

Lars Bremer

Verteiltes Rechnen @ Home

Mehrere Schachengines parallel im Heimnetzwerk betreiben

Im Internet läuft gerade die Live-Übertragung von Kasparov-Fritz. Um den Überblick zu behalten, lassen viele Schachfreunde Fritz mitanalysieren. Aber vielleicht hat Shredder mehr Durchblick in dieser Stellung, oder Ruffian, oder Yace? Aber alle gleichzeitig laufen lassen ergibt wenig Sinn, weil sich die Engines die Rechenzeit des Internet-Computers teilen müssten. Dank Netchess von Bernhard Wallner kann man aber die Analyse-Engines auf anderen Rechnern laufen lassen und dennoch so in Fritz, Shredder oder Arena einbinden, dass kein Unterschied zu lokal geladenen Engines auffällt.

Eigentlich ist es schade. Da hat man so viele Rechner zu Hause, einen für die Frau, einen für die Kinder, dazu das Laptop und natürlich Papas Arbeitspferd, aber man kann nur an einem sitzen – die anderen faulenzten inzwischen statt Schachstellungen zu berechnen. Solche Verschwendung wertvoller Rechenkapazität muss nicht sein, denn es gibt Netchess von Bernhard Wallner. Damit kann der Computerschach-Süchtige eine beliebige UCI- oder Winboard-Engine auf einem x-beliebigen Rechner im Netzwerk mit voller Kraft laufen lassen und trotzdem die Analyse auf dem lokalen Rechner unter Fritz, Shredder, ChessPartner oder Arena verfolgen, als ob die betreffende Engine auf demselben Rechner wie die GUI lief.

Fernbedienung

Technisch ist es ganz einfach: Winboard und UCI sind Protokolle auf Textbasis. Die Kommunikation zwischen Engine und GUI erfolgt ausschließlich über kurze Textzeilen. Startet man eine Engine in der Windows-Konsole, erscheinen diese Textzeilen dort, weil keine GUI sie abfängt.

Netchess fungiert als Mittler zwischen Engine und GUI. Alle Ausgaben der GUI leitet der Mittler an die Engine weiter und jedes Räuspern der Engine an die GUI. Dafür sind zwei Module zuständig: Eines (EngineServer.exe) läuft auf dem entfernten Rechner, startet dort die Engine und sammelt deren Ausgaben ein. Das andere Modul (EngineClient.exe) tut für die GUI so, als sei es das Schachprogramm und wird wie eine normale Engine eingebunden. Ob Winboard oder

UCI, hängt von der Art der entfernten Engine ab.

Beide Netchess-Module kommunizieren über das Netzwerk miteinander, sodass für den Anwender die Illusion einer unter der lokalen GUI geladenen Engine entsteht. Damit kann er in der Tat alles anstellen, ob Analyse im Mehrvariantenmodus oder Engine-Matches, was mit lokalen Engines auch funktioniert. Wer von den Lesern dieses Artikels kann beispielsweise von sich behaupten, den in Fritz integrierten Netzwerk-Autoplayer jemals zum Laufen gebracht zu haben? Mit Netchess kann man das serielle Kabel wenigstens dort einsparen, wo es nicht um ChessBase-Engines geht; ein Engine-Match ersetzt komfortabel den Autoplayer.

Auf diese Weise ist es möglich, nicht nur auf sämtlichen verfügbaren Computern eine Schachengine laufen zu lassen, sondern auch Betriebssystemgrenzen zu überwinden, denn Netchess gibt es auch für Linux.

Netchess-Workshop

Um die Engine seiner Wahl auf dem entfernten Rechner (auf dem die Engine ohne GUI rechnen soll) einzurichten, muss man das Netchess-Modul *EngineServer.exe* starten und per Dialogbox eine Engine auswählen. Für die korrekte Kommunikation ist noch eine Portnummer nötig (siehe

Kasten *Netzwerk-Einmaleins*), die größer als 1024 und für jede auf diesem Rechner laufende Engine verschieden sein sollte. Im Test liefen die entfernten Engines Ruffian1.0.1, Yace Paderborn, Shredder 7.04 und Deep Sjeng 1.6 mit Portnummern von 5001 bis 5004. Bei einem Klick auf „Save“ merkt sich EngineServer die eingegebenen Daten und man kann die nächste Engine konfigurieren.

Hat man alle Engines eingerichtet, markiert man eine oder mehrere mit Häkchen und klickt auf „Start“, wonach das Netchess-Modul auf eine Anfrage aus dem Netzwerk lauert, um



Auf dem entfernten Rechner hilft eine kleine GUI bei der Verwaltung der Engines, ...



... die, sobald gestartet, in Info-Fensterchen erscheinen.

die Engine zu starten. Das war alle Arbeit auf dem entfernten Rechner, nun geht es daran, den Computer einzurichten, auf dem die GUI läuft. Dazu startet man das Kommandozeilenprogramm *EngineClient.exe* und gibt den Engine-Namen, die IP-Adresse des Rechners, auf dem die Engine läuft (steht in EngineServer unter „Host“), sowie die zugewiesene Port-

Kleines Netzwerk-Einmaleins

IP-Adresse: Eindeutige Identifikationsnummer eines Rechners im Netzwerk, bestehend aus vier jeweils maximal dreistelligen Zahlen, die durch Punkte getrennt sind. IP-Adressen im Heimnetzwerk haben meist die Form 192.168.x.y, weil dieser spezielle Nummernbereich für private Netzwerke reserviert ist. Wenn zwei Computer zueinander Verbindung aufnehmen, braucht jeder die IP-Adresse des anderen. Dazu muss man die IP-Adressen natürlich erstmal herausbekommen. Das geht einfach: Im Windows-Konsolenfenster, erreichbar über das Startmenü, Menüpunkt „Ausführen...“ anklicken, *cmd* (Windows 2000 und XP) oder *command* (Windows 98 und Me) eintippen und *Ok* anklicken, den Befehl *ipconfig* eintippen und die Enter-Taste drü-

cken. Das Programm zeigt daraufhin die IP-Adressen aller aktiven Netzwerkadapter an. Die für Netchess zu verwendende IP-Adresse steht unter dem für das Heim-LAN verwendeten Adapter. Manche Windows-Versionen enthalten auch das Programm WinIpCFG (einfach in der Konsole eintippen), das alle Netzwerk-Daten unter einer hübschen Oberfläche zusammengefasst anzeigt.

Ports: Über eine IP-Verbindung können mehrere Datenströme gleichzeitig laufen, wie auch durch ein Telefonkabel viele Telefongespräche gleichzeitig passen. Zur Unterscheidung der Datenströme gibt es Ports, Nummern zwischen 1 und 65535, die an die IP-Adresse angehängt werden. Die Kombination aus IP und Port heißt *Socket*; z.B.: 192.168.1.1:5001

Adresse ein. EngineClient erzeugt daraufhin eine Exe-Datei, etwa „NetRuffian“, die man einfach in jedes Schachprogramm einbinden kann. Bei Fritz funktioniert das nur mit UCI-Engines (die man zunächst per „UCI-Engine erstellen“ erzeugen muss), bei Shredder Classic, Arena und ChessPartner zusätzlich auch noch mit Winboard-Engines. Es ist wirklich kein Unterschied zu lokal gestarteten Engines zu bemerken!

Es gibt aber Fallstricke: Weil Netchess jedwede Kommunikation zwischen GUI und Engine unverändert überträgt, kann es sein, dass eine entfernte Engine beispielsweise die Tablebases nicht findet. Die GUI weiß ja nicht, dass die Engine auf einem anderen Rechner läuft (und wenn sie es wüsste, wäre es ihr auch egal) und übermittelt den eingestellten Pfad zu den Tablebases, der aber nur für den

GUI-Rechner gilt, auf dem die Tablebases lagern. Wenn auf dem entfernten Rechner nicht exakt dieselbe Ordnerstruktur vorhanden ist, der Pfad zu den Tablebases also exakt identisch

```

C:\ J:\schach\Engines\engineClient.exe
engineClient (0.7-5) <C>Bernhard Wallner
Name der Engine oder <ENTER> fuer Beenden: Ruffian
Rechnername oder IP-Adresse des Zielrechners: 192.168.1.2
Portnummer fuer Verbindung: 5001
'netRuffian.exe' und 'netRuffian.rc' wurden erzeugt.
Name der Engine oder <ENTER> fuer Beenden: _

```

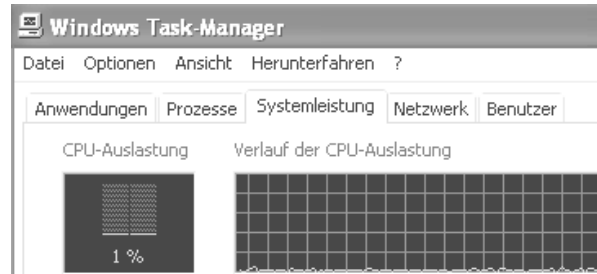
Keine Angst vor der Kommandozeile: *EngineClient* ist ganz einfach zu bedienen.

ist, findet die Engine keine Endspieldatenbanken. Dieses Problem existiert überall, wo GUI und Engine Pfadangaben austauschen und ist durch definierte und auf allen Netzwerkrechnern identische (schachbe-

zogene) Ordnerstrukturen zu lösen, wie etwa „C:\schach\tbs“. Alternativ kann man in der GUI, wenn sie mehrere TB-Pfade unterstützt, zusätzlich die Pfade auf den entfernten Rechnern angeben, wenn die Pfade unbedingt unterschiedlich sein müssen. Leider kommt man aber nicht drum herum, die Endspieldatenbanken auf jeden Rechner zu kopieren, auf dem eine Engine laufen soll.

Verteiltes Rechnen

Potenziell kann Netchess noch viel mehr als nur Engines auf anderen Rechnern laufen zu lassen, nämlich eine Schachintelligenz höherer Ordnung realisieren. Wer erinnert sich noch an das Dreihirn? Zwei voneinander unabhängige Schachcomputer rechnen an derselben Stellung. Ein Operator entscheidet, welcher der beiden vorgeschlagenen Züge wirklich gespielt werden soll. Schlagen beide Programme denselben Zug vor, gibt es nichts zu entscheiden; der Operator muss diesen Zug ausspielen. Dreihirn-Erfinder Professor Ingo Althöfer gelang der Nachweis, dass dieses System (und Variationen davon) signifikant stärker spielt als jede der Einzelkomponenten alleine.



Ruffian, unter der Fritz-GUI eingebunden, arbeitet in Wirklichkeit auf einem anderen Rechner und frisst daher auf diesem ...

... keine Rechenzeit, wie die Anzeige der Prozessorauslastung im Taskmanager beweist.

tet auf rein mathematischem Wege die verschiedenen Bewertungen und wählt dann den von den Schachprogrammen *im Mittel* für am stärksten gehaltenen Zug aus. Voraussetzung dafür ist ein ähnlicher Bewertungsmaßstab der beteiligten Programme, die sämtlich ähnliche Relationen zwischen Bauern und Figuren (z.B.: Turm ca. 4–5 Bauern) haben müssen.

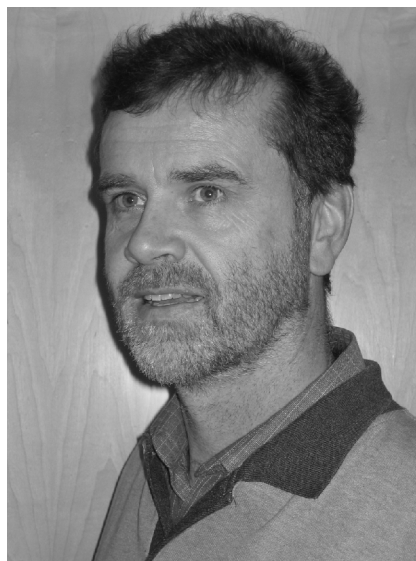
Grau aber ist alle Theorie, und ob man den menschlichen Entscheider durch eine Hand voll mathematischer Formeln adäquat ersetzen kann, muss die Praxis zeigen. Der Autor von Netchess plant tatsächlich Experimente in dieser Richtung. CSS wird über die Ergebnisse berichten.

Fazit

Netchess kann die Ausgaben verschiedener UCI- und Winboard-Engines, die auf Rechnern irgendwo im Netzwerk arbeiten, ob LAN oder Internet, zu einem zentralen Rechner transportieren. Für die darauf laufen-

de Schach-GUI verhält sich alles so, als seien es lokale Engines. Das macht den Autoplayer obsolet; er ist nur mehr gut, will man zwei ChessBase-Native-Engines auf verschiedenen Rechnern gegeneinander spielen lassen, denn diese kommunizieren mit der GUI direkt und nicht über eine (versteckte) Textausgabe. Netchess kann damit prinzipiell nicht funktionieren.

Wer also seine Rechner mit Schach auslasten möchte, sich bei Analysen für die Meinung mehrerer Programme interessiert oder als Engine-Tester eine Autoplayer-Phobie hat, sollte sich Netchess einmal ansehen. Es ist anhand der beiliegenden guten Anleitung einfach zu konfigurieren und funktioniert ohne Auffälligkeiten. Netchess gibt es gratis und mit Quelltext unter <http://home.arcor.de/bernhard.wallner/netChess.html>. Einen Link zu dieser Seite finden Sie auch bei CSS unter www.computer-schach.de/freeware.



Netchess-Autor Bernhard Wallner ist 45 Jahre alt, verheiratet und hat drei Kinder. Er arbeitet in der EDV-Abteilung der Bayerischen Vermessungsverwaltung und spielt in seiner Freizeit außer Schach gern Gitarre.