

Deep Blue



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Knowledge Engineering & Machine Learning in Games

Vortrag von Olexandr Savchuk

Übersicht



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Grundlagen der Schachcomputer
 - Architektur
 - Allgemeiner Aufbau
 - Schachchips
 - Algorithmen
 - Softwaresuche
 - Parallele Suche
 - Erweiterte Elemente
 - Performanz
 - Schluss



Grundlagen der Schachcomputer



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Grunddesign von C. Shannon (1949)
 - Drei Hauptkomponenten
 - *Zuggenerator*
 - *Evaluierungsfunktion*
 - *Suchkontrolle*
 - Minimax-Suche

- Erweiterungen bis in die 1970er
 - Slate and Atkin's Chess 4.5
 - *Quiescence search*
 - *Alpha-beta pruning algorithm*
 - *Lazy evaluation*



Grundlagen der Schachcomputer



- *Quiescence search*
 - Löst das Horizontproblem
 - Sucht bei „interessanten“ Positionen tiefer
- *Alpha-beta pruning*
 - Betrachte nicht alle möglichen Reaktionen
 - Verwerfe einen schlechten Zug, sobald **eine** effektive Reaktion gefunden wurde
 - *Wichtig:* Suchbaum muss „best-first“ angeordnet sein
- *Lazy evaluation*
 - Annäherung an die Evaluierungsfunktion in vielen Fällen ausreichend und wesentlich schneller

Übersicht



TECHNISCHE
UNIVERSITÄT
DARMSTADT

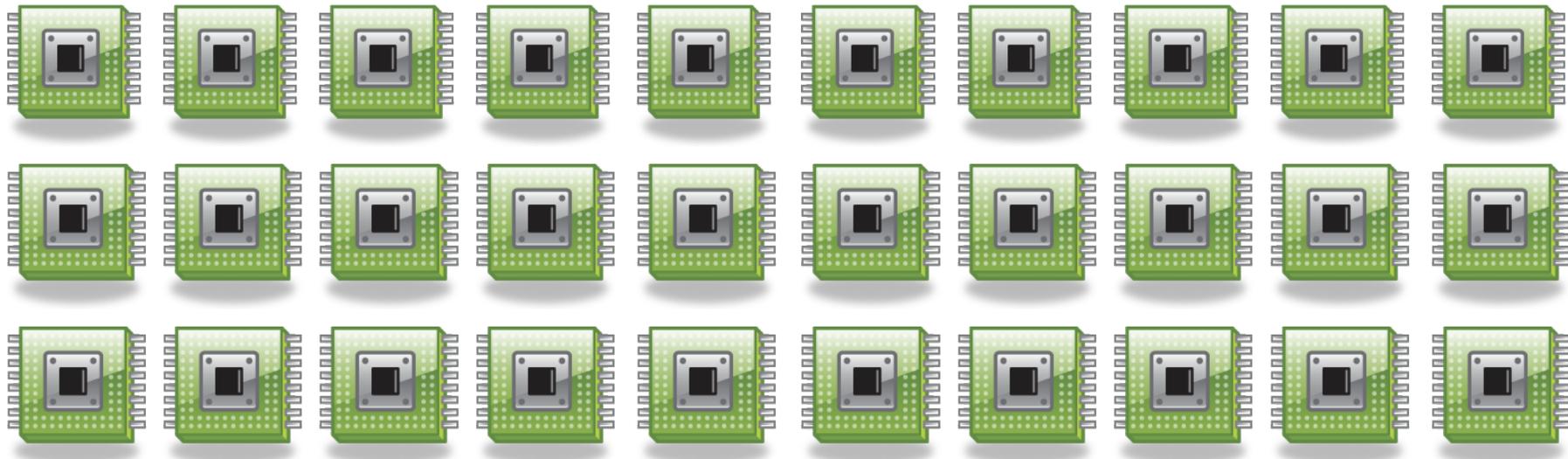
- Grundlagen der Schachcomputer
- Architektur
 - Allgemeiner Aufbau
 - Schachchips
- Algorithmen
 - Softwaresuche
 - Parallele Suche
- Erweiterte Elemente
- Performanz
- Schluss



Architektur



TECHNISCHE
UNIVERSITÄT
DARMSTADT

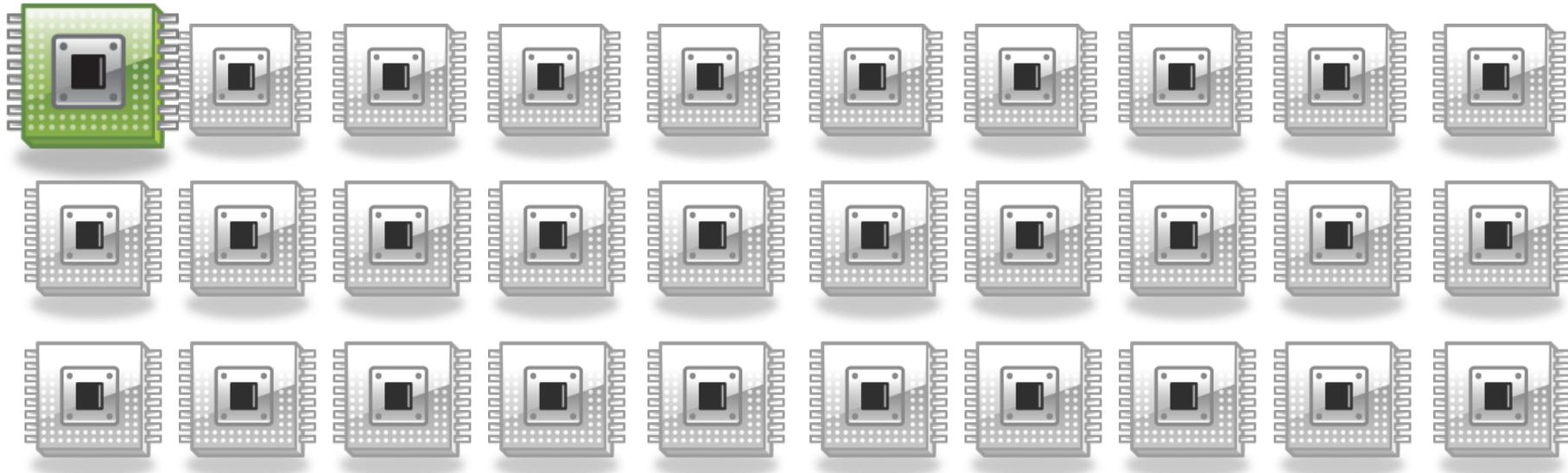


- Basiert auf einem IBM RS/6000 System
- **30** RS/6000 Einheiten, verbunden durch Micro Channel Bus

Architektur



TECHNISCHE
UNIVERSITÄT
DARMSTADT

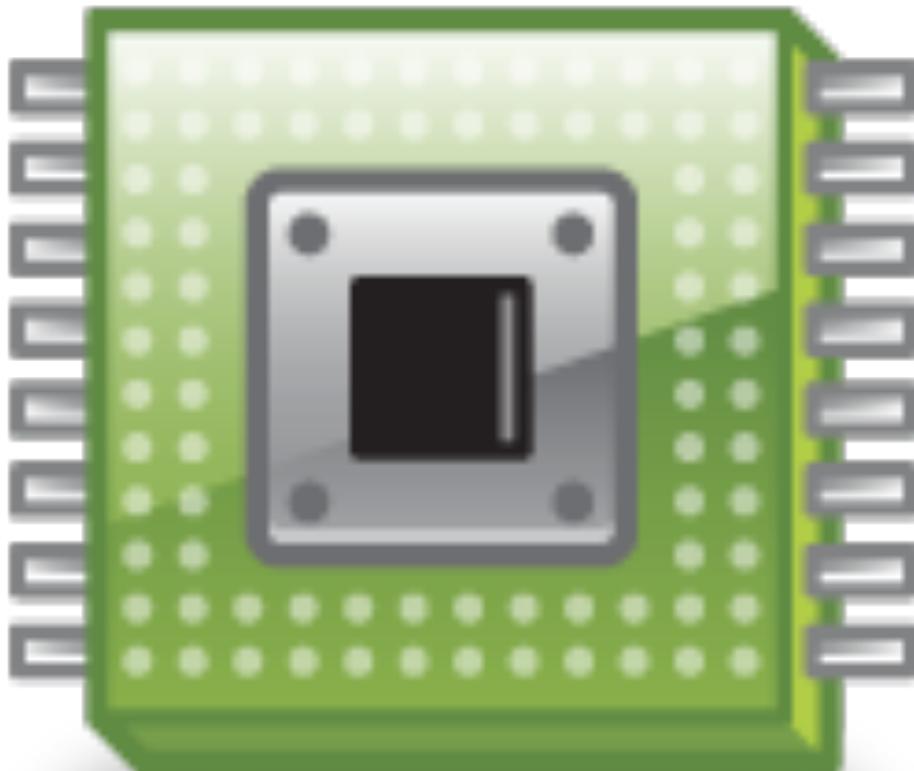


- Ein *Master Node*, 29 *Worker Nodes*



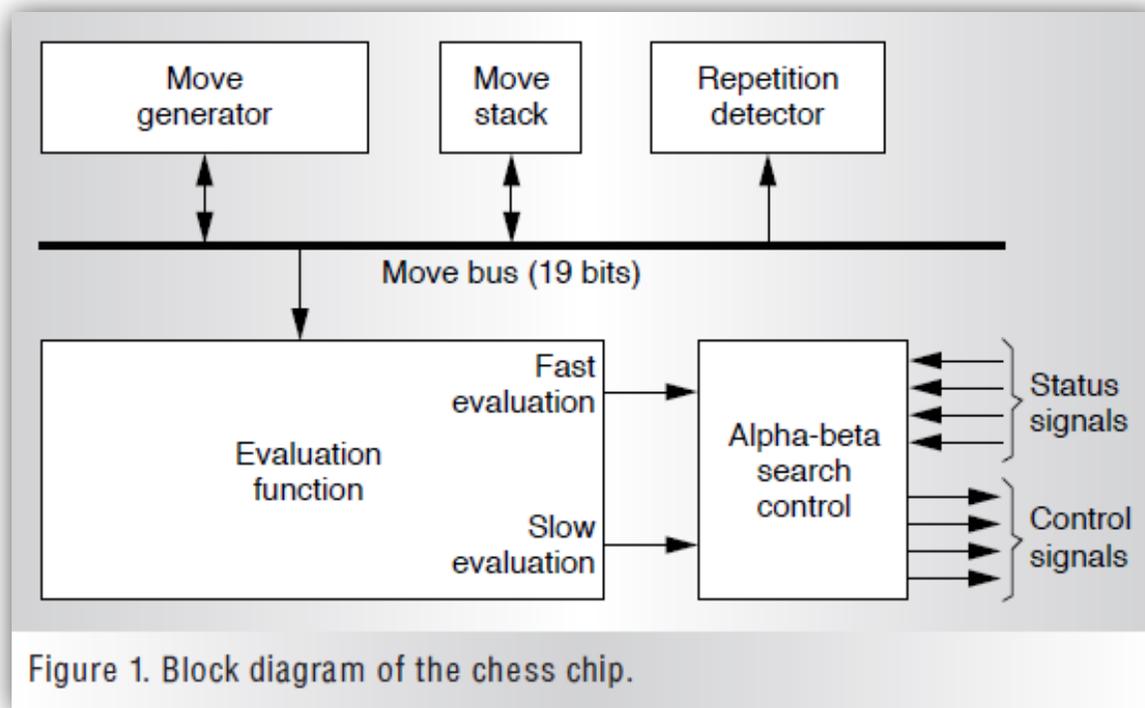
Architektur

- 16 Schachchips auf jedem Node



Architektur

- 16 Schachchips auf jedem Node
- Vier Elemente auf Chip:
 - *Zuggenerator*
 - *Evaluierungsfunktion*
 - *Suchkontrolle*
 - *Smart-move stack*



Zuggenerator

- 8x8 Array von kombinatorischer Logik
 - ca. 52400 Gatter
- Jede Zelle beinhaltet:
 - *Find-victim transmitter*
 - *Find-attacker transmitter*
 - *Receiver*
 - *Distributed arbiter*
- Verschiedene Züge durch Beschalten der Elemente

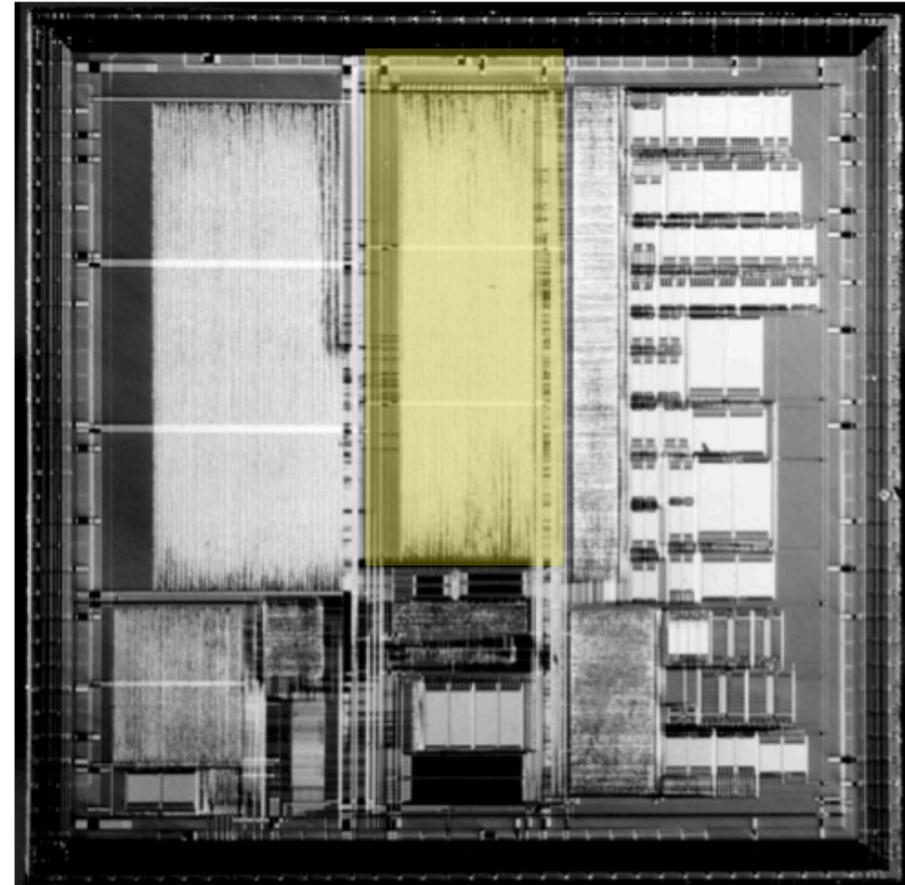


Figure 2. Die photo of the chess chip.

Zuggenerator: Beispiel

- Alle Angreifer aktivieren den *Find-victim Transmitter*
- Es wird abgestimmt, welches Ziel angegriffen wird
 - *Receiver* der Ziele versenden „Treffersignale“ an den *Arbiter*
 - *Arbiter* wählt das höchstwertige Ziel aus
- Das gewählte Ziel aktiviert den *Find-attacker Transmitter*
 - Auf die gleiche Weise wird der niedrigwertigste Angreifer ausgesucht
- Wenn Angreifer und Ziel feststehen, ist ein Zug generiert

Zuggenerator



- Bereits generierte Züge werden maskiert und nicht erneut generiert
- Andere Generationsmodi werden unterstützt
 - Schachgebot
 - Schachabwehr
 - ...
- Erkennt bedrohte Figuren

Evaluierungsfunktion

- 66000 Gatter (ohne RAMs und ROMs)
- Zwei Hauptelemente:
 - Schnelle Evaluation
 - Langsame Evaluation
- Gewichte der Elemente anpassbar

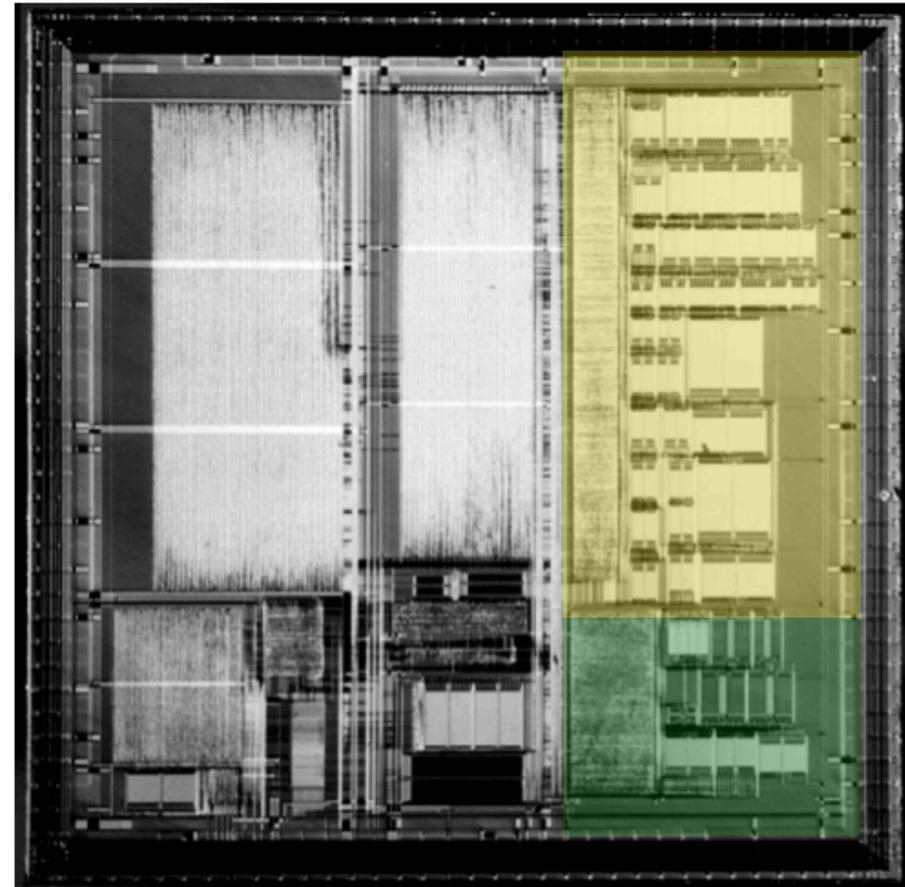


Figure 2. Die photo of the chess chip.

Schnelle Evaluation

- Wird in einem Zyklus berechnet
- *Piece placement table*
 - Werte von Figuren auf ihren Feldern
- *Game phase control*
 - Boni/Mali abhängig von der Spielphase
- *King-and-pawn array*
 - Erkennung von Freibauern und dazugehörigen Zuständen
- *Engdame logic and ROMs*
 - Diverse Logik zur Erkennung von Endspielsituationen

Langsame Evaluation

- Wird selten gebraucht
- „This chess evaluation function probably is more complicated than anything ever described in the computer chess literature“²

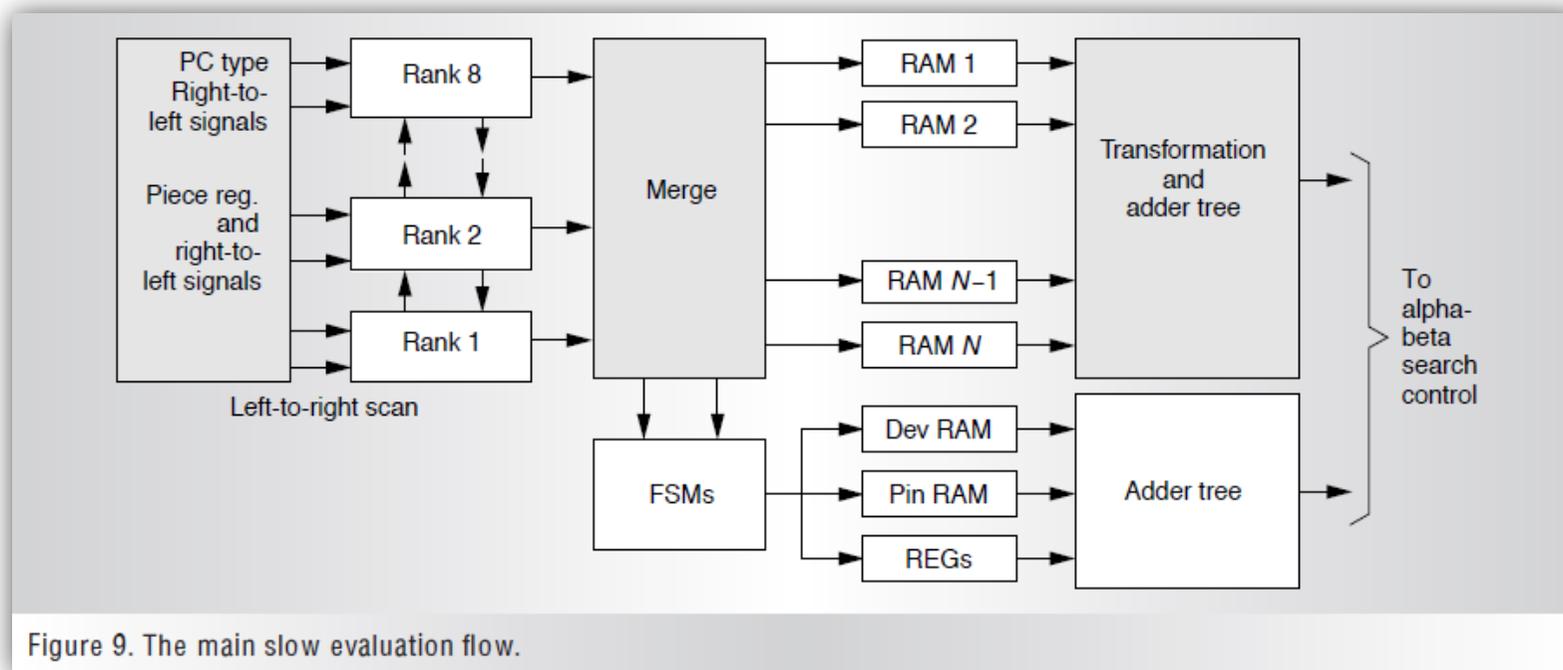


Figure 9. The main slow evaluation flow.

Suchkontrolle



- Implementiert einen *minimum-window alpha-beta search*
- Unterschied zum „klassischen“ Alpha-Beta:
 - Kein Alpha-Beta Stack
 - Kann nur bestimmen, wie eine Position im Vergleich zu einem Testwert steht
- Ungewöhnlicher Element: low-pass filter
 - Letzter Wert der langsamen Evaluation als Filterwert



Smart-move stack



- Besteht aus dem *Move Stack* und *Repetition Director*
- *Repetition Director* erkennt wiederholende Züge sowie Züge, die zu Wiederholungen führen



Übersicht

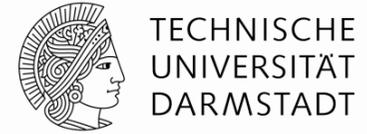


TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Grundlagen der Schachcomputer
- Architektur
 - Allgemeiner Aufbau
 - Schachchips
- **Algorithmen**
 - Softwaresuche
 - Parallele Suche
- Erweiterte Elemente
- Performanz
- Schluss



Suchalgorithmus



- Aufteilung in Software und Hardware

- Beispiel: 12-Schichten Suche
 - Erste 4 Schichten in Software auf dem Master-Node
 - Weitere 4 Schichten in Software auf den Worker-Nodes
 - Letzte 4 Schichten in Hardware auf den Worker-Nodes

- Vorteile solcher Aufteilung:
 - Flexibilität:
 - Software kontrolliert 2/3 der Suchtiefe
 - Viele Sucherweiterungen in den Softwareschichten
 - Geschwindigkeit:
 - Hardware berechnet >99% aller Positionen



- Implementierung des Alpha-Beta Algorithmus
- Erweiterung der Suche entlang der *forced move* Pfade
- *Dual Credit Algorithm*
 - Suchpfad sammelt *credit* auf (für beide Seiten)
 - *credit* wird für Erweiterung der Suchtiefe „eingelöst“
 - *credit* wird für unterschiedliche Beobachtungen im Suchpfad vergeben, zum Beispiel:
 - Einer oder wenige Züge sind signifikant besser als alle Anderen
 - Bedrohung des Schachs oder Schachmatts
 - Ermöglichung guter neuer Züge
 - ...

Hardwaresuche



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Sehr schnell
- Relativ einfach im Vergleich zur Softwaresuche
- Gesteuert durch die Suchkontrolle auf den Schachchips
- Software kann einige Parameter ansteuern:
 - Suchtiefe
 - Anzahl der Schachmatt-Züge für beide Seiten
 - Anzahl der singulären Schachgebot-Züge
 - Flags, die Sucherweiterungen in bestimmten Situationen erlauben
 - ...



Übersicht



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Grundlagen der Schachcomputer
- Architektur
 - Allgemeiner Aufbau
 - Schachchips
- Algorithmen
 - Softwaresuche
 - Parallele Suche
- **Erweiterte Elemente**
- Performanz
- Schluss



Eröffnungsbuch



- Erstellt per Hand vom Großmeister Joel Benjamin
- 4000 Positionen
- Alle vom Deep Blue durchgerechnet
- Erweiterungen vor dem Spiel durch ein *Override book* möglich

Erweitertes Buch



- Große Datenbank der Positionen aus Großmeisterspielen
 - Ca. 700.000 Spiele verarbeitet
- Beeinflusst die Auswertung einzelner Züge
 - Wie oft ein Zug gespielt wurde (absolut und relativ)
 - Wie stark die Großmeister waren, die den Zug verwendet haben
 - Ergebnisse des Zuges
 - Kommentare zu den Zügen
- Manchmal sogar Züge aus dem erweiterten Buch direkt spielbar



Endspieldatenbanken



- Alle Positionen mit 5 oder weniger Spielfiguren und auserwählte Positionen mit 6 Figuren
- Stehen der Software zur Verfügung
 - Positionen mit 4 oder weniger sowie einige mit 5 auf den einzelnen Festplatten der Nodes dupliziert
 - Andere Positionen zentral von einem RAID allen Nodes verfügbar
- Simple Werte zu allen Positionen gespeichert (win-lose)

Übersicht



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Grundlagen der Schachcomputer
- Architektur
 - Allgemeiner Aufbau
 - Schachchips
- Algorithmen
 - Softwaresuche
 - Parallele Suche
- Erweiterte Elemente
- **Performanz**
- Schluss



Performanz als Rechner



- 30 RS-6000 Nodes
 - P2SC Prozessoren (28x 120Mhz, 2x 135MHz), 1GB RAM, 4GB HDD
 - OS: AIX® 4.2
- Sehr spezialisierte Hardware
 - Ein Schachchip berechnet bis 2.5 Millionen Positionen pro Sekunde
 - Auf einem „normalen“ Computer: 40.000 Instruktionen pro Position
 - 100.000 MIPS
- Vergleich:
 - 1997, PowerPC 750: 88 MIPS @ 233MHz
 - 2010, Intel Core i7 i980EE: 147.600 MIPS @ 3.3GHz



Performanz als Schachspieler



- Erste Spiele mit einem Chip (Deep Blue Jr., 1997)
 - 7% bis 14% der möglichen Suchgeschwindigkeit
 - Alle 10 Spiele gegen kommerzielle Schachprogramme gewonnen
 - Bei weiteren 30 Spielen: 95% Siege
- Deep Blue Jr. gegen die Großmeister (Elo-Zahl 2500)
 - 3-zu-1 Verhältnis
 - Somit Wertung von 2700 – unter den 10 besten Spielern der Welt
- Spiel gegen Kasparow (Elo 2815)
 - Sieg mit 3.5 - 2.5
 - Elo-Zahl von Deep Blue: ca. 2875.



Schluss

- Deep Blue war der schnellste Schachcomputer, der jemals gegen einen Großmeister gespielt hat
 - 200.000.000 Positionen pro Sekunde
- Heutzutage:
 - Fokus von der Hardware auf die Software versetzt
 - Geringere Suchgeschwindigkeit
 - Deep Fritz vs. Kramnik (2006), „nur“ 8.000.000 Positionen/Sekunde
 - Pocket Fritz 4: *HTC Touch HD*, 20.000 Positionen/Sekunde
 - Stärkere *pruning*-Algorithmen und weitere Verbesserung der Heuristiken



Vielen Dank für die Aufmerksamkeit!

Fragen?

Quellenangabe

- Hauptquellen:
 - 1: **Deep Blue** (M. Campbell, A. J. Hoane Jr, F. -h. Hsu)
 - 2: **IBM's Deep Blue Chess Grandmaster Chips** (Feng -hsiung Hsu), *IEEE Micro* 19(2):70-81

- Zusätzliche Quellen:
 - **The Alpha-Beta Heuristic (AIM-030)** (Edwards, D.J. and Hart, T.P., Massachusetts Institute of Technology)
 - **A Brief History of RISC, the IBM RS/6000 and the IBM eServer pSeries** (IBM, <http://www-03.ibm.com/ibm/history/documents/pdf/rs6000.pdf>)
 - **Overclock3D Synthetic Benchmarks** (http://www.overclock3d.net/reviews/cpu_mainboard/intel_980x_gulftown/4)