

Kapitel 3 Anforderung an die Rechenleistung

3.1 Rechenleistung

Bei der Durchführung der Versuche stellte der hohe Bedarf an Rechenleistung das größte Problem dar. Schon nach den ersten Versuchen stellte sich heraus, dass ein Rechner allein nicht schnell genug war. Um dem Rechenaufwand gerecht werden zu können, wurde die Arbeit auf bis zu sieben Computer aufgeteilt. Mit einem einzelnen Computer dauert ein Versuch im Durchschnitt 12 Stunden. Die IFN/ENIT Datensammlung [6] beinhaltet 26459 Worte. Dieser Datensammlung ist in 4 Teile geteilt. Drei wurden für das Training und ein Teil wurde für Tests benutzt.

Im ersten Experiment wurde eine einzelne Variable zwanzig mal verändert. Um diesen Versuch zu machen müssen 500000 Wörtern durchgearbeitet werden. Das ganze Experiment dauerte dadurch Tage. Am meisten wurde die Zeit des Projekt für Vorbereitung der Dateien, das Kopieren von Daten zur örtlichen Festplatte, das Erzeugen der Merkmalsvektoren und das Leiten der laufenden Arbeiten, benötigt. Dadurch war die Anwendung so genannter „batch processing software“ nützlich, um die Arbeit automatisch aufzuteilen.

3.2 Ein einfaches Perl Skript um die Arbeit auf mehrere Rechner aufzuteilen.

Am Anfang wurde ein einfaches Perl Skript geschrieben das in zwei Programmteile aufgeteilt wurde (Client- und Serversoftware). Perl wurde gewählt, da der Entwicklungszyklus sehr kurz gehalten werden kann. Das Programm arbeitet folgendermaßen:

Auf jeden teilnehmenden Rechner des Clusters läuft ein Serverskript. Die Arbeit wird mit der Client – Software geteilt. Eine Arbeitsliste wird nun vom Client geladen, geteilt und in so viele Teile aufgeteilt wie es Computer gibt. Danach wird die geteilte Arbeitsliste durch eine TCP Verbindung zu jedem Rechner geschickt. Das Serverskript nutzt den gleichen Port für die Arbeitsverteilung als auch für einen Web Server. Wenn ein Webbrowser (z.B. Mozilla, Microsoft Explorer usw.) mit diesem Port kommuniziert, wird eine HTML Seite zu dem Webbrowser geschickt. Diese Seite hat eine Liste der letzten 100 Arbeiten und der momentan laufenden Arbeiten. Diese Software reduziert den Zeitaufwand, um die Arbeit vorzubereiten. Der Quellcode befindet sich im Anhang (er wurde mit der Hilfe dem „Black Book Perl Programming“ [7] geschrieben). Die Software funktionierte fehlerfrei, außer es wird ein Rechner ausgeschaltet, denn dann geht die Arbeit auf diesem Rechner verloren. Offensichtlich muss eine stabile Lösung gefunden werden. Kostenlose *Batch Processing Software* wurde untersucht aber es gab keine Software, bei der ein Rechner zum Cluster addieren oder wegnehmen werden konnte, während das Cluster arbeitet. Deshalb wurde die selbst programmierte *Batch Processing Software* weiter entwickelt. Um dies realisieren zu können, wurde die Software auf eine Kompilersprache übersetzt. GNU C wurde wegen seiner Fähigkeit gewählt, sich mit dem Betriebssystem in Verbindung zu setzen, dass bedeutet eine optimalere Nutzung des Rechner (z.B. verbesserte Fähigkeit,

Computerlast zu handhaben).

3.3 Übersetzung der Software von Perl zu C

Die *Batch Processing Software* wurde innerhalb von drei Tagen auf C übersetzt. Nachdem es fertig war, hat es die folgenden Funktionen.

- Schutz gegen den Verlust eines Computers. (Alle Rechner außer einer können verloren gehen, die Arbeit wird jedoch nicht abgebrochen.)
- Die Fähigkeit, neue Computer in dem Cluster aufzunehmen. (Die Arbeit wird automatisch wieder geteilt)
- Die Fähigkeit, die Auslastung der Rechner zu überwachen und bei nicht optimaler Ausnutzung der Rechnerkapazitäten neue Arbeit an diesen Rechner zu senden.
- Die Computer Last ist unter zwei zu halten.
- Die Fähigkeit, mit einem Befehl alle Rechner zu bedienen, z.B. auf bestimmte Dateien zu löschen. (for i in /lnks/pyr3.0/*.*; do rm \$i -rf ;done)
- Die Fähigkeit, eine einzige Arbeit auf irgendeinem Rechner zu beenden.
- Die Fähigkeit, die Arbeit noch mal zu teilen, falls ein Rechner sehr schnell ist und wenn er seine Arbeit vor den anderen Rechner geschafft hat.
- Die Fähigkeit, den Arbeitsstand jedes Rechner in eine List zu schreiben.

Das besondere Merkmal dieses Systems ist es, dass es keinen zentralen Server gibt. Mit einem zentralen Server besteht die Gefahr, dass er ausfällt und die gesamte Arbeit verloren geht. Deshalb besitzt jeder Computer eine Kopie der Arbeitsliste. Wenn ein Befehl vom Client ausgegeben wird (z.B. eine neue Arbeit hinzuzufügen oder eine Arbeit löschen). Jeder Computer wird um seine Arbeitsliste gebeten. Eine gemeinsame Arbeitsliste wird erstellt. Bereits bearbeitete Aufträge werden von der List gelöscht, neue Arbeit hinzugefügt und die Liste an jeden Rechner geschickt. Periodisch werden alle Computer aus dem Cluster abgesucht. Wenn einer nicht antwortet wird die Arbeit dieses Rechners zu einem anderen gegeben. Wird ein neuer Rechner zu dem Cluster hinzugefügt, wird die Arbeit wieder geteilt um die Rechnerlast zu optimieren. Die Last wird niedriger als 2 gehalten, indem keine neue Arbeit anfängt, bis die Last dieses Rechners für eine Minute unter 1 fällt. Die Last wurde mit dem Kommando `sysinfo(struct sysinfo *info)` ermittelt. Alle Kommandos wurden mit einer TCP Verbindung versendet. Leider haben nicht alle Rechner im Institut Linux als Betriebssystem. Um auch diese Computer nutzen zu können wurde ein DHCP Server zusammen mit einem „Disc less Client Server“ benutzt. Mit diesen Serverprogrammen kann ein Rechner unter Linux hochgefahren werden und durch eine Netzwerkverbindung am Cluster teilnehmen. Alle Quellencodes befinden sich im Anhang.