

Vortrag zum Paper
*“Cluster of re-configurable
nodes for scanning large genomic banks“*

Arne Brutschy

Forschungsseminar Parallele Algorithmen
und Komplexe Systeme
WS 2005

Prof. Dr. Middendorf
Institut für Informatik
Universität Leipzig

Überblick

- Autoren französische Arbeitsgruppe
- Thema ist Sequenzalignment auf Gendatenbanken mittels einem Cluster von FPGAs
- Filter werden auf Hardwarebasis implementiert
- Ziele sind performantes, kosteneffizientes System
- veröffentlicht 9. Dezember 2004

Einführung

- in der Molekularbiologie werden auf großen Gendatenbanken Sequenzalignments durchgeführt
- erlaubt Rückschlüsse auf Herkunft oder Funktion unbekannter Gensequenzen
- „tägliches Brot“ vieler Biologen
- Menge der Daten steigt schneller, als die Prozessorleistung wächst (Moore's Law)
- große Nachfrage nach effizienten Lösungen

Ähnlichkeits- und Mustersuche

- Ähnlichkeitssuche vergleicht Ketten von Nukleotidsequenzen (*similarity search*)
- dient zum finden von ähnlicher Abschnitte eines Gens
- Mustersuche ist allgemeiner (*pattern search*)
- ermöglicht Suche nach Proteindomänen, vergleicht Gruppen von Aminosäuren

Smith-Waterman Algorithmus

- exakter Algorithmus zur Ähnlichkeitssuche
- Kostenfunktion zur Bewertung von Gaps, Insertions und Deletions
- ähnlich der Editierdistanz in der Sprachverarbeitung
- aufwendig, Komplexität von $O(n^2)$
- bereits 1981 zum Sequenzalignment vorgeschlagen
- heutige Verfahren approximieren Smith-Waterman

BLAST

- effizienter Algorithmus, der mittels einer Heuristik Smith-Waterman approximiert
- ca. 50-mal so schnell wie Smith-Waterman
- häufigste Anwendung in der Bioinformatik
- arbeitet dreistufig, um die große Masse der Daten vor einem komplexen Alignments ausfiltern zu können
- viele heutige Anwendungen sind Variationen davon

BLAST – Skizze

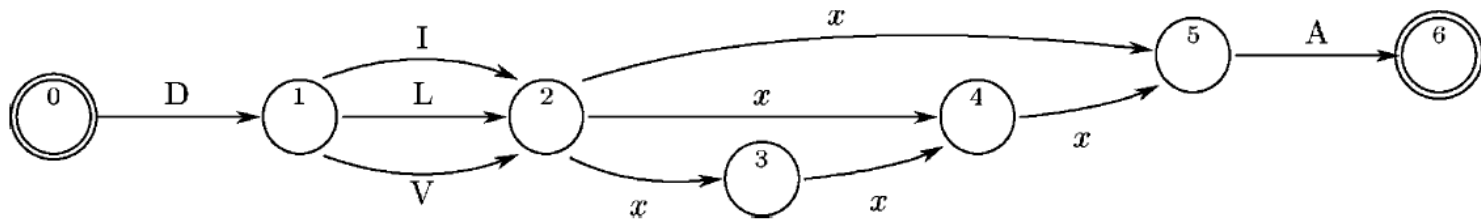
- Prinzip: kurze Wörter fester Länge haben hohe Identität einer Sequenz
- zuerst werden kurze, übereinstimmende Sequenzen gesucht
- gefundene Wörter werden erweitert um Score zu erhöhen
- Sequenzen mit einem über einem bestimmten Schwellwert liegenden Score werden in einem vollen Alignment analog zu Smith-Waterman berechnet

Mustersuche

- Beschreibungen der Muster im Internet, wie z.B. die PROSITE Datenbank
- Proteindomänen werden als reguläre Ausdrücke von Aminosäuren beschrieben
- diese lassen sich mittels gewichteter endlicher Automaten umsetzen
- Kostenfunktionen für die Ähnlichkeiten der Aminosäuren untereinander sind komplexer als bei DNA

Mustersuche – Beispiel

Muster: D - [ILV] - x(1,3) - A



Grundlegende Idee

- Problem gut zur parallelen Verarbeitung geeignet, da Tasks unabhängig voneinander
- direktes Filtern nicht relevanter Daten direkt am Datenspeicher
- Realisierung der Clusternodes als FPGA-Komponenten, die performant und kosteneffizient sind
- schneller/günstiger als übliche PC-Cluster
- geringe Ansprüche an Netzwerk und Hostcomputer

RDISK-Node

- Nodes werden aus günstigen Standardkomponenten aufgebaut (Festplatte, Ram, FPGA, Ethernetcontroller etc...)
- FPGA übernimmt alle wichtigen Funktionen wie Ansteuerung der Festplatte, Dateisystem und Protokoll-stack
- Einschübe im PCB-Format, um bis zu acht Nodes pro Rackebene erlauben
- Kosten unter 200 \$

Ansicht eines RDISK-Nodes



RDISK-48 Cluster

- 48 RDISK-Nodes werden über Ethernet verknüpft
- Datenbank gleichmäßig über Nodes verteilt
- diese reichen statistisch signifikante Hits an den Hostprozessor weiter
- das Postprocessing kann komplett auf einem handelsüblichen PC stattfinden
- durch hohe Selektivität der Filter wird der Flaschenhals Netzwerk umgangen

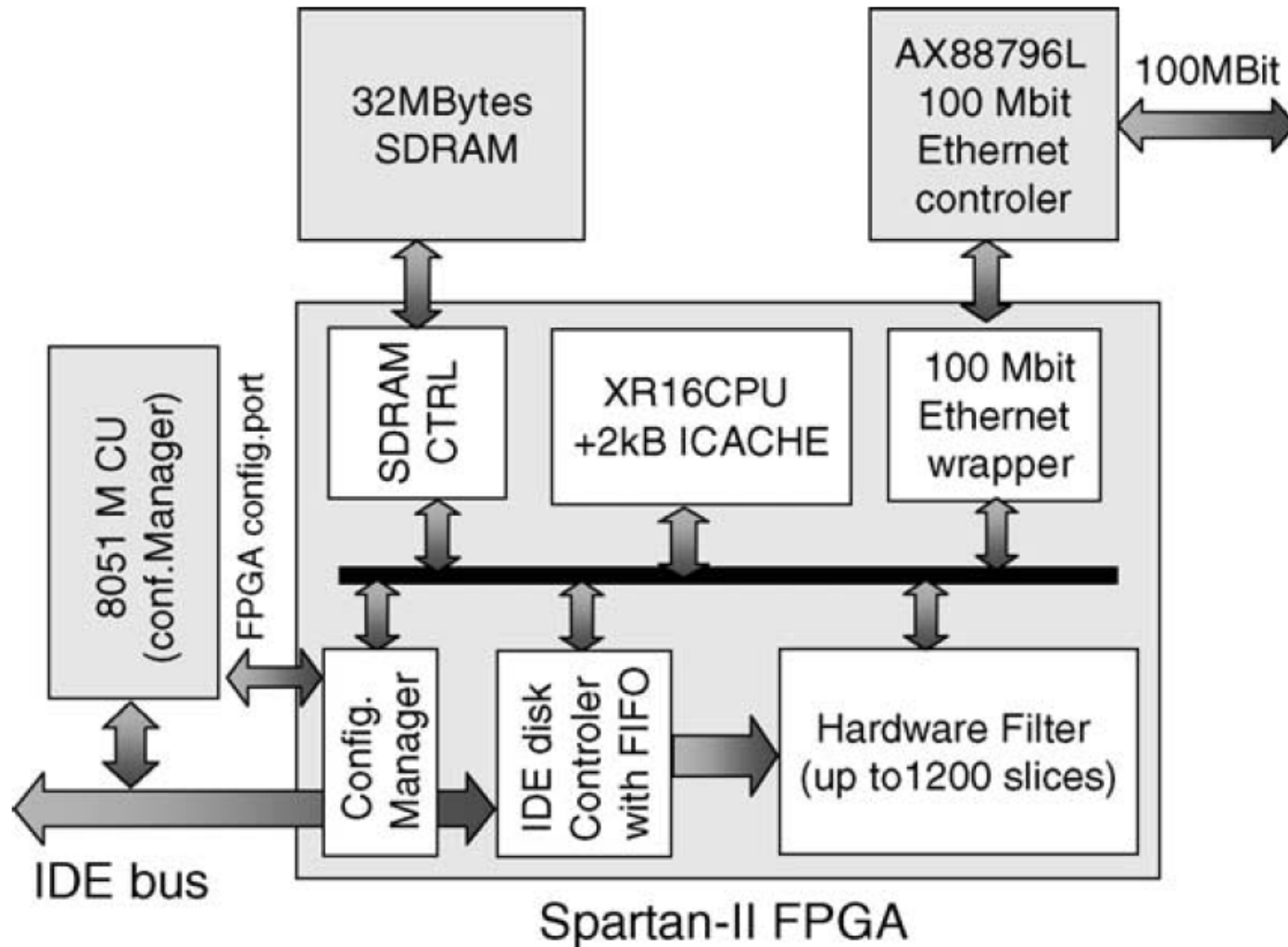
System on a Chip

- komplettes System wird im FPGA verwirklicht
- übernimmt Steuerungsfunktionen der Festplatte, Netzwerk, Filter und Rekonfiguration
- implementiert einfaches Dateisystem zur Ablage von Datenbanken und Konfigurationen
- lässt 3000 LUTs und 5 kb Speicher zur Implementierung von Filtern übrig

System on a Chip – Operationen

- speichern von Dateien oder Konfigurationen auf der Festplatte
- Rekonfiguration des FPGAs
- Start einer Suche
- bei Initialisierung wird ein Basissystem von der Festplatte geladen
- dient auch als Fallback-System in Fehlerfällen

System on a Chip – Struktur



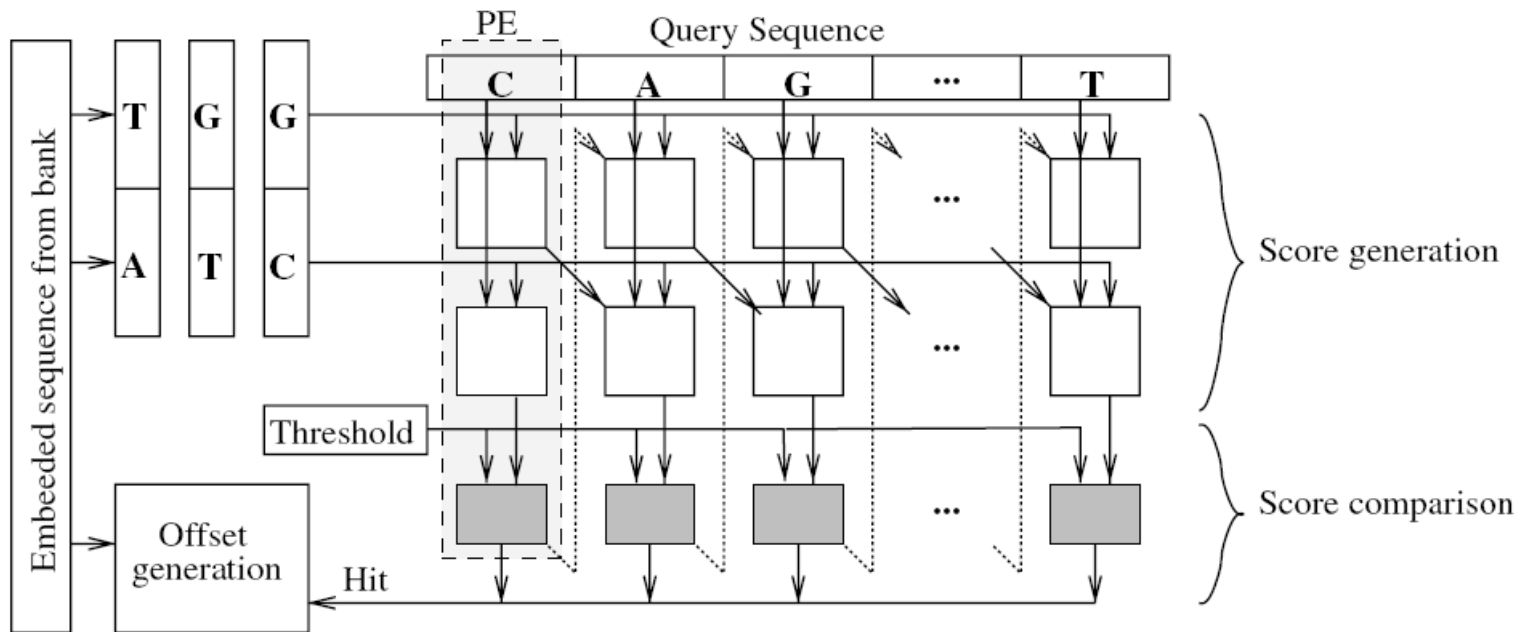
Implementierungsschritte

- Filterdefinitionen werden in VHDL geschrieben und nach Synthese mit dem Basissystem gemerged
- das Postprocessing wird in C mit Hilfe von bereitgestellten Bibliotheken geschrieben
- die Filterimplementierung wird dann auf die Nodes geladen, welche sich damit rekonfigurieren
- das Postprocessing wird auf dem Hostcomputer eingebunden
- die Autoren stellen Templates und Bibliotheken zur Verfügung
- Entwicklung neuer Applikationen wird mit ca. einem Mannmonat veranschlagt

Implementierung – Ähnlichkeitssuche

- die ersten beiden Stufen von BLAST, das Finden statistisch signifikanter Hits, wird in den Nodes erledigt
- das Postprocessing berechnet das volle Alignment nach Smith-Waterman
- im FPGA werden sogenannte „processing elements“ (PEs) implementiert, die jeweils ein Nukleotid der Suchsequenz aufnehmen
- Vergleich findet ohne Gaps statt
- Länge der unterstützten Suchsequenz abhängig vom Schwellwert

Implementierung – Ähnlichkeitssuche (2)

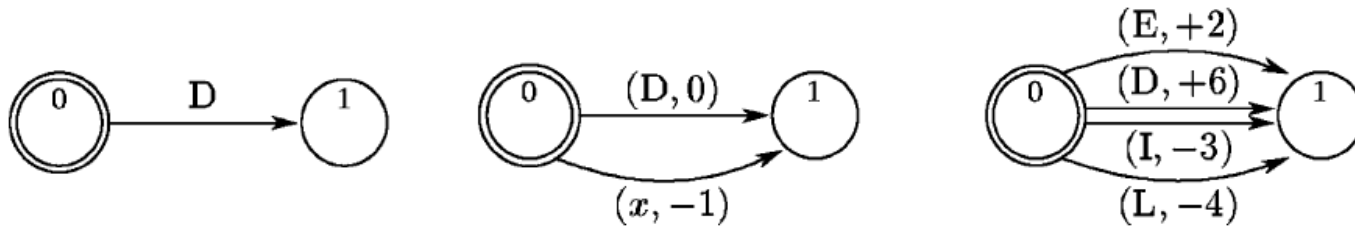


Schwelwert	programmierbar	16	18	20	22	30
max. Sequenzlänge	137	198	180	160	159	157

Implementierung – Mustersuche

- Automat wird linear implementiert, jeder Zustand wird ein Register, jede Transition ein Addierer und Komparator
- komplexere Muster erzeugen größere Automaten
- kleine Automaten können zur Erhöhung der Geschwindigkeit mehrfach implementiert werden
- für jedes Suchmuster muss eine eigene Konfiguration erzeugt werden

Implementierung – Mustersuche (2)



Automat		Bandbreite		max. Suchsequenz			
Einheiten	LUTs	pro Cycle	bei 40 Mhz	p=6	p=8	p=10	p=12
x 1	7p	1 bp	40 Mbps/s	71	53	42	35
x 2	14p	1 bp	40 Mbps/s	35	26	21	17
x 3	15p	3 bp	120 Mbps/s	33	25	20	16
x 6	30p	3 bp	120 Mbps/s	16	12	10	8

Testresultate

- bei der Ähnlichkeitssuche ist ein einzelner Node ungefähr dreimal so schnell wie ein handelsüblicher Prozessor
- die Filter können die Daten bei voller Festplattenbandbreite von 60 Mbp/s verarbeiten
- bei der Mustersuche ist der RDISK-48 Cluster bei der reinen Berechnung äquivalent zu einem PC-Cluster mit 192 Nodes
- die Rekonfiguration der Nodes verbraucht dabei allerdings sehr viel Zeit

Testresultate (2)

Art der Suche	Anfrage	Anzahl der Hits	Ausführungszeit (sec)					Ethernet Bandbreite
			T_b	T_c	T_s	T_{SW}	T	
similarity	Line	5449	0,0	0,8	3,0	38,2	39	29,0 KB/s
similarity	Globin	329	0,0	0,8	3,0	1,6	3,8	1,7 KB/s
similarity	Prion	66	0,0	0,8	3,0	0,3	3,8	0,3 KB/s
similarity	Random	2	0,0	0,8	3,0	0,1	3,8	0,0 KB/s
pattern	OR1	20985	311	0,8	17,7	0,0	330	9,5 KB/s
pattern	OR2	13499	220	0,8	17,7	0,0	239	6,1 KB/s
pattern	OR3	17469	241	0,8	17,7	0,0	260	7,9 KB/s
pattern	OR4	24178	235	0,8	17,7	0,0	244	10,9 KB/s
pattern	OR5	21692	287	0,8	17,7	0,0	306	9,8 KB/s

Vorteile des Systems

- Hardwarefilter erlauben hohe Selektivität
- geringe Kosten bei hoher Leistungsfähigkeit
- Datenbanken können mit maximaler Festplattenbandbreite durchsucht werden
- durch geringe inter-node Kommunikation genügen einfache Netzwerkkomponenten
- effektive Filter vereinfachen das Postprocessing

Nachteile des Systems

- Implementierung neuer Applikationen komplex, aufwändig und damit teuer
- Aufgrund geringer Netzwerkleistung dauern Updates der Datenbank/Konfigurationen sehr lange
- Ähnlichkeitsfilter sind durch fest verdrahteten Schwellwert unflexibel
- Automaten der Suchmuster müssen jedes Mal erneut synthetisiert und übertragen werden
- Leistung im praktischen Einsatz durch Beschränkung der Filtergrößen oft wesentlich geringer

Kritik am Paper

- Tests teilweise wenig aussagekräftig
- es können nur begrenzt komplexe Suchmuster implementiert werden, eine Lösung wird nicht besprochen
- Filter bei der Ähnlichkeitssuche nicht äquivalent zu BLAST, hohe Schwellwerte erzeugen viele nicht gefundene Matches
- dies wird kaum analysiert, quantitative Aussagen fehlen
- Argumentationen manchmal nicht belegt

Quellen

- S. Guy´etant, M. Giraud, S. Derrien, L. L’Hours, S. Rubini, D. Lavenier, F. Raimbault: *Cluster of reconfigurable nodes for scanning large genomic banks* Par. Comp. 31(1), 73-96 (2005).
- T. Smith, M. Waterman: *Identification of common molecular subsequences* J. Mol. Biol. 147, 195-197 (1981).
- S. Altschul, W. Gish, W. Miller, E. Myers, D. Lipman: *Basic local alignment search tool* J. Mol. Biol. 215(3), 403-410 (1990).
- D. Lavenier, S. Guy´etant, S. Derrien, S. Rubini: *A reconfigurable parallel disk system for filtering genomic banks* ERSA’03, Engineering of Reconfigurable Systems and Algorithms, Las Vegas, Nevada, USA (2003).
- M. Giraud, D. Lavenier: *Linear encoding scheme for weighted finite automata* Lecture Notes in Computer Science 3317, 146-155 (2004).





13.12.2005



29