Green Programming! *(Daily Dueck84)*

Gunter Dueck, von www.omnisophie.com , Februar 2009

Der Strom, den die Computer verbrauchen, kostet bald oder jetzt schon mehr, als sie beim Kauf kosten. Über den Kaufpreis jammern alle, aber die Energie für den Rechner fließt nur so unbemerkt davon. Die Green IT Bewegung hat uns in den letzten ein, zwei Jahren wach gerüttelt. Ich habe in der vorigen Woche mit einem der ersten Sturmglockenläuter diskutiert, dem WiWo Redakteur Thomas Kuhn. Und da fiel mir ein, dass wir die Energie vielleicht noch ganz woanders verschwenden: beim schlechten Schreiben von Programmen.

Als ich 1987 beim IBM Wissenschaftlichen Zentrum in Heidelberg zu arbeiten begann, entwickelte im Büro nebenan ein begnadeter Kollege, Ulrich Schauer, Programme in der Sprache APL2, die auch heute noch Jünger hat. Ich durfte bei ihm lernen. Die Sprache APL2 wird interpretiert. Der Rechner liest also Zeile für Zeile des Programms einzeln und führt die Befehle dieser Zeile nacheinander aus. Danach liest er die nächste Zeile, interpretiert, welche Befehle sie enthält und führt die Befehle wieder aus. Alles schön Schritt für Schritt.
Eines Tages wurden wir zu einem Unternehmen gerufen, das wissen wollte, ob sich parallele Programmierungen bei seinen typischen Anwendungen lohnen würden. Ulrich Schauer fuhr mit mir hin – ich in seinem Schlepptau wie ein ahnungsloser Knappe hinter seinem Ritter. Man hatte für uns alle Programme auf Endlospapier ausgedruckt, es war ein größerer Stapel. Mein Kollege fuhr einmal grob blätternd über den Stapel, schüttelte den Kopf und verneinte: „Parallel bringt bei solchen Programmen nichts.“ Ich erinnere mich genau an diesen Augenblick. Ich war kaum je in meinem Leben so voller Bewunderung für eine Kunst. Das Unternehmen war nicht gleich so überzeugt wie ich. Wir vereinbarten, dass einen Tag lang ein Zählprogramm in den Großrechnern mitlaufen sollte. Wir würden dann wissen, welche Zeilen der Programme die Großrechner sehr oft ausführen mussten. Diese Zeilen, so vereinbarten wir, würden wir neu und besonders „schnell laufend“ implementieren. Wir reisten also ab und erschienen eine Woche später wieder.
An der Eingangspforte gab man uns zu verstehen, dass wir leider vergebens gekommen wären. Das Programm mit dem Zeilenzählen habe wohl versagt. Es hätte behauptet, eine einzige Zeile Programm im ganzen Rechenzentrum würde über drei Viertel der Rechenzeit aller Computer verschlingen. 85 Prozent! Das machte uns natürlich sehr neugierig. Wir schauten nach: Es war der Befehl, eine Zahl auf zwei Nachkommastellen zu runden (also damals auf Pfennige, es gibt ja bei Geld nicht mehr als zwei Stellen hinter dem Komma).

Da hatte jemand in dieser Zeile das Runden auf Mark und Pfennig programmiert, und zwar so: Man nehme die zu rundende Zahl mal 100, hacke dann alle Nachkommastellen weg (dafür gibt es einen Befehl, der heißt „Ganzzahlig machen“) und teile das Ergebnis wieder durch 100. Fertig.
Herr Schauer seufzte über diesem Machwerk sehr tief, ich etwas später auch. Es ist nämlich so: Ein Computer kann in einem Takt immer eine Rechnung ausführen. Einfache Rechnungen kann er eventuell in einem Takt schaffen, kompliziertere Rechnungen eben in mehreren Takten oder in vielen. Eine Addition dauert einen Takt, eine Multiplikation auch einen, das Abhacken der Stellen hinten ebenfalls einen Takt. Aber das Teilen oder Dividieren ist VIEL komplizierter als das Addieren, das dauerte damals 17 Takte. In Worten: Siebzehn! Das Dividieren muss man also beim Programmieren scheuen wie der Teufel das Weihwasser. Man muss in diesem Fall ja auch gar nicht dividieren! Überlegen Sie: Statt eine Zahl durch 100 zu teilen, kann ich sie auch mit 0,01 multiplizieren! Es kommt dasselbe heraus! Aber der Computer braucht nur einen Zeittakt statt siebzehn.
Wir haben damals folglich „geteilt durch 100“ durch „nimm mal mit 0,01“ ersetzt. Danach brauchte diese Zeile Rechenzentrum längst nicht mehr 85 Prozent der Rechenzeit! Bedenken Sie: In einem Finanzhaus runden die Rechner fast in jeder Zeile auf Cents, klar? Und diese Rundung, die fast immer dabei ist, machten wir einige Male schneller. Und diese Verbesserung schlug sich fast in jeder Programmzeile nieder! Ich war sehr, sehr stolz.

Die Mitarbeiter des Unternehmens aber freuten sich dann nicht offensichtlich. Es tat ihnen weh, das sah ich in den Gesichtern. Sie konnten sich nicht so naiv freuen wie einer, der Mathe studiert hat wie ich. Und die Leute in meiner Firma IBM schienen auch nicht so wirklich glücklich, dass wir nun serienweise Computer einsparen konnten. Wir sind damals eher ziemlich betreten heimgereist. Eine Analyse der Programme aus dem Stapel zeigte, dass mit echter allerhöchster Programmierkunst von meinem Kollegen so etwa alles drei- bis viermal schneller zu programmieren gewesen wäre …

Ich hatte sofort eine Business Idee. Wir würden das als Service verkaufen! Alles dreimal schneller! Aber niemand wollte so recht. Die Unternehmen kauften lieber schöne Computer und strickten Programme „mit der heißen Nadel“, aber Meisterschaft erwerben? Sie sagten, dass die Computer bald so schnell würden, dass es billiger wäre, die Programme zügig zu entwickeln und dadurch Geld zu sparen.

Und heute? Die Computer sind inzwischen tausend Mal schneller, aber das Hochfahren von einem PC dauert immer länger. Huuh, das ist jetzt eine unfair emotionale Formulierung, aber Sie verstehen, was ich meine? Ich weiß ja, dass die heutigen Programme wahrscheinlich tausend Mal mehr leisten, ich weiß, ich weiß. Tolle Programme! Ich bewundere sie, obwohl ich eigentlich nur tippe und surfe, wie jetzt.
Aber ich darf doch die Gretchenfrage stellen, ob die Programmierer heute auf Schnelligkeit der Programme achten? Viele von ihnen nutzen Programmbibliotheken. Sind wenigstens die auf Top-Performance gestylt? Oder werden die nach Preis eingekauft? Von der Art „Hauptsache, das Programm tut, was es soll?“

Sollten wir nicht einmal Studien anstellen, wie viel Strom das suboptimale Programmieren verschwendet? Nicht nur die Rechner?
Ich habe damals nicht wirklich das Programmieren gelernt, nur einen Hauch mitbekommen, was wahre Kunst ist. Ich wurde gleich zum Manager berufen und habe das wirkliche Programmieren übersprungen. Aber ich fühle, dass da etwas ganz Wichtiges schlummert.
Stellen Sie sich die zukünftige Zeit vor, wo ein Großteil der Energie für Computer verwendet wird. So 85 Prozent! Und dann ist doch die Idee, alles drei- bis viermal so speedy zu programmieren, nicht aus der Welt? Es geht ja dann wieder einmal um das Überleben, oder?

Ich glaube, ich muss einfach so, ohne programmieren zu können,

GREEN PROGRAMMING

fordern. So ganz sicher bin ich dabei nicht, aber eigentlich schon. Mich plagt wohl nur die Angst vor der eigenen Courage. Aber Sie? Sie können doch programmieren? Achten Sie dabei auf die Schnelligkeit des Programms oder doch nur auf die Schnelligkeit des Programmierens?

Und jetzt lesen Sie vielleicht noch den Artikel von Thomas Kuhn aus der WiWo, der gibt Ihnen in dieser Sache den Rest:

http://www.wiwo.de/technik/gezielte-kuehlung-fuer-computer-230086/