

3. März 2002

C/C++ Compiler Vergleich

Projekt für Vorlesung
37-235 Computer Systems Performance Analysis and Benchmarking
Prof. Thomas M. Stricker

Hatz Claudio

Inhaltverzeichnis

1	Einleitung	3
2	SPEC CPU2000	4
2.1	Ausführungsregeln	4
2.2	Metrics	4
3	Hardware	5
4	Parameter	6
4.1	Compiler	6
4.2	Optimierung	6
4.3	Betriebssystem	7
5	Testablauf	8
6	Auswertung	9
6.1	Wie wichtig sind die Parameter?	9
6.2	Optimierung	10
6.3	Compiler	12
7	Schlussfolgerungen	14
8	Referenzen	15
Anhang A:		
SPEC-Berichte, Konfigurationen, Resultate		16
Anhang B:		
Zwischenresultate		39

1. Einleitung

Performance spielt oft eine sehr wichtige Rolle bei der Software Entwicklung. Schnelle Algorithmen und effiziente Programmierung sind sicher sehr wichtig aber was für eine Auswirkung hat die Wahl eines Compilers?

Oft benutzt man einen Compiler nur weil er schon im Betriebssystem integriert ist (gcc in Linux) oder weil er mit einer Entwicklungsumgebung verbunden ist (Microsoft Visual Studio). Lohnt es sich in diesen Fällen einen anderen Compiler zu benutzen?

Wie wichtig ist die Wahl eines Compilers?

Liefern alle Compiler ähnlich schnellen Code oder gibt es Unterschiede?

Lohnt es sich Compiler-Flags zu setzen um den Code optimiert zu kompilieren?

Was für eine Rolle spielt das verwendete Betriebssystem dabei?

Ziel dieses Projekts ist es verschiedene C/C++ Compilers mittels der SPEC CPU2000 Benchmark-Paket [1] zu vergleichen und versuchen Antworten zu den oben gestellten Fragen zu finden.

2. SPEC CPU2000

Das SPEC CPU2000 besteht aus zwei Teilen: CINT2000 und CFP2000.

CFP2000 besteht vor allem aus Fließkomma-Programmen die in Fortran geschrieben sind. CINT2000 besteht vor allem aus Ganzzahlen-Programmen die in C oder C++ geschrieben sind. Da man in diesem Projekt C/C++ Compilers vergleichen will ist nur der CINT Teil verwendet worden.

CINT2000 besteht aus 12 einzelnen Benchmarks. Die Beschreibungen der einzelnen Benchmarks kann man auf [7] sehen. Von den 12 Benchmarks ist nur eon in C++ geschrieben, alle anderen sind in C.

Die meisten Benchmarks benutzen zwischen 100 und 200MB Hauptspeicher, sie sind also zu gross für die CPU Cache haben aber in der RAM noch Platz. Darum wird vor allem die CPU und die RAM getestet, die Harddisk wird nur sehr wenig benutzt. Angaben zu der Speicherbenutzung der einzelnen Benchmarks findet man in [8].

2.1 Ausführungsregeln

Man kann SPEC CPU2000 mit zwei Optimierungsstufen laufen lassen: base und peak. Bei base darf man maximal vier Compiler-Flags benutzen die für alle 12 Benchmarks gleich sein müssen, bei peak darf man jeden Benchmark einzeln optimieren mit so vielen Flags wie man will.

Um den Test zu starten muss man eine Konfigurations-Datei schreiben in der alle Parameter für die base und peak Kompilierung der einzelnen Benchmarks, die Beschreibung des getesteten Systems und die Kommentare stehen. Mit dieser Konfigurations-Datei wird ein Programm gestartet (runspec) das die einzelnen Benchmarks kompiliert, ausführt und ein Resultat-Bericht generiert.

SPEC schreibt eine ganze Serie von Regeln vor die erfüllt werden müssen, damit die Resultate als gültig bezeichnet werden können („Reportable Run“). Ist dies nicht der Fall, sind die Resultate nicht gültig („Invalid Run). Hier noch einige der Regeln:

- alle 12 Benchmarks müssen in einem einzigen Aufruf von runspec durchgeführt werden
- jeder Benchmark muss eine ungerade Anzahl mal durchgeführt werden (mindestens 3)
- es dürfen keine peak Resultate veröffentlicht werden ohne dass man im gleiche runspec Aufruf auch base Werte berechnet

2.2 Metrics

Jeder der 12 Benchmarks wird mindestens drei mal ausgeführt und die Ausführungszeit wird gemessen. Davon wird nur der Median benutzt und auf dem Resultat-Bericht eingetragen („Runtime“).

Für jeden Benchmark gibt es eine Referenz-Ausführungszeit („Reference Time“) die von einer Sun Ultra 10 stammt. Es wird das Verhältnis von dem Median der

Ausführungszeiten und der Referenz-Ausführungszeit berechnet, dieses Verhältnis ist die Anzahl Punkte die erreicht wird („Ratio“).

Das geometrische Mittel aus den Punkten der einzelnen Benchmarks ist der SPECint_base2000 (base) oder SPECint2000 (peak) Wert.

3. Hardware

Da in diesem Projekt nur die Resultate der verschiedenen Compiler verglichen werden sind alle Tests auf dem gleichen PC durchgeführt worden. Es ist anzunehmen, dass die verwendete Hardware keinen grossen Einfluss auf die unterschiede zwischen den Compilers hat; einzig der Intel Compiler könnte für einen Pentium 4 besseren Code generieren weil er der einzige ist der dafür spezielle Optimierungen verwendet.

Das Testsystem ist mit folgenden Komponenten aufgebaut:

- CPU : AMD Athlon 1.2GHz
- RAM : 512MB PC133 SDRAM CL2
- Mainboard : Shuttle AK12A mit VIA AK133A Chipset
- Harddisk : Maxtor 91536U6 14GB

Weitere Komponenten wie Grafikkarte, Netzwerkkarte, usw. haben keinen oder nur sehr geringen Einfluss auf die Resultate und werden deswegen hier nicht aufgelistet.

4. Parameter

Es gibt 3 Parameter die in diesem Projekt geändert werden:

- Compiler
- Optimierung
- Betriebssystem

4.1 Compiler

Die drei Compilers die getestet werden sind:

- *Intel C++ Compiler v5.0.1* [2] für Windows oder Linux
Kommerzieller C++ Compiler von Intel der auch Optimierungen für Pentium4 und 64-bit Code für Itanium Prozessoren generieren kann;
Von beiden Versionen (Windows/Linux) gibt es eine 30-Tage gratis Testversion, die Linux Version ist für den Privatgebrauch kostenlos.
- *Microsoft C/C++ Compiler v12.00* [3] nur für Windows
Kommerzieller C++ Compiler von Microsoft der in MS Visual C++ 6.0 enthalten ist.
- *GNU gcc v2.95.3* [4] für Linux oder Windows
Freeware C/C++ Compiler von GNU der unter der „General Public License“ [5] verteilt wird. Für Windows ist die portierte Version von cygwin [6] benutzt worden.

Der Intel Compiler wird in fast allen Tests verwendet die auf der SPEC Web-Seite publiziert sind. Es ist darum anzunehmen, dass dieser Compiler besonders gute Resultate liefert.

Der Microsoft Compiler wird unter Windows sehr oft benutzt, einfach weil er in MS Visual Studio standardmässig integriert ist. Es ist darum interessant zu sehen ob dieser Compiler auch leistungsfähig ist oder ob es sich lohnt einen anderen Compiler zu benutzen.

Der GNU Compiler ist vor allem in Linux/Unix sehr verbreitet auch weil er gratis benutzbar ist. Kann dieser gratis Compiler aber mit den kommerziellen Versionen mithalten?

4.2 Optimierung

Die Tests sind mit drei verschiedenen Optimierungen durchgeführt worden:

- *keine Optimierung*, der Compiler wird ohne Optimierungs-Flags aufgerufen

- *einfache Optimierung*, der Compiler wird mit zwei Flags aufgerufen: einer der eine generelle Optimierung aktiviert (-O3, /Ox, usw.) und einer der dem Compiler sagt für welche CPU er den Code generieren soll
- *2-Pass Optimierung* (nur Intel), der Code wird ein erstes mal kompiliert, mit Testdaten laufen gelassen um Informationen zu sammeln und dann mit diesen Zusatzinformationen ein zweites mal kompiliert. Zusätzlich werden das -ipo (multi-file Interprocedural Optimization) und -QaxK (generate code for streaming SIMD extensions) Flag aktiviert

Der erste Fall ist interessant, weil oft die Programme kompiliert werden ohne an die Optimierung zu denken, also ohne irgendein Flag zu setzen.

Die zweite Optimierung ist die, welche wahrscheinlich am häufigsten benutzt wird, wenn man ein Programm optimiert kompilieren will: man sagt einfach dem Compiler er soll das Programm optimieren ohne den Code und alle Compiler-Flags lang studieren und testen zu müssen.

Bei der 2-Pass Optimierung ist der Aufwand relativ gross. Man kompiliert das Programm ein erstes mal mit speziellen Flags, dann muss man das Programm mit Testdaten trainieren damit der Compiler Informationen sammeln kann. Zum Schluss wird das Programm mit den gesammelten Informationen optimiert und ein zweites mal kompiliert. Solche Testdaten zu finden ist nicht immer einfach und bei manchen Anwendungen auch nicht sinnvoll weil es keine „typische“ Daten für das Programm gibt. Bei SPEC CPU2000 werden die Testdaten mitgeliefert aber das kompilieren und trainieren dauert relativ lange.

Auf der SPEC Homepage sind die Resultate von einem Test, der AMD selbst mit einem sehr ähnlichen System durchgeführt hat. Das von AMD getestete System hat genau die gleiche CPU, das RAM ist gleich schnell aber nur 256MB gross. Da die Benchmarks nicht mehr als 200MB RAM benutzen sollte das unterschiedlich grosse RAM keinen zu grossen Unterschied machen. Die Resultate sollten also mit denen aus diesem Projekt vergleichbar sein.

In dem Test von AMD wird eine 2-Pass Optimierung verwendet, jeder Benchmark einzeln optimiert und es werden zum Teil spezielle Library's benutzt. Es ist anzunehmen, dass diese Resultate das Beste sind was man aus diesem System holen kann.

4.3 Betriebssystem

Die 2 Getesteten Betriebssysteme sind:

- *Windows 2000*
- *SuSE Linux 7.3 Kernel 2.4.10*

Es sind zwei verschiedene Betriebssysteme getestet worden, vor allem um jeden Compiler in „seinem“ System laufen zu lassen (GNU: Linux, Intel und MS: Windows) und um zu sehen ob die Portierung zu einem anderen Betriebssystem eine Verringerung der Performance mit sich bringt. Dieses Projekt soll nicht ein Vergleich zwischen den beiden Betriebssystemen sein.

Von beiden Betriebssystemen ist eine „Standard“ Installation gemacht worden ohne Updates, Service-Packs, usw. Die einzige Software die installiert worden ist sind die Compilers (gcc v2.95.3 war in Linux schon dabei) und das SPEC CPU2000 Benchmark-Paket.

Windows 2000 ist in einer NTFS Partition installiert worden, Linux in einer ext3 Partition.

5. Testablauf

Folgende Kombinationen sind getestet worden:

- | | | |
|---------------------|---------------------|----------------------|
| 1. Windows 2000, | Intel Compiler, | keine Optimierung |
| 2. Windows 2000, | Intel Compiler, | einfache Optimierung |
| 3. Windows 2000, | Intel Compiler, | 2-Pass Optimierung |
| 4. Windows 2000, | Microsoft Compiler, | keine Optimierung |
| 5. Windows 2000, | Microsoft Compiler, | einfache Optimierung |
| 6. Windows 2000, | GNU Compiler, | keine Optimierung |
| 7. Windows 2000, | GNU Compiler, | einfache Optimierung |
| 8. SuSE Linux 7.3, | Intel Compiler, | keine Optimierung |
| 9. SuSE Linux 7.3, | Intel Compiler, | einfache Optimierung |
| 10. SuSE Linux 7.3, | Intel Compiler, | 2-Pass Optimierung |
| 11. SuSE Linux 7.3, | GNU Compiler, | keine Optimierung |
| 12. SuSE Linux 7.3, | GNU Compiler, | einfache Optimierung |

Für jede Kombination sind 3 Durchläufe gemacht worden um (SPEC-)gültige Resultate zu erhalten.

Im Anhang A befinden sich die SPEC-Berichte und Konfigurationsdateien von allen durchgeführten Tests. In diesen Berichten findet man nur den Median aus den 3 Durchläufen, darum gibt es für jeden Test eine Tabelle in der man die einzelnen Werte der 3 Durchläufe sehen kann. Diese Werte kommen direkt aus den Log-Dateien die von runspec erzeugt werden.

Fast alle Werte stammen aus „reportable runs“. Ausnahmen sind die Resultate der 2-Pass Optimierung: die Resultate sind „Ungültig“ weil aus Zeitgründen nur die Peak-Werte berechnet worden sind.

Probleme gab es mit gcc unter Windows: es ist mir nicht gelungen 2 der 12 Benchmarks zu kompilieren (eon und pearlbmk) und ein Benchmark kompilierte ohne Fehlermeldungen stürzte aber während den Tests immer ab.

In einzelnen Fällen musste ich Flags setzen um die Floating-Point Präzision zu erhöhen oder gewisse Optimierungen auszuschalten damit die Benchmarks korrekt ausgeführt werden konnten.

Ein anderes Problem war die Zeit die nötig war um alles zum laufen zu bringen: eine komplette gültige Ausführung mit base und peak Werten dauert zwischen 10 und 15 Stunden.

6. Auswertung

Als erstes wird eine 2^3 Einflussgrößenanalyse durchgeführt um zu klären welchen Einfluss die 3 Parameter (Compiler, Optimierung und Betriebssystem) auf das Resultat haben.

Danach werden die wichtigsten Parameter noch einzeln analysiert.

6.1 Wie wichtig sind die drei Parameter?

Mit dieser Analyse kann man gut sehen wie gross der Effekt von jedem Parameter (oder Kombination der Parameter) auf das Resultat ist. Man kann also entscheiden welche Parameter relevant sind und welche nicht.

Ich habe eine Analyse mit einem $2^3 \times 3$ –Versuchsplan durchgeführt. Für jeden der 3 Parameter sind 2 Werte benutzt worden:

- Compiler: GNU (-1) und Intel (+1)
- Optimierung: keine (-1) und einfache (+1)
- Betriebssystem: Linux (-1) und Windows (+1)

Für die Analyse habe ich nur die 9 Benchmarks die mit allen Kombinationen aus diesen Parameter gelaufen sind (also nicht gcc, eon und pearlbmk). Jedes Experiment ist 3 mal wiederholt worden.

Die Analyse ist mit den Resultaten aus jedem der 9 Benchmarks und dem geometrischen Mittel durchgeführt worden.

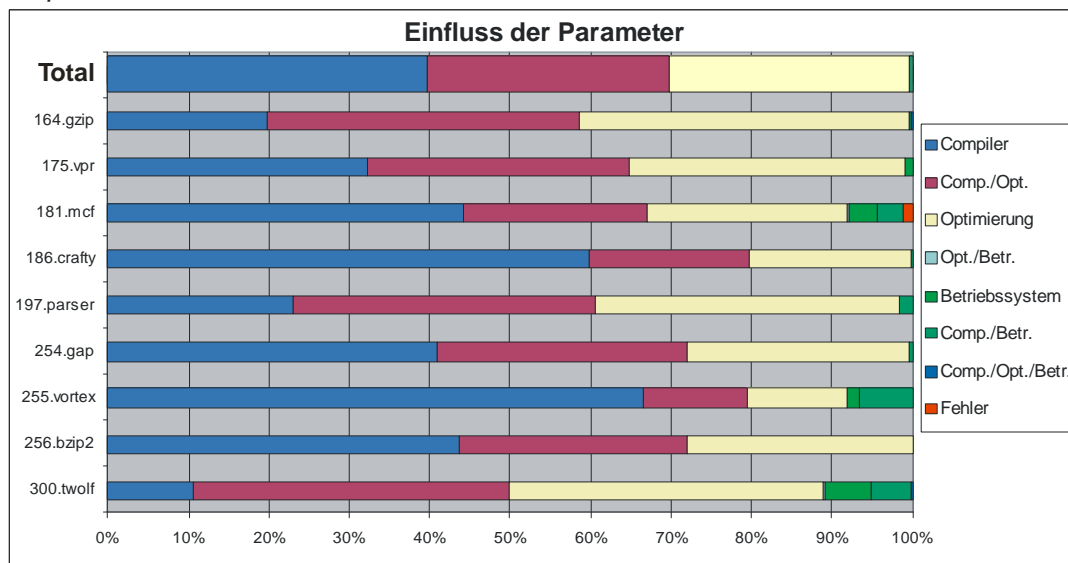
Im Anhang B findet man die Zwischenresultate und die Daten zu den Vertrauensintervallen.

Hier die wichtigsten Resultate:

Tabelle 1: Effekt der Einzelnen Parameter auf das Resultat

	Compiler	C/O	Optim.	O/B	Betriebss.	C/B	C/O/B	Fehler
Total	39.83%	29.98%	29.89%	0.01%	0.03%	0.26%	0.00%	0.00%
164.gzip	19.97%	38.56%	41.03%	0.05%	0.01%	0.18%	0.13%	0.08%
175.vpr	32.21%	32.54%	34.49%	0.00%	0.72%	0.02%	0.01%	0.02%
181.mcf	44.19%	22.79%	24.93%	0.27%	3.46%	3.17%	0.12%	1.08%
186.crafty	59.73%	20.04%	20.07%	0.00%	0.00%	0.16%	0.00%	0.00%
197.parser	23.00%	37.46%	37.87%	0.00%	0.01%	1.65%	0.00%	0.01%
254.gap	40.90%	31.05%	27.60%	0.08%	0.01%	0.33%	0.03%	0.00%
255.vortex	66.46%	13.00%	12.55%	0.00%	1.17%	6.79%	0.01%	0.02%
256.bzip2	43.64%	28.29%	28.00%	0.00%	0.02%	0.03%	0.00%	0.01%
300.twolf	10.52%	39.28%	39.20%	0.12%	5.86%	4.88%	0.10%	0.05%

Graph 1: Effekt der Einzelnen Parameter auf das Resultat



Man sieht sofort, dass der Compiler, die Optimierung und die Kombination aus den beiden eine sehr wichtige Rolle spielen: zusammen sind sie für über 99% des Resultats verantwortlich.

Das Betriebssystem hingegen hat auf das Gesamtergebnis praktisch keinen Einfluss. Nur in einzelnen Benchmarks (mcf, vortex und twolf) hat es eine kleine Einwirkung, die aber im Vergleich zu Compiler und Optimierung irrelevant bleibt.

Man sieht, dass der Fehler in den einzelnen Benchmarks eine sehr kleine Rolle spielt und in dem Gesamtergebnis praktisch gar keine. Das zeigt, dass 3 Wiederholungen der Experimente genügen um sehr genaue Ergebnisse zu bekommen.

Das sieht man auch bei der Berechnung des Vertrauensintervalls: auch mit 99.9% Sicherheit bleibt der Intervall sehr klein und hat keinen Einfluss auf die klare Dominanz von Compiler und Optimierung.

Da die Wahl des Compilers und die Optimierung eine wichtige Rolle spielen lohnt es sich, diese zwei Parameter in einer anderen Analyse genauer anzuschauen.

Der Effekt von dem Betriebssystem hat in den meisten Benchmarks die gleiche Größenordnung des Fehlers. Darum ist es irrelevant und es lohnt sich nicht diesen Parameter noch einzeln anzuschauen.

6.2 Optimierung

Ziel dieser Analyse ist es zu sehen wie gross die Performance Änderung ist wenn bei der Kompilierung Optimierungs-Flags gesetzt werden.

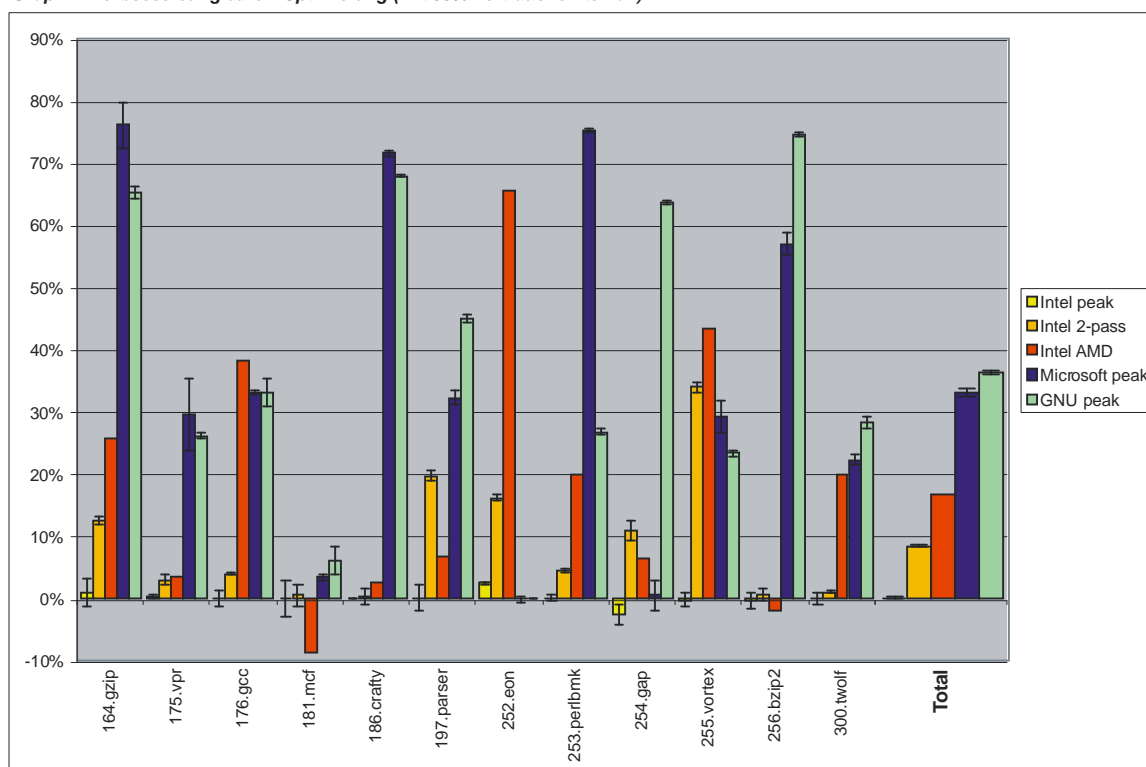
Um zu sehen wie gross die Änderungen sind habe ich bei jedem Compiler das Verhältnis zwischen dem Resultat mit und ohne Optimierung berechnet. Zum Vergleich sieht man auch die hoch optimierten Peak-Resultate aus den Tests von AMD.

In der folgenden Tabelle sieht man die Resultate zusammen mit den Vertrauensintervallen:

Tabelle 2: Verbesserung durch Optimierung (mit 99% Vertrauensintervall)

	Intel		AMD		Microsoft		GNU		
	peak	2-pass	peak	2-pass	peak	2-pass	peak	2-pass	
164.gzip	1.03%	±0.00%	12.47%	±0.00%	25.86%	76.11%	±0.00%	65.33%	±0.00%
175.vpr	0.22%	±0.00%	2.94%	±0.00%	3.40%	29.57%	±0.00%	26.18%	±0.00%
176.gcc	-0.04%	±0.00%	3.94%	±0.00%	38.37%	33.11%	±0.00%	32.98%	±0.00%
181.mcf	0.03%	±0.00%	0.49%	±0.00%	-8.73%	3.37%	±0.00%	6.21%	±0.00%
186.crafty	0.01%	±0.00%	0.37%	±0.00%	2.59%	71.66%	±0.00%	67.95%	±0.00%
197.parser	0.13%	±0.00%	19.63%	±0.00%	6.60%	32.18%	±0.00%	45.09%	±0.00%
252.eon	2.50%	±0.00%	16.18%	±0.00%	65.73%	-0.14%	±0.00%	-0.01%	±0.00%
253.perlbmk	0.09%	±0.00%	4.54%	±0.00%	19.75%	75.25%	±0.00%	26.82%	±0.00%
254.gap	-2.61%	±0.00%	10.90%	±0.00%	6.31%	0.51%	±0.00%	63.65%	±0.00%
255.vortex	-0.28%	±0.00%	33.91%	±0.00%	43.34%	29.14%	±0.00%	23.30%	±0.00%
256.bzip2	-0.25%	±0.00%	0.62%	±0.00%	-1.90%	57.06%	±0.00%	74.55%	±0.00%
300.twolf	-0.07%	±0.00%	1.00%	±0.00%	19.76%	22.28%	±0.00%	28.34%	±0.00%
Total	0.06%	±0.00%	8.50%	±0.00%	16.68%	33.16%	±0.00%	36.31%	±0.00%

Graph 2: Verbesserung durch Optimierung (mit 99% Vertrauensintervall)



Intel

Bei der einfachen Optimierung liegt das 0% in den meisten Fällen innerhalb des Vertrauensintervalls. Darum kann man nicht sagen, dass diese Optimierung ein Effekt auf die Resultate hat. Diese Optimierung bringt also praktisch nichts.

Interessanter sind die Auswirkungen der 2-Pass-Optimierung: die Resultate sind bei den einzelnen Benchmarks sehr unterschiedlich. Es gibt Benchmarks mit einer Verbesserung von über 30% (vortex) und andere in der auch diese Optimierung keine Wirkung hat.

Bei den Resultaten aus den Tests von AMD sieht man, dass eine gezielte Optimierung und die Verwendung von optimierten Library's noch einiges bringen

kann. Die Benchmarks, die am meisten Speicher benutzen (mcf, bzip2, ...) [8], haben zum Teil schlechtere Resultate bekommen weil das System von AMD nur 256MB statt 512MB RAM hat.

Microsoft und GNU

Diese zwei Compiler haben sehr ähnliche Resultate: bei beiden haben die Optimierungs-Flags eine sehr grosse Wirkung. Die Resultate sind mit der einfachen Optimierung meistens 20 bis 70% besser. Nur in einzelnen Fällen hat die Optimierung keine Einwirkung.

Diese Analyse zeigt, dass bei den Microsoft und GNU Compiler einfache Optimierungs-Flags sehr wichtig sind, da sie mit wenig Aufwand sehr viel bewirken. Beim Intel Compiler kann man einfache Optimierungs-Flags problemlos weglassen, für bessere Resultate muss man aber mehr Zeit investieren um eine 2-Pass Optimierung zu machen oder gezielt Flags zu setzen.

6.3 Compiler

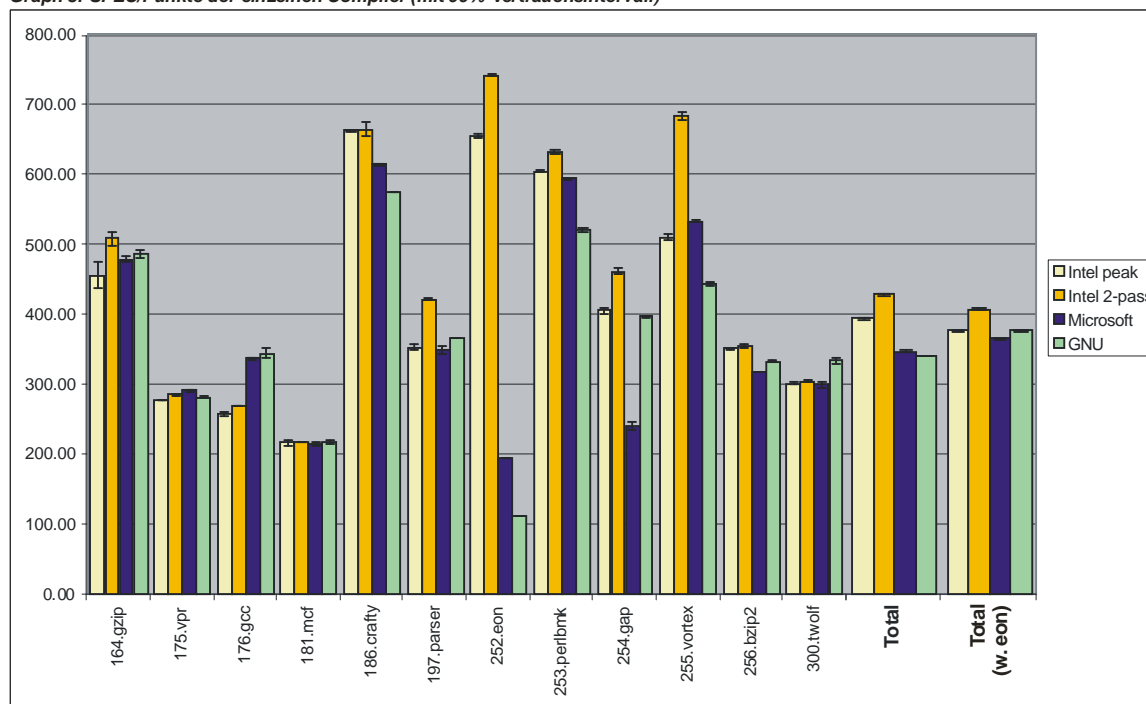
Zum Schluss noch der Vergleich zwischen den einzelnen Compilern.

In der folgenden Tabelle werden die Resultate, welche die drei Compiler mit der einfachen Optimierung und der Intel Compiler mit der 2-Pass Kompilierung erreicht haben, verglichen.

Tabelle 3: SPEC/Punkte der einzelnen Compiler (mit 99% Vertrauensintervall)

	Intel peak		Intel 2-pass		Microsoft		GNU	
164.gzip	455.01	±18.86	506.53	±9.07	477.47	±4.81	485.05	±5.28
175.vpr	276.15	±0.76	283.64	±1.78	289.36	±1.26	280.36	±0.39
176.gcc	256.59	±2.53	266.82	±0.51	335.17	±0.59	343.42	±7.36
181.mcf	215.40	±3.83	216.41	±0.31	214.15	±2.23	215.99	±1.69
186.crafty	661.08	±0.65	663.44	±9.25	612.84	±1.25	574.29	±0.19
197.parser	351.29	±5.01	419.70	±1.54	348.50	±5.17	364.49	±0.16
252.eon	653.74	±2.21	740.96	±2.32	193.70	±0.47	110.88	±0.03
253.perlbmk	603.00	±1.36	629.79	±2.72	592.70	±1.71	519.58	±3.30
254.gap	404.09	±3.91	460.13	±4.58	239.07	±5.13	395.34	±0.27
255.vortex	509.17	±3.85	683.74	±5.77	531.45	±1.80	442.15	±1.68
256.bzip2	349.50	±0.39	352.56	±2.92	316.74	±0.59	331.02	±1.62
300.twolf	300.10	±2.09	303.33	±0.96	297.97	±3.15	332.34	±3.85
Total	393.35	±1.37	426.54	±1.30	345.72	±1.63	339.43	±0.83
Total (without eon)	375.60	±1.31	405.65	±1.38	364.42	±1.81	375.78	±1.00

Graph 3: SPEC/Punkte der einzelnen Compiler (mit 99% Vertrauensintervall)



Der Intel Compiler hat beim geometrischen Mittel aus den 12 Benchmarks etwa 15% mehr Punkte als der Microsoft und den GNU Compiler und scheint somit der klare Sieger zu sein.

Wenn man aber die Resultate der Einzelnen Benchmarks besser anschaut merkt man dass der Intel Compiler in eon einen enormen Vorsprung auf die anderen beiden Compilers hat (~330% auf Microsoft und ~590% auf GNU). Der eon Benchmark ist der einzige der in C++ geschrieben ist, alle anderen sind in C geschrieben.

Berechnet man das geometrische Mittel aus den C Benchmarks (ohne eon) sind die Intel und GNU Compiler auf dem gleich Niveau, der Microsoft Compiler liegt noch ein wenig zurück.

7. Schlussfolgerungen

Die Schlussfolgerungen unterscheiden sich für C und C++ Programme.

Wenn man nur C Programme schreibt sind alle drei Compilers im Normalfall etwa gleich gut. Man kann also problemlos ein gratis Compiler benutzen (GNU) oder den Compiler, der schon in der Entwicklungsumgebung integriert ist (Microsoft). Wichtig ist bei den Microsoft und GNU Compilers, dass man einfache Optimierungs-Flags setzt da sie mit wenig Aufwand sehr viel bringen. Der Intel Compiler liefert auch ohne Flags schnellen Code.

Wenn es hingegen sehr wichtig ist ein Programm optimal zu kompilieren, kann es sich lohnen, eine 2-Pass Kompilierung mit dem Intel Compiler durchzuführen oder verschiedene Flags auszuprobieren um sie optimal setzen zu können.

Will man C++ Programme kompilieren, spielt die Wahl des Compilers eine entscheidende Rolle: mit dem Intel Compiler kann man 3 bis 6 mal schnellere Programme haben. Schon beim Wechsel vom dem GNU zu dem Microsoft Compiler werden die Programme fast doppelt so schnell.

Wenn man statt dem benutzten AMD Prozessor die Tests auf einem Pentium4 Prozessor durchgeführt hätte, würden sich die Resultate wahrscheinlich noch mehr zu Gunsten des Intel Compilers ändern da er der einzige ist, der den Code für den Pentium4 optimieren kann.

Der Intel Compiler ist also eine interessante Alternative zu den beiden „standard“ Compiler (Microsoft und GNU). Vor allem in Linux, wo der Intel Compiler für den nicht-kommerziellen gebrauch gratis ist, lohnt es sich diesen Compiler zu benutzen.

8. Referenzen

- [1] SPEC CPU2000 Homepage
<http://www.spec.org/osg/cpu2000/>
- [2] Intel Compiler Homepage
<http://developer.intel.com/software/products/compilers/>
- [3] Microsoft Visual Studio Homepage
<http://msdn.microsoft.com/vstudio/>
- [4] GNU GCC Homepage
<http://gcc.gnu.org/>
- [5] GPL Lizenz
<http://www.gnu.org/copyleft/gpl.html>
- [6] Cygwin Homepage
<http://www.cygwin.com/>
- [7] Beschreibung der einzelnen Benchmarks aus CINT2000
<http://www.spec.org/osg/cpu2000/CINT2000/>
- [8] Speicherbenutzungsprofile der Benchmarks
<http://www.spec.org/osg/cpu2000/analysis/memory/>

Anhang A:

SPEC-Berichte, Konfigurationen, Resultate

Windows 2000, Intel Compiler,	keine Optimierung	17
Windows 2000, Intel Compiler,	einfache Optimierung	17
Windows 2000, Intel Compiler,	2-Pass Optimierung	20
Windows 2000, Microsoft Compiler,	keine Optimierung	23
Windows 2000, Microsoft Compiler,	einfache Optimierung	23
Windows 2000, GNU Compiler,	keine Optimierung	26
Windows 2000, GNU Compiler,	einfache Optimierung	26
SuSE Linux 7.3, Intel Compiler,	keine Optimierung	29
SuSE Linux 7.3, Intel Compiler,	einfache Optimierung	29
SuSE Linux 7.3, Intel Compiler,	2-Pass Optimierung	32
SuSE Linux 7.3, GNU Compiler,	keine Optimierung	35
SuSE Linux 7.3, GNU Compiler,	einfache Optimierung	35
AMD Resultat		38

CINT2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel C++ Compiler v5.0.1
Windows 2000

SPECint2000 = 393
SPECint_base2000 = 393

SPEC license #: 925 Tested by: Hatz Claudio Test date: 7-8.2.2002 Hardware Avail: -- Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	310	452	306	458	
175.vpr	1400	508	276	507	276	
176.gcc	1100	428	257	429	256	
181.mcf	1800	838	215	838	215	
186.crafty	1000	151	661	151	661	
197.parser	1800	513	351	514	350	
252.eon	1300	204	638	199	654	
253.perlbmk	1800	299	603	298	603	
254.gap	1100	265	415	273	403	
255.vortex	1900	372	511	374	509	
256.bzip2	1500	429	350	429	349	
300.twolf	3000	999	300	999	300	

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: Windows 2000
Compiler: Intel C++ Compiler Version 5.0.1
File System: NTFS
System State: Default

Notes/Tuning Information

Base tuning: no flags
Peak tuning: /Ox /G6

Portability:
176.gcc /Op (enable better floating-point precision)

```
# Invocation command line:
# C:\spec2000\bin\runspec --reportable --tune all --config win_intel.cfg int
#####

#####
# Compiler selection

default=default=default=default:
CC      = icl
CXX     = icl

#####
# Portability flags

164.gzip=default=default=default:
CPORTABILITY = -DNO_UTIME -DNO_CHOWN

175.vpr=default=default=default:
CPORTABILITY = -Op

176.gcc=default=default=default:
CPORTABILITY = -Dalloca=_alloca -Op /F10000000
EXTRA_LDFLAGS = -Op /F10000000 /Fm

181.mcf=default=default=default:
CPORTABILITY = -D__STDC__

186.crafty=default=default=default:
CPORTABILITY = -DNT_i386

253.perlbnk=default=default=default:
EXTRA_CFLAGS = -DSPEC_CPU2000_NTOS -DPERLDLL /MT

254.gap=default=default=default:
CPORTABILITY = -DSYS_HAS_CALLOC_PROTO -DSYS_HAS_MALLOC_PROTO

#####
# Peak Tuning Flags

int=peak=default=default:

COPTIMIZE    = /Ox /G6
CXXOPTIMIZE  = /Ox /G6

#####
# Note information

int=default=default=default:
hw_vendor    = Intel C++ Compiler v5.0.1
hw_model     = Windows 2000
hw_cpu       = AMD Athlon 1.2 GHz
hw_cpu_mhz   = 1200
hw_fpu       = Integrated
hw_ncpu      = 1
hw_ncpuorder= 1
hw_pcache   = 64KBI + 64KBD on chip
hw_scache   = 256KB(I+D) on chip
hw_tcache   = N/A
hw_ocache   = N/A
hw_memory   = 512MB PC133 SDRAM CL2
hw_disk     = Maxtor 91536U6
hw_other    = Shuttel AK12A Mainboard
sw_os       = Windows 2000
sw_compiler = Intel C++ Compiler Version 5.0.1
sw_file     = NTFS
sw_state    = Default
license_num = 925
tester_name = Hatz Claudio
test_date   = 7-8.2.2002
```

393.15

keine Optimierung

Windows 2000

Intel C++ Compiler v5.0.1

	Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
		Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	1400	312.980	309.986	309.616	447	452	452	450	7.10	2.66	0.59%
175.vpr	1400	507.950	508.011	508.371	276	276	275	276	0.02	0.12	0.04%
176.gcc	1100	428.756	428.346	428.446	257	257	257	257	0.02	0.13	0.05%
181.mcf	1800	838.176	838.286	831.185	215	215	217	215	1.10	1.05	0.49%
186.crafty	1000	151.317	151.257	151.277	661	661	661	661	0.02	0.13	0.02%
197.parser	1800	514.419	512.217	512.557	350	351	351	351	0.66	0.81	0.23%
252.eon	1300	204.023	203.793	203.683	637	638	638	638	0.29	0.54	0.09%
253.perlbnk	1800	299.381	298.529	298.459	601	603	603	602	1.07	1.03	0.17%
254.gap	1100	265.171	264.480	265.702	415	416	414	415	0.92	0.96	0.23%
255.vortex	1900	372.555	371.915	371.905	510	511	511	511	0.26	0.51	0.10%
256.bz2	1500	429.027	426.343	428.977	350	352	350	350	1.59	1.26	0.36%
300.twolf	3000	997.975	998.937	999.867	301	300	300	300	0.08	0.28	0.09%
Geo.mean		392.53	393.47	393.37	393.15	393.15	393.12	393.12	0.27	0.52	0.13%
Geo.mean (without eon)		375.62	376.56	376.44	376.23	376.23	376.20	376.20	0.26	0.51	0.14%
Geo.mean (without gcc, eon, perlbnk)		371.91	372.89	372.75	372.50	372.50	372.52	372.52	0.28	0.53	0.14%

393.29

einfache Optimierung

Windows 2000

Intel C++ Compiler v5.0.1

	Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
		Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	1400	312.099	305.219	305.830	449	459	458	455	31.27	5.59	1.23%
175.vpr	1400	507.430	506.618	506.889	276	276	276	276	0.05	0.23	0.08%
176.gcc	1100	427.275	429.217	429.608	257	256	256	257	0.56	0.75	0.29%
181.mcf	1800	838.235	830.615	838.206	215	217	215	215	1.29	1.14	0.53%
186.crafty	1000	151.318	151.237	151.247	661	661	661	661	0.04	0.19	0.03%
197.parser	1800	509.903	513.688	513.598	353	350	350	351	2.20	1.48	0.42%
252.eon	1300	199.086	198.726	198.756	653	654	654	654	0.43	0.66	0.10%
253.perlbnk	1800	298.739	298.399	298.389	603	603	603	603	0.16	0.40	0.07%
254.gap	1100	272.682	272.662	271.320	403	403	405	404	1.35	1.16	0.29%
255.vortex	1900	372.195	373.657	373.627	510	508	509	509	1.30	1.14	0.22%
256.bz2	1500	429.297	429.247	429.027	349	349	350	349	0.01	0.12	0.03%
300.twolf	3000	1001.97	999.006	997.995	299	300	301	300	0.38	0.62	0.21%
Geo.mean		392.89	393.67	393.48	393.29	393.29	393.35	393.35	0.16	0.41	0.10%
Geo.mean (without eon)		375.16	375.91	375.71	375.52	375.52	375.60	375.60	0.15	0.39	0.10%
Geo.mean (without gcc, eon, perlbnk)		371.13	372.18	371.97	371.70	371.70	371.76	371.76	0.31	0.56	0.15%

CINT2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel C++ Compiler v5.0.1
Windows 2000

SPECint2000 = 426
SPECint_base2000 = 0.00

SPEC license #: 925 | Tested by: Hatz Claudio | Test date: 8.2.2002 | Hardware Avail: -- | Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio		200	400	600	800
164.gzip				276	507					
175.vpr				494	283					
176.gcc				412	267					
181.mcf				832	216					
186.crafty				151	662					
197.parser				429	419					
252.eon				175	741					
253.perlbmk				286	630					
254.gap				239	459					
255.vortex				278	683					
256.bzip2				425	353					
300.twolf				989	303					

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: Windows 2000
Compiler: Intel C++ Compiler Version 5.0.1
File System: NTFS
System State: Default

Notes/Tuning Information

Peak tuning: 2-Pass
-QaxK -Qipo

Portability:
176.gcc /Op (enable better floating-point precision)

Errors

'reportable' flag not set during run
No 'base' runs! Base measurement required!

```
# Invocation command line:

# C:\spec2000\bin\runspec --tune peak --config win_intel_2pass.cfg int
#####

#####
# Compiler selection

default=default=default=default:
CC      = icl
CXX     = icl

#####
# Portability flags

164.gzip=default=default=default:
CPORTABILITY = -DNO_UTIME -DNO_CHOWN

175.vpr=default=default=default:
CPORTABILITY = -Op

176.gcc=default=default=default:
CPORTABILITY = -Dalloca=_alloca -Op /F10000000
EXTRA_LDFLAGS = -Op /F10000000 /Fm

181.mcf=default=default=default:
CPORTABILITY = -D__STDC__

186.crafty=default=default=default:
CPORTABILITY = -DNT_i386

253.perlbnk=default=default=default:
EXTRA_CFLAGS = -DSPEC_CPU2000_NTOS -DPERLDLL /MT

254.gap=default=default=default:
CPORTABILITY = -DSYS_HAS_CALLOC_PROTO -DSYS_HAS_MALLOC_PROTO

#####
# Peak Tuning Flags

int=peak=default=default:

PASS1_CFLAGS= -Qprof_gen
PASS2_CFLAGS= -QaxK -Qipo -Qprof_use
PASS1_CXXFLAGS= -Qprof_gen
PASS2_CXXFLAGS= -QaxK -Qipo -Qprof_use
PASS1_LDFLAGS= -Qprof_gen
PASS2_LDFLAGS= -QaxK -Qipo -Qprof_use

#####
# Note information

int=default=default=default:
hw_vendor   = Intel C++ Compiler v5.0.1
hw_model    = Windows 2000
hw_cpu      = AMD Athlon 1.2 GHz
hw_cpu_mhz  = 1200
hw_fpu      = Integrated
hw_ncpu     = 1
hw_ncpuorder= 1
hw_pcache  = 64KBI + 64KBD on chip
hw_scache   = 256KB(I+D) on chip
hw_tcache   = N/A
hw_ocache   = N/A
hw_memory   = 512MB PC133 SDRAM CL2
hw_disk     = Maxtor 91536U6
hw_other    = Shuttel AK12A Mainboard
sw_os       = Windows 2000
sw_compiler = Intel C++ Compiler Version 5.0.1
sw_file     = NTFS
sw_state    = Default
license_num = 925
tester_name = Hatz Claudio
test_date   = 8.2.2002
```

Intel C++ Compiler v5.0.1 Windows 2000 2-Pass 426.47

	Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
		Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	1400	278.060	275.266	275.866	503	509	507	507	7.23	2.69	0.53%
175.vpr	1400	493.950	494.281	492.548	283	283	284	284	0.28	0.53	0.19%
176.gcc	1100	412.333	412.463	412.013	267	267	267	267	0.02	0.15	0.06%
181.mcf	1800	831.996	831.355	831.947	216	217	216	216	0.01	0.09	0.04%
186.crafty	1000	151.148	150.016	151.027	662	667	662	662	7.52	2.74	0.41%
197.parser	1800	429.197	429.097	428.346	419	419	420	420	0.21	0.46	0.11%
252.eon	1300	175.393	175.633	175.322	741	740	741	741	0.47	0.69	0.09%
253.perlbmk	1800	286.172	285.811	285.440	629	630	631	630	0.65	0.81	0.13%
254.gap	1100	238.253	239.524	239.414	462	459	459	460	1.84	1.36	0.30%
255.vortex	1900	278.471	278.069	277.119	682	683	686	684	2.92	1.71	0.25%
256.bz2	1500	426.664	424.770	424.941	352	353	353	353	0.75	0.87	0.25%
300.twolf	3000	988.391	988.612	990.093	304	303	303	303	0.08	0.28	0.09%
Geo.mean					426.09	426.73	426.79	426.47	0.15	0.39	0.09%
Geo.mean (without eon)					405.18	405.89	405.89	405.57	0.17	0.41	0.10%
Geo.mean (without gcc, eon, pearlbnk)					404.20	405.02	404.91	404.62	0.20	0.45	0.11%

CINT2000 Result

© Copyright 1999, Standard Performance Evaluation Corporation

Microsoft C/C++ Compiler Version 12.00
Windows 2000

SPECint2000 = 346
SPECint_base2000 = 260

SPEC license #: 925 Tested by: Hatz Claudio Test date: 17-18.1.2002 Hardware Avail: -- Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	518	270	293	478	
175.vpr	1400	623	225	484	289	
176.gcc	1100	437	252	328	335	
181.mcf	1800	869	207	840	214	
186.crafty	1000	280	357	163	613	
197.parser	1800	682	264	517	348	
252.eon	1300	670	194	671	194	
253.perlbnk	1800	532	338	304	593	
254.gap	1100	463	238	461	239	
255.vortex	1900	463	411	357	532	
256.bzip2	1500	744	202	474	317	
300.twolf	3000	1231	244	1008	298	

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: Windows 2000
Compiler: Microsoft C/C++ Compiler Version 12.00.8168
File System: NTFS
System State: Default

Notes/Tuning Information

Base tuning: no flags
Peak tuning: /Ox /G6

Portability:

176.gcc /Op (enable better floating-point precision)
254.gap peak /Oityb2 (/Ox without global optimization)
252.eon peak /Oityb2 (/Ox without global optimization)

```
# Invocation command line:

# C:\spec2000\bin\runspec --reportable --tune all --config win_ms.cfg int
#####

#####
# Compiler selection

default=default=default=default:
CC      = cl -Bd
CXX     = cl -Bd

#####
# Portability flags

176.gcc=default=default=default:
EXTRA_CFLAGS = -Dalloca=_alloca -Op
EXTRA_LDFLAGS = -F10000000

178.galgel=default=default=default:
EXTRA_FFLAGS = -fixed
LDOPT = -Fe$@ -link -stack:300000000
#OBJOPT= -Fo$@

186.crafty=default=default=default:
EXTRA_CFLAGS = -DNT_i386

252.eon=default=default=default:
SOURCE_PREFIX_CXX = -Tp

253.perlbnk=default=default=default:
EXTRA_CFLAGS = -DSPEC_CPU2000_NTOS -DPERLDLL /MT

254.gap=default=default=default:
EXTRA_CFLAGS=-DSYS_HAS_MALLOC_PROTO -DSYS_HAS_CALLOC_PROTO

#####
# Peak Tuning Flags

int=peak=default=default:
COPTIMIZE   = /Ox /G6
CXXOPTIMIZE = /Ox /G6

254.gap=default=default=default:
COPTIMIZE   = /Oityb2 /G6

252.eon=default=default=default:
CXXOPTIMIZE = /Oitib2 /G6

#####
# Note information

int=default=default=default:
hw_vendor   = Microsoft C/C++ Compiler Version 12.00
hw_model    = Windows 2000
hw_cpu      = AMD Athlon 1.2 GHz
hw_cpu_mhz  = 1200
hw_fpu      = Integrated
hw_ncpu     = 1
hw_ncpuorder= 1
hw_pcache   = 64KBI + 64KBD on chip
hw_scache   = 256KB(I+D) on chip
hw_tcache   = N/A
hw_ocache   = N/A
hw_memory   = 512MB PC133 SDRAM CL2
hw_disk     = Maxtor 91536U6
hw_other    = Shuttel AK12A Mainboard
sw_os       = Windows 2000
sw_compiler = Microsoft C/C++ Compiler Version 12.00.8168
sw_file     = NTFS
sw_state    = Default
tester_name = Hatz Claudio
test_date   = 17-18.1.2002
```


Microsoft C/C++ v12.00 Windows 2000 keine Optimierung 259.73

	Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
		Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	1400	519.517	517.745	511.906	269	270	273	270	4.40	2.10	0.77%
175.vpr	1400	636.235	622.866	621.634	220	225	225	225	8.20	2.86	1.28%
176.gcc	1100	437.499	436.488	436.557	251	252	252	252	0.11	0.33	0.13%
181.mcf	1800	865.124	872.355	869.200	208	206	207	207	0.75	0.86	0.42%
186.crafty	1000	280.193	280.083	280.053	357	357	357	357	0.01	0.09	0.03%
197.parser	1800	685.316	681.230	681.610	263	264	264	264	0.76	0.87	0.33%
252.eon	1300	671.856	669.282	669.503	193	194	194	194	0.17	0.41	0.21%
253.perlbmk	1800	532.476	532.095	532.146	338	338	338	338	0.02	0.13	0.04%
254.gap	1100	461.994	462.716	462.675	238	238	238	238	0.04	0.21	0.09%
255.vortex	1900	463.947	458.329	462.816	410	415	411	411	7.06	2.66	0.65%
256.bz2	1500	746.383	741.406	743.529	201	202	202	202	0.46	0.68	0.34%
300.twolf	3000	1233.50	1229.26	1230.75	243	244	244	244	0.18	0.43	0.17%
Geo.mean					258.89	259.98	260.02	259.73	0.41	0.64	0.25%
Geo.mean (without eon)					265.83	266.96	267.02	266.69	0.45	0.67	0.25%
Geo.mean (without gcc, eon, perlbnk)					260.44	261.70	261.78	261.39	0.57	0.75	0.29%

Microsoft C/C++ v12.00 Windows 2000 einfache Optimierung 345.74

	Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
		Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	1400	294.223	292.661	292.751	476	478	478	477	2.03	1.43	0.30%
175.vpr	1400	483.855	483.195	484.447	289	290	289	289	0.14	0.37	0.13%
176.gcc	1100	328.372	328.032	328.182	335	335	335	335	0.03	0.17	0.05%
181.mcf	1800	838.345	843.393	839.837	215	213	214	214	0.44	0.66	0.31%
186.crafty	1000	163.095	163.145	163.285	613	613	612	613	0.14	0.37	0.06%
197.parser	1800	518.565	514.069	516.874	347	350	348	349	2.35	1.53	0.44%
252.eon	1300	671.696	670.845	670.864	194	194	194	194	0.02	0.14	0.07%
253.perlbmk	1800	303.998	303.546	303.547	592	593	593	593	0.26	0.51	0.09%
254.gap	1100	462.765	456.987	460.623	238	241	239	239	2.31	1.52	0.64%
255.vortex	1900	357.925	357.354	357.264	531	532	532	531	0.28	0.53	0.10%
256.bz2	1500	473.601	473.290	473.812	317	317	317	317	0.03	0.18	0.06%
300.twolf	3000	1008.90	1003.18	1008.34	297	299	298	298	0.87	0.93	0.31%
Geo.mean					345.27	346.23	345.67	345.74	0.23	0.48	0.14%
Geo.mean (without eon)					363.92	364.99	364.35	364.42	0.29	0.54	0.15%
Geo.mean (without gcc, eon, perlbnk)					347.95	349.10	348.37	348.46	0.34	0.58	0.17%

CINT2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

GNU gcc version 2.95.3
Windows 2000

SPECint2000 = --
SPECint_base2000 = --

SPEC license #: 925 Tested by: Hatz Claudio Test date: 13.2.2002 Hardware Avail: -- Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	475	295	298	470	
175.vpr	1400	643	218	505	277	
176.gcc	1100	X	X	X	X	
181.mcf	1800	900	200	853	211	
186.crafty	1000	305	328	177	566	
197.parser	1800	751	240	513	351	
252.eon	1300	X	X	X	X	
253.perlbmk	1800	X	X	X	X	
254.gap	1100	466	236	284	387	
255.vortex	1900	560	339	448	424	
256.bzip2	1500	790	190	454	331	
300.twolf	3000	1147	262	917	327	

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: Windows 2000
Compiler: GNU gcc version 2.95.3-5 (cygwin)
File System: NTFS
System State: Default

Notes/Tuning Information

Base tuning: no flags
Peak tuning: -O3 -march=i686

Errors

'reportable' flag not set during run
253.perlbmk base did not have enough runs!
176.gcc base did not have enough runs!
252.eon base did not have enough runs!
253.perlbmk peak did not have enough runs!
176.gcc peak did not have enough runs!
252.eon peak did not have enough runs!

```
# Invocation command line:

# C:\spec2000\bin\runspec --tune all --config win_gcc.cfg int ^gcc ^eon ^perlbnk
#####

#####
# Compiler selection

default=default=default=default:
CC = gcc
CXX = g++
OBJ=.o
OBJOPT = -c -o $@
LDOPT = -o $@

#####
# Portability flags

175.vpr=default=default=default:
CPORTABILITY = -DSPEC_CPU2000

186.crafty=default=default=default:
EXTRA_CFLAGS=-DLINUX_i386

254.gap=default=default=default:
EXTRA_CFLAGS=-DSYS_IS_USG -DSYS_HAS_IOCTL_PROTO -DSYS_HAS_TIME_PROTO -DSYS_HAS_SIGNAL_PROTO -
DSYS_HAS_ANSI -DSYS_HAS_CALLOC_PROTO

#####
# Peak Tuning Flags

int=peak=default=default:
COPTIMIZE = -O3 -march=i686

#####
# Note information

int=default=default=default:
hw_vendor = GNU gcc version 2.95.3
hw_model = Windows 2000
hw_cpu = AMD Athlon 1.2 GHz
hw_cpu_mhz = 1200
hw_fpu = Integrated
hw_ncpu = 1
hw_ncpuorder= 1
hw_pcache = 64KBI + 64KBD on chip
hw_scache = 256KB(I+D) on chip
hw_tcache = N/A
hw_ocache = N/A
hw_memory = 512MB PC133 SDRAM CL2
hw_disk = Maxtor 91536U6
hw_other = Shuttel AK12A Mainboard
sw_os = Windows 2000
sw_compiler = GNU gcc version 2.95.3-5 (cygwin)
sw_file = NTFS
sw_state = Default
license_num = 925
tester_name = Hatz Claudio
test_date = 13.2.2002
```

GNU gcc v2.95.3 (Cygwin) Windows 2000 keine Optimierung 243.87

Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
	Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	476.586	475.043	474.983	294	295	295	294	0.32	0.56	0.19%
175.vpr	641.112	643.245	642.784	218	218	218	218	0.15	0.38	0.17%
176.gcc										
181.mcf	900.495	898.793	899.874	200	200	200	200	0.04	0.19	0.10%
186.crafty	302.414	305.139	305.048	331	328	328	329	2.81	1.68	0.51%
197.parser	750.119	752.512	750.880	240	239	240	240	0.15	0.39	0.16%
252.eon	1300									
253.perlbmk	1800									
254.gap	465.970	465.740	464.998	236	236	237	236	0.07	0.26	0.11%
255.vortex	560.857	560.476	558.824	339	339	340	339	0.43	0.66	0.19%
256.bzip2	793.310	788.204	789.616	189	190	190	190	0.40	0.63	0.33%
300.twolf	1146.96	1147.07	1148.38	262	262	261	261	0.03	0.18	0.07%
Geo.mean				251.49	251.41	243.94	243.91	56.81	7.54	3.09%
Geo.mean (without eon)				251.49	251.41	262.56	262.48	122.73	11.08	4.22%
Geo.mean (without gcc, eon, perlbmk)				251.49	251.41	251.51	251.47	0.00	0.05	0.02%

GNU gcc v2.95.3 (Cygwin) Windows 2000 einfache Optimierung 332.1

Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
	Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	300.102	297.808	297.598	467	470	470	469	4.74	2.18	0.46%
175.vpr	503.514	505.307	504.646	278	277	277	278	0.25	0.50	0.18%
176.gcc										
181.mcf	852.406	852.505	858.615	211	211	210	211	0.76	0.87	0.42%
186.crafty	177.015	175.693	176.744	565	569	566	567	5.05	2.25	0.40%
197.parser	512.798	512.537	512.437	351	351	351	351	0.02	0.13	0.04%
252.eon	1300									
253.perlbmk	1800									
254.gap	284.149	284.359	284.540	387	387	387	387	0.07	0.27	0.07%
255.vortex	448.184	447.804	444.319	424	424	428	425	4.13	2.03	0.48%
256.bzip2	456.367	453.563	453.653	329	331	331	330	1.33	1.15	0.35%
300.twolf	916.538	917.940	913.894	327	327	328	327	0.54	0.73	0.22%
Geo.mean				357.71	358.38	332.21	332.10	676.13	26.00	7.83%
Geo.mean (without eon)				357.71	358.38	367.79	367.67	91.35	9.56	2.60%
Geo.mean (without gcc, eon, perlbmk)				357.71	358.38	358.40	358.16	0.15	0.39	0.11%





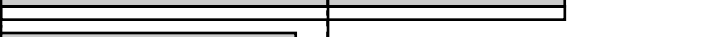

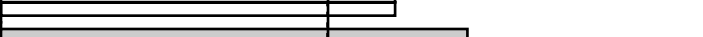


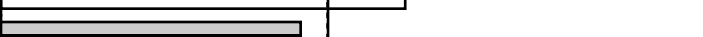

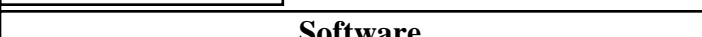
CINT2000 Result

Copyright ©1999-2001, Standard Performance Evaluation Corporation

Intel C++ Compiler v5.0.1
SuSE Linux 7.3

SPECint2000 = 377
SPECint_base2000 = 377

SPEC license #00925 Tested by: Hatz Claudio Test date: 6.2.2002 Hardware Avail: -- Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	312	448	312	449	
175.vpr	1400	499	281	497	282	
176.gcc	1100	413	266	412	267	
181.mcf	1800	835	216	835	216	
186.crafty	1000	154	650	154	651	
197.parser	1800	529	340	529	340	
252.eon	1300	286	455	286	455	
253.perlbmk	1800	334	538	334	539	
254.gap	1100	275	400	274	401	
255.vortex	1900	408	466	407	466	
256.bzip2	1500	434	346	434	345	
300.twolf	3000	921	326	919	326	

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: SuSE Linux 7.3
Compiler: Intel C++ Compiler v5.0.1
File System: ext3
System State: Default

Notes/Tuning Information

Base tuning: no flags
Peak tuning: -O3 -tpp6

Portability:
176.gcc -mp (maintain floating-point precision)

```
# Invocation command line:

# /home/claudio/spec2000/bin/runspec --reportable --tune=all --config=linux_intel.cfg int
#####

#####
# Compiler selection

default=default=default=default:
CC      = icc
CXX     = icc

#####
# Portability flags

176.gcc=default=default=default:
COPTIMIZE = -mp

186.crafty=default=default=default:

EXTRA_CFLAGS = -DLINUX_i386

252.eon=default=default=default:
EXTRA_CXXFLAGS=-DHAS_ERRLIST -DFMAX_IS_DOUBLE

253.perlbmk=default=default=default:
EXTRA_CFLAGS=-DSPEC_CPU2000_LINUX_I386 -DSPEC_CPU2000_NEED_BOOL -DSPEC_CPU2000_GLIBC22

254.gap=default=default=default:
EXTRA_CFLAGS=-DSYS_IS_USG -DSYS_HAS_IOCTL_PROTO -DSYS_HAS_TIME_PROTO -DSYS_HAS_SIGNAL_PROTO -
DSYS_HAS_ANSI -DSYS_HAS_CALLOC_PROTO

#####
# Peak Tuning Flags

int=peak=default=default:
COPTIMIZE = -O3 -tpp6
CXXOPTIMIZE = -O3 -tpp6

#####
# Note information

int=default=default=default:
hw_vendor   = Intel C++ Compiler v5.0.1
hw_model    = SuSE Linux 7.3
hw_cpu      = AMD Athlon 1.2 GHz
hw_cpu_mhz  = 1200
hw_fpu      = Integrated
hw_ncpu     = 1
hw_ncpuorder= 1
hw_pcache  = 64KBI + 64KBD on chip
hw_scache  = 256KB(I+D) on chip
hw_tcache  = N/A
hw_ocache  = N/A
hw_memory  = 512MB PC133 SDRAM CL2
hw_disk    = Maxtor 91536U6
hw_other   = Shuttel AK12A Mainboard
sw_os      = SuSE Linux 7.3
sw_compiler = Intel C++ Compiler v5.0.1
sw_file    = ext3
sw_state   = Default
license_num = 000925
tester_name = Hatz Claudio
test_date  = 6.2.2002
```

Intel C++ Compiler v5.0.1 SuSE Linux 7.3 keine Optimierung 376.82

Reference	Runtime				Ratio				Mean	Variance	Std. Dev.	C.O.V.
	Run1	Run2	Run3	Run1	Run2	Run3	Run1	Run2				
164.gzip	1400	314.164	312.253	312.227	446	448	448	448	447	2.51	1.59	0.35%
175.vpr	1400	500.920	498.326	498.984	279	281	281	281	280	0.57	0.76	0.27%
176.gcc	1100	412.962	413.237	410.582	266	266	268	266	267	0.90	0.95	0.35%
181.mcf	1800	838.866	834.995	835.127	215	216	216	216	215	0.32	0.56	0.26%
186.crafty	1000	153.779	153.675	153.738	650	651	650	650	650	0.05	0.22	0.03%
197.parser	1800	529.323	529.320	529.873	340	340	340	340	340	0.04	0.20	0.06%
252.eon	1300	285.935	285.921	285.956	455	455	455	455	455	0.00	0.03	0.01%
253.perlbmk	1800	335.367	334.394	334.459	537	538	538	538	538	0.76	0.87	0.16%
254.gap	1100	275.629	275.314	275.336	399	400	400	400	399	0.06	0.25	0.06%
255.vortex	1900	408.090	407.619	406.971	466	466	467	466	466	0.41	0.64	0.14%
256.bz2	1500	435.484	432.996	433.821	344	346	346	346	346	1.02	1.01	0.29%
300.twolf	3000	918.364	921.321	921.969	327	326	325	326	326	0.46	0.68	0.21%
Geo.mean		376.18	376.93	376.99	376.82	376.70	376.82	376.82	376.70	0.20	0.45	0.12%
Geo.mean (without eon)		369.76	370.55	370.63	370.44	370.31	370.44	370.44	370.31	0.23	0.48	0.13%
Geo.mean (without gcc, eon, perlbnk)		367.93	368.80	368.64	368.64	368.46	368.64	368.64	368.46	0.22	0.47	0.13%

Intel C++ Compiler v5.0.1 SuSE Linux 7.3 einfache Optimierung 377.33

Reference	Runtime				Ratio				Mean	Variance	Std. Dev.	C.O.V.
	Run1	Run2	Run3	Run1	Run2	Run3	Run1	Run2				
164.gzip	1400	313.111	311.621	311.702	447	449	449	449	449	1.44	1.20	0.27%
175.vpr	1400	497.017	497.154	498.193	282	282	281	282	281	0.13	0.36	0.13%
176.gcc	1100	409.563	413.484	411.701	269	266	267	267	267	1.63	1.28	0.48%
181.mcf	1800	834.879	835.615	833.021	216	215	216	216	216	0.12	0.35	0.16%
186.crafty	1000	153.712	153.727	153.678	651	651	651	651	651	0.01	0.11	0.02%
197.parser	1800	529.543	529.057	529.158	340	340	340	340	340	0.03	0.16	0.05%
252.eon	1300	285.883	285.827	285.947	455	455	455	455	455	0.01	0.10	0.02%
253.perlbmk	1800	335.030	334.259	334.236	537	539	539	539	538	0.53	0.73	0.13%
254.gap	1100	274.414	274.032	274.023	401	401	401	401	401	0.11	0.33	0.08%
255.vortex	1900	408.150	407.354	407.371	466	466	466	466	466	0.27	0.52	0.11%
256.bz2	1500	434.319	434.605	432.808	345	345	347	345	346	0.59	0.77	0.22%
300.twolf	3000	919.290	917.246	923.871	326	327	325	326	326	1.44	1.20	0.37%
Geo.mean		377.17	377.24	377.30	377.33	377.33	377.30	377.33	377.23	0.00	0.06	0.02%
Geo.mean (without eon)		370.81	370.88	370.95	370.99	370.88	370.95	370.99	370.88	0.01	0.07	0.02%
Geo.mean (without gcc, eon, perlbnk)		368.82	369.21	369.12	369.16	369.05	369.12	369.16	369.05	0.04	0.20	0.05%

CINT2000 Result

© Copyright 1999, Standard Performance Evaluation Corporation

Intel C++ Compiler v5.0.1
SuSE Linux 7.3

SPECint2000 = 410
SPECint_base2000 = 0.00

SPEC license #: 925 Tested by: Hatz Claudio Test date: 13.2.2002 Hardware Avail: -- Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	200 400 600 800			
164.gzip				281	498				
175.vpr				461	304				
176.gcc				399	276				
181.mcf				828	217				
186.crafty				148	677				
197.parser				444	405				
252.eon				286	455				
253.perlbnk				295	609				
254.gap				236	466				
255.vortex				320	595				
256.bzip2				421	357				
300.twolf				896	335				

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: SuSE Linux 7.3
Compiler: Intel C++ Compiler v5.0.1
File System: ext3
System State: Default

Notes/Tuning Information

Peak tuning: 2-pass

Portability:
252.eon 1-Pass (-O3 -tpp6)

Errors

'reportable' flag not set during run
No 'base' runs! Base measurement required!


```
# Invocation command line:

# /home/claudio/spec2000/bin/runspec --tune=peak --config=linux_intel_2pass.cfg int
#####

#####
# Compiler selection

default=default=default=default:
CC      = icc
CXX     = icc

#####
# Portability flags

176.gcc=default=default=default:
COPTIMIZE = -mp

186.crafty=default=default=default:
EXTRA_CFLAGS=-DLINUX_i386

252.eon=default=default=default:
EXTRA_CXXFLAGS=-DHAS_ERRLIST -DFMAX_IS_DOUBLE

253.perlbmk=default=default=default:
EXTRA_CFLAGS=-DSPEC_CPU2000_LINUX_I386 -DSPEC_CPU2000_NEED_BOOL -DSPEC_CPU2000_GLIBC22

254.gap=default=default=default:
EXTRA_CFLAGS=-DSYS_IS_USG -DSYS_HAS_IOCTL_PROTO -DSYS_HAS_TIME_PROTO -DSYS_HAS_SIGNAL_PROTO -
DSYS_HAS_ANSI -DSYS_HAS_CALLOC_PROTO

#####
# Peak Tuning Flags

252.eon=peak=default=default:
CXXOPTIMIZE = -O3 -tpp6

int=peak=default=default:
PASS1_CFLAGS= -prof_gen
PASS2_CFLAGS= -axK -ipo -prof_use
PASS1_LDFLAGS= -prof_gen
PASS2_LDFLAGS= -axK -ipo -prof_use

#####
# Note information

int=default=default=default:
hw_vendor   = Intel C++ Compiler v5.0.1
hw_model    = SuSE Linux 7.3
hw_cpu      = AMD Athlon 1.2 GHz
hw_cpu_mhz  = 1200
hw_fpu      = Integrated
hw_ncpu     = 1
hw_ncpuorder= 1
hw_pcache  = 64KBI + 64KBD on chip
hw_scache   = 256KB(I+D) on chip
hw_tcache   = N/A
hw_ocache   = N/A
hw_memory   = 512MB PC133 SDRAM CL2
hw_disk     = Maxtor 91536U6
hw_other    = Shuttel AK12A Mainboard
sw_os       = SuSE Linux 7.3
sw_compiler = Intel C++ Compiler v5.0.1
sw_file     = ext3
sw_state    = Default
#license_num=
tester_name = Hatz Claudio
test_date   = 13.2.2002
```

Intel C++ Compiler v5.0.1 SuSE Linux 7.3 2-Pass 410.47

	Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
		Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	1400	283.153	280.917	280.920	494	498	498	497	5.15	2.27	0.46%
175.vpr	1400	460.730	462.411	460.818	304	303	304	303	0.39	0.62	0.20%
176.gcc	1100	398.142	398.740	401.398	276	276	274	275	1.42	1.19	0.43%
181.mcf	1800	828.447	828.072	827.406	217	217	218	217	0.02	0.14	0.06%
186.crafty	1000	147.657	147.673	147.621	677	677	677	677	0.02	0.12	0.02%
197.parser	1800	444.072	444.057	443.952	405	405	405	405	0.00	0.06	0.01%
252.eon	1300	285.687	285.501	285.543	455	455	455	455	0.02	0.16	0.03%
253.perlbnk	1800	296.149	295.432	295.496	608	609	609	609	0.67	0.82	0.13%
254.gap	1100	236.627	236.084	236.046	465	466	466	466	0.41	0.64	0.14%
255.vortex	1900	319.781	319.156	319.520	594	595	595	595	0.34	0.58	0.10%
256.bzip2	1500	421.368	420.061	420.639	356	357	357	357	0.31	0.56	0.16%
300.twolf	3000	895.713	896.729	895.405	335	335	335	335	0.07	0.26	0.08%
Geo.mean					409.98	410.41	410.30	410.23	0.05	0.22	0.05%
Geo.mean (without eon)					406.11	406.55	406.44	406.37	0.05	0.23	0.06%
Geo.mean (without gcc, eon, pearlbnk)					405.30	405.79	405.97	405.69	0.12	0.35	0.09%

CINT2000 Result

© Copyright 1999, Standard Performance Evaluation Corporation

GNU gcc version 2.95.3
SuSE Linux 7.3

SPECint2000 = 340
SPECint_base2000 = 249

SPEC license #00925 Tested by: Hatz Claudio Test date: 24.1.2002 Hardware Avail: -- Software Avail: --

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	477	293	288	486	
175.vpr	1400	630	222	499	280	
176.gcc	1100	427	258	319	344	
181.mcf	1800	886	203	833	216	
186.crafty	1000	292	342	174	574	
197.parser	1800	717	251	494	364	
252.eon	1300	1173	111	1172	111	
253.perlbmk	1800	439	410	346	520	
254.gap	1100	455	242	278	395	
255.vortex	1900	530	359	430	442	
256.bzip2	1500	791	190	453	331	
300.twolf	3000	1158	259	902	333	

Hardware

CPU: AMD Athlon 1.2 GHz
CPU MHz: 1200
FPU: Integrated
CPU(s) enabled: 1
CPU(s) orderable: 1
Parallel: --
Primary Cache: 64KBI + 64KBD on chip
Secondary Cache: 256KB(I+D) on chip
L3 Cache: N/A
Other Cache: N/A
Memory: 512MB PC133 SDRAM CL2
Disk Subsystem: Maxtor 91536U6
Other Hardware: Shuttel AK12A Mainboard

Software

Operating System: SuSE Linux 7.3
Compiler: GNU gcc version 2.95.3
File System: ext3
System State: Default

Notes/Tuning Information

Base tuning: no flags
Peak tuning: -O3 -march=i686

Portability:
252.eon peak: no flags

```
# Invocation command line:

# /home/claudio/spec2000/bin/runspec --tune=all --reportable --config=linux_gcc.cfg -I int
#####

#####

# Compiler selection

default=default=default=default:
CC      = gcc
CXX     = g++

#####

# Portability flags

186.crafty=default=default=default:
EXTRA_CFLAGS=-DLINUX_i386

252.eon=default=default=default:
EXTRA_CXXFLAGS=-DHAS_ERRLIST
#CXXOPTIMIZE = -O0

253.perlbmk=default=default=default:
EXTRA_CFLAGS=-DSPEC_CPU2000_LINUX_I386 -DSPEC_CPU2000_NEED_BOOL -DSPEC_CPU2000_GLIBC22

254.gap=default=default=default:
EXTRA_CFLAGS=-DSYS_IS_USG -DSYS_HAS_IOCTL_PROTO -DSYS_HAS_TIME_PROTO -DSYS_HAS_SIGNAL_PROTO -
DSYS_HAS_ANSI -DSYS_HAS_CALLOC_PROTO

#####

# Peak Tuning Flags

int=peak=default=default:
COPTIMIZE = -O3 -march=i686

#####

# Note information

int=default=default=default:
hw_vendor   = GNU gcc version 2.95.3
hw_model    = SuSE Linux 7.3
hw_cpu      = AMD Athlon 1.2 GHz
hw_cpu_mhz  = 1200
hw_fpu      = Integrated
hw_ncpu     = 1
hw_ncpuorder= 1
hw_pcache  = 64KBI + 64KBD on chip
hw_scache  = 256KB(I+D) on chip
hw_tcache  = N/A
hw_ocache  = N/A
hw_memory  = 512MB PC133 SDRAM CL2
hw_disk    = Maxtor 91536U6
hw_other   = Shuttel AK12A Mainboard
sw_os      = SuSE Linux 7.3
sw_compiler= GNU gcc version 2.95.3
sw_file    = ext3
sw_state   = Default
#license_num=
tester_name = Hatz Claudio
test_date  = 24.1.2002
```

GNU gcc v2.95.3 **SuSE Linux 7.3** **keine Optimierung** **248.96**

Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
	Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	478.074	477.056	476.436	293	293	294	293	0.26	0.51	0.17%
175.vpr	630.064	629.362	630.770	222	222	222	222	0.06	0.25	0.11%
176.gcc	424.607	426.678	426.552	259	258	258	258	0.50	0.70	0.27%
181.mcf	881.063	887.903	886.301	204	203	203	203	0.68	0.82	0.40%
186.crafty	292.431	292.482	292.451	342	342	342	342	0.00	0.03	0.01%
197.parser	717.103	715.521	716.859	251	252	251	251	0.09	0.30	0.12%
252.eon	1172.68	1172.57	1171.98	111	111	111	111	0.00	0.04	0.03%
253.perlbmk	439.898	439.294	438.841	409	410	410	410	0.24	0.49	0.12%
254.gap	455.012	455.442	455.558	242	242	241	242	0.02	0.15	0.06%
255.vortex	529.820	529.714	530.005	359	359	358	359	0.01	0.10	0.03%
256.bz2	791.943	790.522	790.404	189	190	190	190	0.04	0.21	0.11%
300.twolf	1160.25	1156.89	1158.50	259	259	259	259	0.14	0.38	0.15%
Geo.mean				249.05	249.01	248.98	248.96	0.00	0.03	0.01%
Geo.mean (without eon)				268.06	268.01	267.97	267.96	0.00	0.04	0.02%
Geo.mean (without gcc, eon, perlbnk)				256.73	256.77	256.69	256.70	0.00	0.04	0.02%

GNU gcc v2.95.3 **SuSE Linux 7.3** **einfache Optimierung** **339.66**

Reference	Runtime			Ratio			Mean	Variance	Std. Dev.	C.O.V.
	Run1	Run2	Run3	Run1	Run2	Run3				
164.gzip	289.690	288.264	287.936	483	486	486	485	2.45	1.57	0.32%
175.vpr	499.243	499.589	499.219	280	280	280	280	0.01	0.12	0.04%
176.gcc	318.905	322.661	319.390	345	341	344	343	4.76	2.18	0.64%
181.mcf	835.539	831.831	832.738	215	216	216	216	0.25	0.50	0.23%
186.crafty	174.110	174.129	174.145	574	574	574	574	0.00	0.06	0.01%
197.parser	493.880	493.877	493.768	364	364	365	364	0.00	0.05	0.01%
252.eon	1172.58	1172.42	1172.42	111	111	111	111	0.00	0.01	0.01%
253.perlbmk	347.190	346.053	346.064	518	520	520	520	0.96	0.98	0.19%
254.gap	278.309	278.211	278.211	395	395	395	395	0.01	0.08	0.02%
255.vortex	429.633	429.283	430.240	442	443	442	442	0.25	0.50	0.11%
256.bz2	453.849	452.547	453.024	331	331	331	331	0.23	0.48	0.15%
300.twolf	906.222	900.318	901.589	331	333	333	332	1.30	1.14	0.34%
Geo.mean				339.17	339.47	339.66	339.43	0.06	0.25	0.07%
Geo.mean (without eon)				375.46	375.82	376.04	376.05	0.09	0.30	0.08%
Geo.mean (without gcc, eon, perlbnk)				365.66	366.44	366.29	366.13	0.17	0.41	0.11%



CINT2000 Result

Copyright ©1999-2000, Standard Performance Evaluation Corporation

Advanced Micro Devices
ASUS A7V Motherboard 1.2GHz Athlon processor

SPECint2000 = 458
SPECint_base2000 = 409

SPEC license #: 49 Tested by: AMD Austin TX Test date: Dec-2000 Hardware Avail: Oct-2000 Software Avail: Oct-2000

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio	
164.gzip	1400	297	472	249	563	
175.vpr	1400	549	255	491	285	
176.gcc	1100	444	248	310	355	
181.mcf	1800	927	194	918	196	
186.crafty	1000	158	632	147	678	
197.parser	1800	484	372	483	373	
252.eon	1300	188	692	123	1056	
253.perlbnk	1800	270	668	250	720	
254.gap	1100	249	441	249	441	
255.vortex	1900	271	702	260	731	
256.bzip2	1500	448	335	437	343	
300.twolf	3000	883	340	832	360	

Hardware

CPU: 1.2GHz AMD Athlon processor A1200AMT3B
 CPU MHz: 1200
 FPU: Integrated
 CPU(s) enabled: 1
 CPU(s) orderable: 1
 Parallel: No
 Primary Cache: 64KBI + 64KBD on chip
 Secondary Cache: 256KB(I+D) on chip
 L3 Cache: N/A
 Other Cache: N/A
 Memory: 256MB PC133 SDRAM CL2
 Disk Subsystem: WDC WD153BA
 Other Hardware: Ethernet, NVidia Riva TNT2 Model 64

Software

Operating System: Windows 2000
 Compiler: Intel C/C++ 5.0, Microsoft Visual Studio 6.0 (libraries)
 MicroQuill Smartheap Library 5.0
 File System: FAT32
 System State: Default

Notes/Tuning Information

```
+FDO: PASS1=-Qprof_gen PASS2=-Qprof_use
Baseline C: icl -QaxK -Qipo +FDO shlw32m.lib ONESTEP
Baseline C++: icl -QaxK -Qipo -GX -GR ONESTEP
shlw32m.lib is the SmartHeap library V5.0 from MicroQuill www.microquill.com
Portability: 176.gcc -Dalloca=_alloca /F10000000, 186.crafty -DNT_i386
253.perlbnk -DSPEC_CPU2000_NTOS -DPERLDLL /MT
254.gap -DSYS_HAS_CALLOC_PROTO -DSYS_HAS_MALLOC_PROTO
Peak tuning: ONESTEP plus
164.gzip: -O3 -Qxi -Qipo -Oi- +FDO
175.vpr: -O3 -Qxi -QaxK -Qwp_ipo +FDO
176.gcc: -O3 -QxiM -QaxK -Qipo -Oi- +FDO shlw32m.lib
181.mcf: -O3 -Qxi -QaxK -Qwp_ipo +FDO
186.crafty: -O3 -Qxi -Qwp_ipo +FDO
197.parser: -O3 -Qxi -QaxK -Qwp_ipo +FDO
252.eon: -O3 -QxiM -Qwp_ipo +FDO
253.perlbnk: -O3 -QxiM -Qwp_ipo +FDO shlw32m.lib
254.gap: -O3 -QaxK -Qwp_ipo +FDO shlw32m.lib
255.vortex: -O3 -Qxi -QaxK -Qipo +FDO shlw32m.lib -Oa -Oi-
256.bzip2: -O3 -Qxi -QaxK -Qwp_ipo -Oa +FDO
300.twolf: -O3 -Qxi -QaxK -Qwp_ipo -Oa +FDO shlw32m.lib
```

Anhang B:

Zwischenresultate

I	C	O	B	CO	CB	OB	COB	y	Mean y	Estim. y	Errors
1	1	1	1	1	1	1	1	371.13	371.97	371.76	-0.63
1	1	1	-1	1	-1	-1	-1	368.82	369.12	369.05	-0.23
1	1	-1	1	-1	1	-1	-1	371.91	372.89	372.52	-0.61
1	1	-1	-1	-1	-1	1	1	367.93	368.80	368.46	-0.53
1	-1	1	1	-1	1	1	-1	357.71	358.38	358.40	0.24
1	-1	1	-1	-1	-1	-1	1	365.66	366.44	366.13	-0.47
1	-1	-1	1	1	-1	-1	1	251.49	251.41	251.47	0.02
1	-1	-1	-1	1	1	1	-1	256.73	256.77	256.73	0.00
2714	249	216	-6	-216	20	-4	1	Total			
339	31	27	-1	-27	2	-1	0	Total/8			

se = 0.38317
sq = 0.07821
t16 = 4.015
0.314

SSY = 2821242.38
SSO = 2762741.03
SSC = 23303.90
SSO = 17484.17
SSB = 15.64
SSCO = 17538.50
SSCB = 149.91
SSOB = 6.19
SSCOB = 0.69
SSE = 2.35
SST = 58501.36

39.83%
29.89%
0.03%
29.98%
0.26%
0.01%
0.00%
0.00%
100.00%

Compiler: 1 = Intel, -1 = GNU
Optimierung: 1 = einfache Opt., -1 = keine Opt.
Betriebssystem: 1 = Win2000, -1 = Linux