN

- **Certain features are very welcome:**
  - **IPO works when building libraries**
    - **Concerns all of CERN's large applications**
  - **Better inlining**
    - **Concrete example for CERN: certain critical methods**
    - **Mathematical functions: sine, cosine, log, exp, etc.**
  - **Smarter code generation**
    - **Better use of predication, large number of registers, etc.**
- **Will improve 7.1 results**

Some words on gcc

S.Jarp
CERN

- Seems to survive on IPF because of compiler improvements in general
  - For instance: Suse added new high-level loop optimizer and register scheduler for AMD-64
  - Note that starting-point is (for CERN) not as bad as what, for instance, NCAR sees
  - Interested in 3.4 and its new code generator by V.Makarov
- Still an issue over long-term gcc/IPF support

Compiler conclusions

- IPF compilers continue to improve
  - But, maybe I only mean Intel
- Definitely room for more support
  - gcc, orc, Open Impact (supporting C++)
- Enormous difference between:
  - "toy" benchmarks (20 KLOC) and real-world (1 MLOC)
- Self-help:
  - Since IPF is strict in-order execution
  - Relatively easy to check if the compiler does the right thing for you or not
    - But 1 MLOC C++ might keep us busy for some time