

**ho-COMPUTER Entwicklertag
High Performance
Programmierung
in München am 18.Mai**

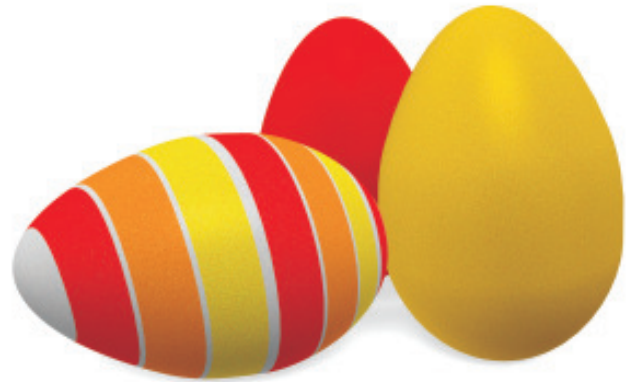
**8 Regeln zum parallelen
Programmieren**

Intel® Parallel Studio Promotion

Pakete & Lizenzmodelle

Inhalt

Editorial	2
ho-COMPUTER HPC-Entwicklertag	2,3
8 Regeln f.d. Parallelprogrammierung	3-5
Pakete und Lizenzmodelle	6
Sonderpreisliste Ostern 2010	7
Intel® Parallel Studio Promotion	8



15% **Osterrabatt**
auf alle Intel-Softwareprodukte
nur bis Ende April



h.o.-COMPUTER Software GmbH - <http://www.hocomputer.de> - info@hocomputer.de
Amsterdamer Str.91, D-50735 Köln, Tel.: (+49) 0221 / 76 20 86, Fax: (+49) 0221/ 760 17 49

© 2010 h.o.-COMPUTER Software GmbH. Nachdruck, Vervielfältigung oder Publikation in elektronischen Medien nur mit ausdrücklicher, schriftlicher Genehmigung.
Druckfehler, Preisänderungen, Versionswechsel und Irrtümer vorbehalten. Alle Angaben ohne Gewähr.
© 2009 Intel Corporation Intel, the Intel logo, Pentium, Itanium, Intel Xeon and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. *Other names and brands may be claimed as the property of others.



Liebe Leserinnen und Leser,

Ostern steht vor der Tür und gerade als ich diese Zeilen schreibe, schiebt sich auch die Sonne durch die Wolken. Wie es scheint ist der Winter endlich vorbei. Grund genug für eine österliche Sonderkation: bis Ende April sparen Sie 15% bei allen Intel-Softwareprodukten! Zusätzlich schenken wir Ihnen noch Parallel Studio beim Kauf bestimmter Compiler-Bundles, alle Details hierzu finden Sie auf der Rückseite.

Paralleles Programmieren wird immer wichtiger. James Reinders, Director of Software Development Products bei Intel, hat hierzu einen spannenden Leitfaden geschrieben, den wir Ihnen nicht vorenthalten möchten.

Neben der Parallelisierung ist die Optimierung von Programmen der wohl wichtigste Aspekt in der Softwareentwicklung. Was hilft Ihnen der beste Compiler, wenn Sie nicht wissen, wie Sie ein Maximum an Performance aus ihm herausholen können? Genau darum geht es beim nächsten ho-COMPUTER Entwicklertag. Auf vielfachen Wunsch sind wir diesmal bei Intel in München zu Gast. Melden Sie sich rechtzeitig an, die Teilnehmerzahl ist eng begrenzt.

Viel Spass beim Lesen dieser ho.-COMPUTER News! wünscht Ihnen Ihr



Harald Odendahl, Geschäftsführer der
h.o.-COMPUTER Software GmbH

ho-COMPUTER Entwicklertag High Performance Programmierung

HPC-Programmierung mit den Intel Software Tools

Die Architektur moderner Rechner stellt eine enorme Herausforderung für den Programmierer dar. Die Intel Compiler & Tools bieten Ihnen maximale Performance - wenn Sie denn richtig genutzt werden. Am 18. Mai informieren Sie die Experten von Intel aus erster Hand über den aktuellen Stand der modernen HPC-Programmierung. Wir beginnen erst um 10 Uhr morgens, um Ihnen eine Anreise am selben Tag zu ermöglichen.

Vorläufige Agenda

■ Moderne Multi-Core Prozessoren von Intel

Einführung in die aktuelle Prozessor-Architektur (Intel Core™ i7), Erweiterungen der SSE-Instruktionen

und der Multi-Core Architektur; Ausblick auf die zukünftige Prozessorentwicklungen bei Intel. Diese Präsentation wird insbesondere deutlich machen, warum ohne eine effiziente Parallelisierung der Software die Leistung moderner Prozessoren nicht mehr ausgenutzt werden kann.

■ Intel Fortran- und C/C++-Compiler und Intel Bibliotheken

Dieser Teil behandelt die wichtigste Funktionalität der Intel Compiler und warum die Intel Compiler sinnvoll als Ersatz oder Ergänzung der sonst verwendeten Compiler benutzt werden können. Ein Schwerpunkt hier ist die Generierung von parallelem Code durch automatische Vektorisierung, OpenMP oder automatische Parallelisierung. Anschliessend Mittagessen.

■ **Test und Fehlersuche bei parallelen Programmen mit Intel Thread Checker und Debugger**

Bei der Entwicklung paralleler Programme wird der Programmierer mit Fehlern konfrontiert, die es in einem sequentiellen Programm nicht gibt. Dazu gehören Zugriffs-Konflikte (Race-Conditions) und Verklemmungen (Deadlocks). Intel bietet mit dem Parallelen Debugger, dem Intel ThreadChecker und dem Intel Inspector des Parallel Studios eine Reihe von Werkzeugen an, die helfen, diese Probleme schnell zu finden und zu korrigieren. Anschliessend Kaffeepause.

■ **Software Werkzeuge zur Programmierung von Systemen mit verteiltem Speicher - MPI, Intel Trace Tools, Message Checker**

Dieser Vortrag widmet sich den Intel Software Tools für verteilte Programmierung. Es geht hierbei in der Hauptsache um Programmierung und Performanceanalyse in MPI Anwendungen.

■ **Leistungsanalyse paralleler Programme mit Intel VTune™ Performance Analyzer und Intel Thread Profiler**

FastFacts ho-Entwicklertag

Wann? 18. Mai 2010, 10-17 Uhr

Wo? Intel GmbH, Feldkirchen bei München

**Zielgruppe: Professionelle C++-
und Fortran-Programmierer**

Kosten: Keine, aber Anmeldung erforderlich:

<http://hocomputer.de/ET>

Auch das Optimieren eines parallelen Programmes stellt neue Anforderungen an den Entwickler. Wie gut ist die Parallelisierung gelungen? Benutzt das Programm alle Kerne des Prozessors optimal und skaliert es gut? Ist die Lastverteilung ausgeglichen? Wie viel Zeit geht durch kritische Regionen verloren, in denen nur ein Thread aktiv sein kann? Auch dafür stellt Intel diverse Analysetools zur Verfügung, die hier vorgestellt werden.

■ **Fragen, Antworten und offene Diskussion**

Gegen 17 Uhr Ende der Veranstaltung.

8 Regeln zur parallelen Programmierung für Mehrkernprozessoren

Von James Reinders

Das Schreiben von Programmen für Mehrkernprozessoren bringt neue Herausforderungen mit sich. Hier finden Sie acht Regeln, wie Sie erfolgreich Multicore-Anwendungen programmieren.

Denken Sie parallel. Suchen Sie bei allen Problemen nach Möglichkeiten zur Parallelisierung. Ergründen

1

Sie, wo Parallelisierung vorhanden sein könnte, und versuchen Sie in Ihrem Denken die Parallelität zum Ausdruck zu bringen. Wählen Sie den besten parallelen Ansatz, noch bevor Sie über Entwurf oder Implementierung nachdenken. Lernen Sie, parallel zu denken.

2 Verwenden Sie zum Programmieren Abstraktionen. Konzentrieren Sie sich auf das Schreiben von Code, um Parallelisierung zu formulieren, aber schreiben Sie keinen Code, der Threads oder Prozessorkerne verwalten soll. Bibliotheken, OpenMP* und Intel® Threading Building Blocks (Intel® TBB) sind Beispiele für den Einsatz von Abstraktionen. Verwenden Sie keine Raw Native Threads (Pthreads, Windows*-Threads, Boost-Threads oder ähnliche). Native Threads und MPI sind wie Assemblersprachen für Parallelisierung. Sie bieten maximale Flexibilität; das Schreiben, Debuggen und Pflegen erfordern aber viel Zeit. Ihre Programmierung sollte auf einer so hohen Ebene angesiedelt sein, dass der Code sich mit dem eigentlichen Problem befasst und nicht mit Thread- oder Kernverwaltung.

3 Programmieren Sie in Tasks (Routinen), nicht in Threads (Kernen). Überlassen Sie die Abbildung von Aufgaben in Threads oder Prozessorkerne einer eigenständigen Funktionalität in Ihrem Programm, vorzugsweise in Form einer Abstraktion, die die Thread-/Kernverwaltung für Sie übernimmt. Erstellen Sie viele Tasks in Ihrem Programm oder einen Task, der automatisch auf Prozessorkerne verteilt werden kann (wie eine OpenMP-Schleife). Sie können so viele Tasks erstellen, wie Sie möchten und müssen sich nicht darum sorgen, dass etwas überschrieben wird.



4 Sehen Sie eine Option vor, Nebenläufigkeit abzuschalten. Schreiben Sie Programme, die ohne Nebenläufigkeit korrekt ablaufen, um die Fehlersuche zu vereinfachen. So können Sie während des Debuggings Programme zunächst mit, dann ohne Nebenläufigkeit ausführen und sehen so, ob beide Durchläufe fehlschlagen oder nicht. Das Debugging von Standardproblemen ist einfacher, wenn das Programm ohne Nebenläufigkeit ausgeführt wird. Das ist immer noch

der Normalfall und wird von den zur Zeit verfügbaren Tools besser unterstützt. Wenn etwas nur bei nebenläufiger Ausführung fehlschlägt, gibt dies einen Hinweis auf den gesuchten Fehler. Wenn Sie diese Regel nicht berücksichtigen und das Programm nicht zur Ausführung in nur einem Thread zwingen können, werden Sie mehr Zeit für das Debugging benötigen. Da Sie die Möglichkeit haben möchten, ein Programm nur für Debugging-Zwecke in einem Thread auszuführen, muss diese Version nicht besonders effizient sein. Sie dürfen nur keine parallelen Programme schreiben, die Nebenläufigkeit für eine korrekte Ausführung benötigen, so wie viele Producer/Consumer-Modelle. MPI-Programme verstoßen häufig gegen diese Regel, weshalb MPI-Programme teilweise schwer zu implementieren und auszutesten sind.

5 Vermeiden Sie Sperren (engl. locks). Sagen Sie einfach „Nein“ zu Sperren. Sperren verlangsamten Programme, reduzieren ihre Skalierbarkeit und sind Ursache für Fehler in

parallelen Programmen. Setzen Sie stattdessen implizite Synchronisation ein. Sollten Sie explizite Synchronisation benötigen, verwenden Sie atomare Vorgänge. Verwenden Sie Sperren nur als letztes Mittel. Tun Sie alles dafür, dass Sperren in Ihrem Programm nicht mehr erforderlich sind.

6 Verwenden Sie Tools und Bibliotheken, die dafür konzipiert sind, Sie bei der Parallelisierung zu unterstützen. Plagen Sie sich nicht mit alten Tools. Achten Sie darauf, wie das Tool mit paralleler Verarbeitung umgeht. Viele Tools sind noch nicht für Parallelisierung ausgelegt. Suchen Sie nach Thread-sicheren Bibliotheken, idealerweise solche, die selbst Parallelisierung verwenden.

7 Verwenden Sie eine skalierbare Speicherverwaltung. Nebenläufige Programme müssen unbedingt skalierbare Routinen zur Speicherallokation benutzen. Es gibt eine Reihe von Lösungen dafür, und ich würde sagen, dass alle besser sind als die `malloc()`-Funktion. Mit skalierbaren Speicherverwaltungsroutinen machen Sie Ihre Anwendungen schneller, denn Sie eliminieren damit globale Engpässe, können Speicher in Threads wiederverwenden, um den Cache besser zu nutzen und nehmen die richtige Partitionierung vor, um Cache Line Sharing zu vermeiden.

8 Entwerfen Sie skalierbare Programme, die auf höhere Auslastung reagieren können. Denken Sie heute schon daran, dass Ihr Programm in Zukunft mit immer mehr Belastung fertigwerden muss. Berücksichtigen Sie das. Je

größer die Anzahl an Prozessorkernen, desto mehr Arbeit kann Ihr Programm bewältigen, wenn Sie Skalierbarkeit mit einplanen. Jahr um Jahr verlangen wir von unseren Rechnern mehr Leistung. Ihr Entwurf sollte auf Parallelisierung bauen, damit Sie zukünftig größere Workloads abarbeiten können.

Ich habe beim Schreiben dieser Regeln immer Threading vorausgesetzt. Aber nur Regel 7 bezieht sich speziell auf Threading. Threading ist nicht der einzige Weg, wie Sie Mehrkernprozessoren besser nutzen können. Die Verteilung auf mehrere Programme oder Prozesse wird ebenfalls häufig eingesetzt, insbesondere bei Anwendungen auf Servern.

Diese Regeln werden Ihnen dabei helfen, das Optimum aus Mehrkernprozessoren herauszuholen. Einige werden in den nächsten 10 Jahren an Bedeutung gewinnen, wenn die Anzahl der Prozessorkerne zunimmt und auch die Kerne selbst vielfältiger werden. Die Einführung heterogener Prozessoren und NUMA beispielsweise machen Regel Nummer 3 zunehmend wichtig.

Sie sollten alle acht Regeln verstehen und beachten. Ich freue mich auf Ihre Anmerkungen zu diesen Regeln, aber auch zum Thema Parallelisierung im Allgemeinen.

James Reinders ist Chief Software Evangelist und Director of Software Development Products bei der Intel Corporation. Seine Artikel und Bücher über Parallelität umfassen „Intel® Threading Building Blocks Outfitting C++ for Multicore Processor Parallelism.“ Lesen Sie mehr unter www.go-parallel.com.

Pakete und Lizenzmodelle

Sie benötigen mehr als nur den C++ oder Fortran-Compiler? Dann sind die Intel Compiler Suites für Windows oder Linux genau richtig für Sie.

Den Inhalt der einzelnen Pakete entnehmen Sie bitte der Grafik. Immer beliebter wird das "Gesamtpaket" mit dem etwas sperrigen Namen Intel Cluster Toolkit Compiler Edition (CTC). Es beinhaltet neben den Compilern und Bibliotheken (ausser TBB und IPP) noch das Cluster Toolkit mit den Komponenten

- Intel MPI Library
- Intel Trace Analyzer and Collector
- Intel MPI Benchmarks
- Intel Debugger

Single-user-Lizenz

Die "normalen" Versionen der Intel Softwaretools sind Single-User-Lizenzen.

Diese dürfen nur von einer Person, ggf. aber auch auf mehreren Rechnern genutzt werden.

Floating Lizenzen

Soll der Compiler von mehreren Entwicklern genutzt werden, so empfehlen sich Floating-Lizenzen. Bei einer 2-user Floatinglizenz

**Jetzt Osterrabatt sichern:
shop.hocomputer.de
15% auf alle Intel-Software**

können immer 2 Nutzer (aus einer beliebig grossen Gruppe) gleichzeitig compilieren. Danach wird die Lizenz wieder freigegeben.

Da der eigentliche Compilervorgang oft nur wenige Minuten dauert, kann z.B. eine 5-user Floatinglizenz für eine Vielzahl von Entwicklern reichen. Dies relativiert auch den Preisunterschied zwischen den Single-user- und Floatinglizenzen.

Support & Upgrades

Alle hier vorgestellten Tools (ausser Intel Parallel Studio) beinhalten ein Jahr Intel Premier Support inkl. aller Updates. Für viele Produkte sind Versionen mit bis zu 5 Jahren Support erhältlich.

15% Osterrabatt sichern

Noch bis Ende April 2010 erhalten Sie 15% Rabatt auf alle in unserem Online-Shop shop.hocomputer.de gelisteten Intel-Softwareprodukte. Nutzen Sie die Gelegenheit!



Aktionspreisliste Ostern 2010

Auch auf alle anderen Intel-Produkte unter shop.hocomputer.de erhalten Sie 15% Osterrabatt!

Alle Preise in Euro zzgl. 19% MwSt. (im Inland) und längstens gültig bis Ende April 2010.

Wir liefern versandkostenfrei in die gesamte EU.

Intel C++ v11.1 für Windows oder Linux

Intel C++ Professional Edition LK	479 408,-
Intel C++ Prof. Edition 2 user floating LK	4679 1428,-
Intel C++ Prof. Edition 5 user floating LK	3599 3060,-
Intel C++ Professional Edition hs LK	429 110,-

Intel Visual Fortran v11.1 für Windows

Intel Visual Fortran Professional LK	559 476,-
Intel Visual Fortran Prof. 2 user floating LK	4949 1631,-
Intel Visual Fortran Prof. 5 user floating LK	4349 3672,-
Intel Visual Fortran Professional hs LK	229 195,-

Intel Fortran v11.1 für Linux

Intel Fortran Professional Linux LK	749 612,-
Intel Fortran Prof. Linux 2 user floating LK	2399 2040,-
Intel Fortran Prof. Linux 5 user floating LK	5599 4760,-
Intel Fortran Professional Linux hs LK	289 246,-

Intel Compiler Suite für Windows

Intel Compiler Suite Win. LK	799 680,-
Intel Compiler Suite Win. 2 user float. LK	2799 2380,-
Intel Compiler Suite Win. 5 user float. LK	6459 5236,-
Intel Compiler Suite Win. hs LK	269 229,-
Intel Compiler Suite Win. hs 2 user float. LK	979 833,-

Intel Compiler Suite für Linux

Intel Compiler Suite Linux LK	4079 918,-
Intel Compiler Suite Lin. 2 user float. LK	3869 3288,-
Intel Compiler Suite Lin. 5 user float. LK	8279 7038,-
Intel Compiler Suite Linux hs LK	369 314,-
Intel Compiler Suite Lin. hs 2 user float. LK	4325 1127,-

Intel Cluster Toolkit Compiler Edition f.Windows

Intel Cluster Toolkit Comp.Ed. (CTC) Win.LK	4039 884,-
mit 3 Jahren Support	4839 1563,-
mit 5 Jahren Support	2639 2244,-
Intel CTC 2 user float. Win. LK	3199 2720,-
mit 3 Jahren Support	5759 4895,-
Intel CTC 5 user float. Win. LK	7999 6799,-
Intel CTC hs LK	399 339,-
Intel CTC 2 user float. hs Win. LK	4279 1088,-

Intel Cluster Toolkit Compiler Edition für Linux

Intel Cluster Toolkit Comp.Ed. (CTC) Lin.LK	4359 1156,-
mit 3 Jahren Support	2399 2040,-
mit 5 Jahren Support	3535 2999,-
Intel CTC 2 user float. Lin. LK	3999 3399,-
mit 3 Jahren Support	7199 6120,-
mit 5 Jahren Support	40399 8840,-
Intel CTC 5 user float. Lin. LK	40459 8636,-
Intel CTC Lin. hs LK	545 464,-
Intel CTC 2 user float. hs Lin. LK	4599 1360,-

Intel Parallel Studio für Windows

Intel Parallel Studio LK	639 544,-
zusammen mit C++ oder Fortran Compiler (Win.)	320 0,-
zusammen mit Compiler Suite oder CTC (Win.)	0 0,-
Intel Parallel Composer LK	349 271,-
Intel Parallel Inspector LK	349 271,-
Intel Parallel Amplifier LK	349 271,-
Intel Parallel Studio - Single user hs LK	459 135,-

Hinweise und Erläuterungen: **hs=Hochschulversion** (NUR für Hochschulen, NICHT für Forschungseinrichtungen o.ä.), **LK=Licence Key Version** (nur Lizenzschlüssel, Software per Download oder auf kostenloser Test-CD), BOX-Versionen (CD, Seriennummer) sind für viele Produkte gegen einen Aufpreis von 30 EUR lieferbar. Studentenlizenzen und viele weitere Produkte finden Sie unter shop.hocomputer.de.

Liefer- und Zahlungsbedingungen: Alle Preise sind in Euro und verstehen sich **zuzüglich 19% Mehrwertsteuer und inkl. Versandkosten in die EU**, sind generell freibleibend und gelten nur für gewerbliche Abnehmer oder öffentliche Institutionen. Der Versand erfolgt per DHL, LK-Versionen werden per E-Mail geliefert. Großunternehmen und öffentliche Institutionen im In- und EU-Ausland werden auf Rechnung beliefert. Wechselkursänderungen oder Irrtum vorbehalten. In unserer Preiskalkulation sind bereits Skontoabzüge berücksichtigt. Es gelten ausschließlich unsere allgemeinen Geschäftsbedingungen. Bestellen Sie bitte per Fax (0221/7601749), Post (h.o.-COMPUTER, Amsterdamer Straße 91, 50735 Köln) oder in unserem Online-Shop unter shop.hocomputer.de.

Intel Parallel Studio Promotion

Sie möchten Multithread-Code schreiben, um die Vorteile von mehreren Prozessoren und Mehrkernprozessoren voll auszunutzen? Das ist keine leichte Aufgabe, aber mit dem neuen Intel Parallel Studio wird es Ihnen gelingen.

Mit der zunehmenden Anzahl neuer Mehrkernprozessoren steigt der Druck auf den Anwendungsentwickler, diese Rechenleistung auch effektiv einzusetzen. Entwickler müssen die Möglichkeiten für Multithreading und unabhängige parallele Rechenvorgänge in ihren Anwendungen erkennen und diese Techniken auch beim Programmieren umsetzen können.

Mit Intel Parallel Studio steht Entwicklern, die parallele Programme mit C++ schreiben, eine umfassende Lösung für die Parallelisierung zur Verfügung. Parallel Studio arbeitet perfekt mit Microsoft Visual Studio zusammen und unterstützt höhere Abstraktionen für die Parallelverarbeitung, um die Entwicklung zu ver-

einfachen und zu beschleunigen. Die Tools sind so konzipiert, dass sich Neueinsteiger während der laufenden

Arbeit schrittweise mit der Lösung vertraut machen, während professionelle

Entwickler ihre vorhandenen und neuen Projekte noch einfacher auf Parallelverarbeitung umstellen können.

Intel Parallel Studio Sonderaktion

Bestellen Sie bis zum 30.4.2010 Parallel Studio zusammen mit einer kommerziellen Lizenz des Intel C++ oder Fortran Compiler für Windows und Sie bekommen Parallel Studio zum halben Listenpreis.

Oder nehmen Sie die Compiler Suite oder das Cluster Toolkit Compiler Edition (beides für Windows, kommerzielle Lizenzen) und wir schenken Ihnen eine Lizenz von Parallel Studio!

Die 15% Osterrabatt auf die Compiler erhalten Sie natürlich trotzdem.

„Ich weiß, wie man 4 Pferde dazu bringt, einen Wagen zu ziehen – aber nicht, wie man 1024 Hühner dazu bringt.“
—Enrico Clementi



oder



oder

