Zu starke Optimierung

Ein Student der Elektrotechnik, Emil H., der zu Beginn seines Studiums mehr im RRZN als Hilfskraft tätig war als in seinem Studienfach, saß brütend vor einer kleinen Programmliste. Weil sein Programm intensiv rechnen sollte, hatte er den FORTRAN-Compiler um OPT=2 gebeten. Das Programm brach ab. Nun hatte er sich endlich durchgerungen, etwas für sein Diplom zu tun, und wurde auf diese Weise abgestraft.

Emil hatte das Programm mit OPT=0 (keine Optimierung), OPT=1 (Optimierung) und OPT=2 (höchste Optimierung) übersetzen lassen, sich die erzeugten Assembler-Befehle listen lassen und getestet. Mit OPT=0 und OPT=1 lief es korrekt. OPT=2 sollte dann die Laufzeit auf ein erträgliches Maß verringern.

Die drei Listen brachten die bittere Wahrheit ans Tageslicht, die für Emil unfassbar war. Eine ganz einfache Wertzuweisung - sinngemäß A=1.0 - stand vor eine DO-Schleife. Der Compiler hatte nicht nur die Schleife optimiert, sondern auch die Wertzuweisung wegoptimiert! Emil musste sich mit dem länger, aber immerhin korrekt laufenden OPT=1 begnügen und wegen der zu starken Optimierung seine Kollegen in der Beratung bzw. den Compiler-Experten Herrn Gehrke konsultieren.

Ich selber habe bei meinen Programmen OPT=2 wegen der deutlich längeren Zeit für die Compilation vermieden und immer OPT=1 benutzt. War anscheinend eine kluge Entscheidung.

Die Grenzen eines Compilers

Wie schön ist es doch, wenn ein Institut den Quellentext eines umfangreichen Programmsystems geschenkt bekommt und einem Mitarbeiter in die Hand drückt mit der Aufforderung, es auf der Cyber76 zu laufen zu bringen.

Keine Frage, die Programme waren gut strukturiert und flexibel. Was man zu einem Formalparameter machen konnte, das war dann auch einer. Sehr lobenswert, aber leider hatte der FORTRAN-Compiler von CDC sich in dieser Hinsicht eine Grenze gesetzt: nicht mehr als 64 Formalparameter. Einige Unterprogramme hatten deutlich mehr Parameter.

Auch in diesem Fall konnte das RRZN helfen. Mit einem Satz von Steuerkarten, die der Benutzer nicht publik machen sollte, übersprang der Compiler auch diese Hürde.

Die Grenzen des Rechners

Das Institut für Theoretische Geodäsie bekam ein vergleichbares Geschenk von der Uni Boulder, Colorado. Dort war ebenfalls eine Cyber76 installiert, aber mit deutlich mehr Speicher und einem anderen Betriebssystem. Im RRZN waren 60000B Worte nutzbar, in Boulder 200000B. Auf der Cyber76 im RRZN lief SCOPE 2.0, in Boulder NOS/BE.

Die Programme schrieben Zwischenergebnisse in sequentielle Dateien. Andere Programme wussten, ab welchem Satz sie eine Datei lesen mussten. Aber die Sätze davor wurde nicht überlesen, sondern mit einem Unterprogramm in COMPASS, dem CDC-Assembler, übersprungen. Dieses Unterprogramm wurde Skip-Routine genannt.

Die erste Aufgabe bestand also darin, eine Skip-Routine für die Cyber76 zu entwickeln und zu testen. Die Skip-Routine aus Boulder konnte leider nicht übernommen werden. Aber im RRZN gab es Handbücher mit Macros. Damit war das Problem lösbar.

Im nächsten Schritt ging es an die FORTRAN-Programme. Das eigentliche Problem waren die Felder mit den Daten. Sie mussten in den sog. LCM ausgelagert werden. Dort waren 200000B kein Problem. Auf den LCM konnte man auf zweierlei Art zugreifen. Im Blockmodus wurden Speicherbereiche zwischen LCM und SCM, dem eigentlichen Speicher für Programme und Daten, ausgetauscht. Das war aber keine Option, weil die Felder zu groß waren. Es gab denn noch den direkten Zugriff auf ein Wort im LCM. Das war zwar langsamer, konnte aber umgesetzt werden. Dazu mussten besondere Anweisungen eingefügt werden (LEVEL 2). Unangenehm wurde es, wenn nur ein einzelnes Feldelement als Aktualparameter in einem Aufruf vorkam, aber andere Aufrufe an dieser Stelle einen Aktualparameter aus dem SCM (LEVEL 1) hatten. Es ging nur so, dass vor dem Aufruf das Element aus dem LCM einer lokalen Variablen aus dem SCM zugewiesen wurde und nach dem Aufruf wieder an das Element im LCM.

Das gesamte Paket, das den Namen BOMM trug, bestand aus sehr vielen Unterprogrammen mit insgesamt etwa 60000 Anweisungen. Es gab auch Datensätze zum Testen und Vergleichslisten. Bis auf die hohen Speicheranforderungen war alles bedacht.

Die Umstellung auf Felder im LCM habe ich zusammen mit einem Kommilitonen vorgenommen, der im Programmieren sehr erfahren war und ein Praktikum im RRZN absolviert hatte - Wolfgang G.

Wir haben es tatsächlich geschafft. Die Ausdrucke zu den mitgelieferten Datensätzen konnten alle reproduziert werden.

Aber seitdem habe ich einen gewissen Horror, wenn jemand ein ganz tolles Programmpaket anbietet, das „nur“ für den gerade vorhandenen Rechner angepasst werden muss. Wie schrieb doch Vergil in der Aenaeis so schön? „Ich fürchte die Danaer, auch wenn sie Geschenke bringen.“