

# **DOPPELT HÄLT BESSER**

## **Die Spracklens über Mehrprozessor-systeme**

In PLY 1/90 berichten die Schweden über einen Geschwindigkeitsvergleich in 24 Teststellungen zwischen einem Elite Nr.2 (= Fidelity Elite Avant Garde Version 2) und der Doppelprozessor-Version. Das Ergebnis: der Elite Nr.5 (= Fidelity Elite Avant Garde Version 5) war einmal sogar um 8% langsamer, dann wieder bis zu 118% schneller als die Einprozessor-Version! (Der durchschnittliche Geschwindigkeitszuwachs betrug 59%.) Woher diese unterschiedlichen Werte? In der Bedienungsanleitung geben Dan und Kathe Spracklen selbst Antwort auf diese Frage. Wir haben ihre kurze Einführung in die Problematik der Mehrprozessor-Systeme so interessant gefunden, daß wir Fidelity um die Erlaubnis zum Nachdruck ersucht haben, die uns auch erteilt wurde. Wir danken der Firma für ihr Entgegenkommen.



Wenn Sie den Elite Nr.5 die gleichen Stellungen einmal im Einprozessor-Modus und dann im Doppelprozessor-Modus berechnen lassen, werden Sie feststellen, daß die Doppelprozessor-Version etwa 1,7 mal so schnell ist wie der Einfachprozessor. Vielleicht fragen Sie sich, warum sie nicht genau doppelt so schnell ist, wo sich doch zwei Prozessoren die Arbeit teilen. Daher wollen wir im folgenden ein paar Anmerkungen zur Funktionsweise des neuen Elite-Doppelprozessors machen.

Beginnen wir mit ein wenig Terminologie! Der Prozessor, der bei einem Schachcomputer die Tasteneingaben registriert und die LEDs aufleuchten läßt, wird auch "Meister" genannt. Der Prozessor, der ihm bei der Auswahl eines Zuges assistiert, heißt "Sklave". Den Vorgang, bei dem ein Zug ausgewählt wird, nennt man "Suche". Die Aufgabe des "Meisters" ist es, die Suche zu verteilen, d.h. die Sucharbeit gemeinsam mit dem "Sklaven" durchzuführen und über die Auswahl des besten Zuges zu entscheiden. In den meisten Stellungen gibt es viele mögliche Züge, und die Suche muß darüber entscheiden, welcher der beste ist.

Eine einfache und naheliegende Art, die Suche aufzuteilen, wäre, den Meister und den Sklaven an je einem Zug rechnen zu lassen. Sobald einer der beiden mit seinem Zug fertig ist, könnte er mit dem nächsten Zug auf der Liste beginnen und so weiter, bis alle Züge untersucht worden sind. Diese Methode ist durchaus anwendbar und ergibt eine Geschwindigkeitssteigerung von etwa 1,4 im Vergleich mit einem allein arbeitenden Prozessor. Warum so wenig?

Der Grund liegt in der Natur des Alpha-Beta-Algorithmus. Ohne auf lange Erörterungen dieses Verfahrens einzugehen, kann man vereinfacht sagen, daß der erste Zug, der untersucht wird, wesentlich mehr Zeit in Anspruch nimmt als irgendeiner der übrigen Züge. Der Grund dafür ist, daß die Untersuchung des ersten Zuges eine Bewertung ergibt, die für die verkürzte Bearbeitung der folgenden Züge angewendet werden kann. Wenn die Zugliste aufgeteilt ist, wobei Meister und Sklave jeweils an verschiedenen Zügen arbeiten, dann fehlt beiden Prozessoren dieser Vergleichswert, und sie sind zu einer langen und schwierigen Arbeit verurteilt.



## **Fidelity Elite Avant Garde Version 5**

**Zwei 68000-Prozessoren, 16 MHz, 192K RAM »Hash-Tabellen«. Der erste kommerziell erhältliche Multiprozessor der Welt. Die erhöhte Anzahl der »Hash-Tabellen« dient der Optimierung der Zusammenarbeit zwischen den beiden Prozessoren.**

(photo copyright © by <http://www.schaakcomputers.nl/>) (600 dpi)

Das bedeutet viel nutzlosen Aufwand, und das Endergebnis ist alles andere als beeindruckend. Es würde auch nichts helfen, den Sklaven untätig abwarten zu lassen, bis der Meister den ersten Zug durchgerechnet hat; das Ergebnis wäre in etwa das gleiche.

Eine weitaus effektivere Methode zur Aufteilung der Suche wurde von Professor Tony Marsland von der University of Alberta vorgeschlagen. Seine Methode, die auch von uns angewandt wird, beruht darauf, daß die Arbeit der langwierigen ersten Zugberechnung gemeinsam durchgeführt und erst dann der Rest der Züge aufgeteilt wird. Bei dieser Methode gibt der Meister dem Sklaven den Auftrag, gewisse Teile des Suchbaums zu bearbeiten, die für die Berechnung des ersten Zuges benötigt werden. Das Ergebnis ist die beachtliche Verschnellerung auf das 1,7-fache, die von unserem Doppelprozessor erreicht wird.

Unter "Overhead" versteht man den Zeit- bzw. Leistungsverlust, der bei der Anwendung eines bestimmten Programmierverfahrens auftritt. Bei einem Mehrfachprozessor-System gibt es vier Arten von "Overhead", die alle dazu beitragen, daß die ideale Geschwindigkeitssteigerung um den Faktor 2 nicht erreicht werden kann.

1.) Bei der KOMMUNIKATION: Das ist die Zeit, die verloren geht, weil die beiden Prozessoren miteinander in Kontakt treten und Informationen austauschen müssen. Der eine Prozessor muß Meldungen erstellen, die den anderen über die Richtung und die Ergebnisse der Sucharbeit informieren sollen. Der andere Prozessor muß diese Meldungen entgegennehmen und darauf reagieren. Das nennt man den **"COMMUNICATION OVERHEAD"**.

2.) Bei der SYNCHRONISATION: Um diesen Begriff zu verstehen, müssen wir uns überlegen, was passiert, wenn die Suche beinahe zu Ende ist. Nehmen wir an, daß der Meister gerade über den letzten und der Sklave über den vorletzten Zug nachdenkt. Einer von den beiden wird früher fertig sein als der andere und danach "arbeitslos" sein. Die Zeit, die ein Prozessor untätig damit verbringen muß, auf den anderen zu warten, nennt man den **"SYNCHRONIZATION OVERHEAD"**.

3.) Bei der ARBEITSTEILUNG: Denken wir nocheinmal an das, was wir über die Aufteilung der Sucharbeit nach der Methode "Du nimmst einen, ich nehm' einen" gesagt haben. Wir haben erwähnt, daß der Grund für die relativ geringe Effizienz dieser Methode darin liegt, daß die Bewertung, die bei der Untersuchung des ersten Zuges gefunden wird, die Bearbeitung der folgenden Züge beschleunigt. Je besser dieser erste Wert ist, desto schneller werden die anderen Züge abgearbeitet. Wenn z.B. unser erster Zug die Dame des Gegners gewinnt, brauchen wir uns nicht lange mit den übrigen Zügen aufzuhalten, die entweder weniger Material oder gar keines gewinnen. Wenn hingegen unser erster Zug nur dazu führt, daß eine Figur ein wenig besser positioniert wird, dann müssen wir uns ausführlicher mit den anderen Zugmöglichkeiten befassen.

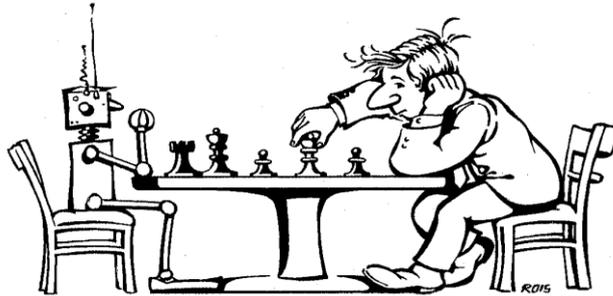
Infolge der ständigen Sortierung der Zugliste ist der erste Zug in vielen Fällen auch der beste. Manchmal kommt es aber vor, daß ein Zug, der auf der Liste weiter hinten steht, sich als besser erweist. Wenn das geschieht, ist die Bewertung, die im ersten Zug gefunden wurde, für die Bearbeitung der weiteren Züge weniger nützlich als die Bewertung des neuen Zuges. Der Prozessor, der den besseren Zug gefunden hat, wird daher sofort mit der neuen Bewertung weiterarbeiten. Der andere Prozessor muß hingegen mit dem alten Wert weitermachen, bis er mit der Bearbeitung des laufenden Zuges fertig ist. Der Leistungsverlust, der dadurch entsteht, daß die neue Bewertung nicht sofort beiden Prozessoren zur Verfügung steht, wird als "**DIVIDED SEARCH OVERHEAD**" bezeichnet.

Ein interessanter Nebeneffekt dieses Overheads zeigt sich bei Tests, die auf dem Lösen von Mattaufgaben beruhen. Der Grundgedanke bei der Verwendung von Mattproblemen zur Messung der Leistung von Schachcomputern ist, daß es einen eindeutigen, aber schwer zu findenden Gewinnzug gibt. Der Tester kann die Zeit messen, die der Computer braucht, um den Gewinnzug zu finden, und erhält dadurch einen meist recht vernünftigen Maßstab dafür, wie schnell und effizient der Schachcomputer arbeitet. Schachprobleme bestehen jedoch naturgemäß aus Stellungen, bei denen der beste Zug fast nie mit dem ersten untersuchten Zug übereinstimmt. Es sind daher Positionen mit künstlich erhöhtem Arbeitsteilungs-Overhead! Aus diesem Grund sollten Programmierer, die die Leistungsfähigkeit ihrer geteilten Suche testen wollen, besser beliebige Stellungen aus Meisterpartien verwenden als mit Mattproblemen zu arbeiten, bei denen sich der Mehrfachprozessor nicht von seiner besten Seite zeigt.

4.) Bei der Aufteilung der HASH TABLES: eine Hash Table ist ein großer Speicherbereich, in dem ein Prozessor Informationen ablegen kann, die er während des Suchvorgangs gewonnen hat. Wenn die gleiche Stellung in einer anderen Variante wieder auftaucht, kann der Prozessor das Ergebnis in der Hash Table nachschlagen, statt mit der Berechnung von vorne beginnen zu müssen. Hash Tables sind besonders in Endspielen wirksam, wo die Suche mit ihrer Hilfe um ein Vielfaches schneller abläuft als ohne sie. Der Grund dafür ist, daß im Endspiel besonders oft Stellungswiederholungen auftreten.

Der "**SPLIT HASH OVERHEAD**" entsteht dadurch, daß jeder Prozessor seine eigene Hash Table hat. Der Meister kann nicht in die Tabelle des Sklaven schauen und der Sklave nicht in die des Meisters. Wenn einer von beiden daher eine Stellung erreicht, die in der Hash Table des anderen bereits gespeichert ist, muß er dennoch mit der Berechnung von vorne beginnen. In der Eröffnung, dem Mittelspiel und sogar dem frühen Endspiel macht das keinen entscheidenden Unterschied aus. Wenn aber die Anzahl der Figuren auf dem Brett stark reduziert ist, kann der Hash-Table-Overhead die Wirksamkeit eines Mehrfachprozessor-Systems drastisch verringern. In manchen Fällen brauchen Doppelprozessor-Systeme für Bauernendspiele sogar länger als Einfachprozessoren!

Eben diese lähmende Auswirkung des SPLIT TABLE OVERHEAD hat uns dazu veranlaßt, nach Leibeskräften nach einem Ausweg zu suchen. Während wir uns die Verkaufsabteilung mit einem Arm vom Leibe hielten, suchten und fanden wir eine Lösung: wenn die Aufteilung der Hash Tables das problem verursacht, so überlegten wir, warum dann nicht mit GEMEINSAMEN Hash Tables arbeiten?



In dieser Ausführung würde nur eine einzige Tabelle verwendet werden (und zwar die des Meisters, weil sie größer ist). Der Nachteil dabei wäre ein erhöhter Kommunikations-Overhead, weil die beiden Prozessoren mehr Informationen über die Hash-Tables austauschen müßten. In der Hardware-Konfiguration, die ursprünglich für den Doppelprozessor vorgesehen war, wäre diese Methode nicht anwendbar gewesen, weil keiner der beiden Prozessoren die Möglichkeit hatte, dem anderen sozusagen "auf die Schulter zu klopfen" und ihm zu sagen: "Ich brauche Informationen über die Hash Tables" oder "Ich habe Informationen über die Hash Tables für dich". Jeder Prozessor hätte so viel Zeit damit verschwendet, darauf zu warten, daß der andere Seine Aufforderung zur Kenntnis nimmt, daß der dadurch entstehende Kommunikations-Overhead schlimmer gewesen wäre als der durch die geteilten Hash Tables verursachte. Die Methode könnte aber funktionieren, wenn der Sklave eine Möglichkeit hätte, den Meister jederzeit durch einen "Interrupt" zu unterbrechen, um Hash-Tables-Informationen abzurufen oder zu speichern.

Während wir mit der Vertriebsabteilung um Terminaufschub kämpften, setzten wir uns mit der Technischen Abteilung zusammen. Ja, durch eine genial einfache Modifizierung des Designs würde es uns möglich sein, Interrupts zu verwenden! Wie man so sagt: the rest is history! Statt im Endspiel langsamer zu werden, kann der Doppelprozessor nunmehr stolz auf eine Verschnellerung um den Faktor 1,5 bis 1,6 verweisen.

Jetzt haben Sie eine Vorstellung davon, wie intensiv die Soft- und Hardware-Experten von Fidelity gearbeitet haben, um Ihnen noch mehr Freude am Computerschach zu vermitteln. Wir hoffen, daß Ihnen die aufregenden Fortschritte auf dem Gebiet der Mehrfachprozessoren genausoviel Spaß machen werden, wie es uns Spaß gemacht hat, sie Ihnen näherzubringen.

Quelle: Fachzeitschrift Modul – Nr. 2, Juni 1990.

photo copyright © by <http://www.schaakcomputers.nl/>