

# High Performance Computing

Kristian Rietveld (krietvel@liacs.nl, kamer 138)



Universiteit Leiden  
The Netherlands

Discover the world at Leiden University.

# Groep Computer Systems

- High-Performance Computing
  - Optimizing compilers (generieke codes, maar ook specifieke rekenkernels).
  - Parallel & Distributed Processing.
  - Optimaliseren op bestaande hardware.
- Embedded Systems
  - Specifieke software mappen op specifieke hardware.
  - Hardware synthesis.
  - Real-time scheduling.

# High Performance Computing

- Een computersysteem heeft een eindige set resources.
  - Processing power, geheugen, disk space ...
- Hoe zorgen we ervoor dat we het maximale eruit halen wat erin zit?

# Waar optimaliseren?

- CPU:
  - Zo weinig mogelijk stalls / idle.
  - Maximize cache hits, striding, prefetching, ...
- Memory:
  - Bandwidth, space, ...
- Disk:
  - Seek time, bandwidth, ...
- Interconnect/network
  - Bandwidth, latency, ...
- GPGPU:
  - Parallelism, coalescing, bank conflicts, overlap compute/transfer.

# High Performance Computing

- Onze missie is het onderzoeken, analyseren en optimaliseren van computer- en softwaresystemen.
- Hieruit volgen nieuwe technieken.
- We laten zien dat deze technieken effectief zijn.

# Gebieden

- In wat voor gebieden zijn we actief?
  - Optimizing Compilers.
  - Sparse matrix computations.
  - Parallel & distributed computing.
  - Database systems & data integration.
- Hardware / experimentatieplatformen
  - DAS clusters: DAS4, DAS3.
  - LGM: cluster van GPUs.
  - Leiden Zipper.

# Vakken

- Vakken gerelateerd aan onze groep:
  - Computer Architectuur
  - Compiler Constructie
  - Operating Systems
  - Netwerken
  - Advanced Compilers and Architectures
  - Parallel Programming

# Concreet voorbeeld

- Bij computationele mechanica worden zeer veel stelsels van vergelijkingen opgelost (lineaire algebra).
- De CPU besteedt veel tijd in een klein aantal loops.
- Kunnen we ervoor zorgen dat deze loops in minder tijd kunnen worden uitgevoerd?



# Concreet voorbeeld (2)

- Enige opties:
  - Data in goede volgorde in geheugen plaatsen.
  - Zorgen dat data die niet wordt gebruikt niet in de CPU cache terecht komt.
  - Loop anders organiseren zodat data in de cache kan worden hergebruikt.
  - Loop opsplitsen en in parallel draaien.
  - ...

# Concreet voorbeeld (3)

- Uit het optimaliseren van zulke “rekenkernels” volgen nieuwe technieken.
- Deze kunnen in een generieke compiler worden geïmplementeerd.
- Effect van deze optimalisatie op andere codes?

# Projecten



# Opzet

- Deel theorie.
  - Academische literatuur.
  - Technische documentatie/specificatie.
- Deel implementatie.
- Deel experimentatie.

# Projectgebieden

- Parallel & Distributed Computing
- Optimizing Compilers (gcc, LLVM/Clang)
  - Optimalisaties
  - Programmatransformatie
  - Compiler Plug-ins
  - Code generatie
- Operating Systems
- Computernetwerken

# Projectvoorstel

- Op het moment onderzoeken wij een raamwerk voor het afleiden en aanpassen van algoritmen met compilertransformaties.
- Hierdoor ontstaan er vele varianten van algoritmen.
- Het is nodig inzicht te krijgen welke varianten wel en niet efficiënt zijn.

# Projectvoorstel (2)

- Parallellisatie & benchmarking van simpele algoritmen.
- Denk aan: matrix algebra, sorteren, graafalgoritmen als Dijkstra.
- Ofwel distribueren op cluster, of paralleliseren op CPU/GPU. (Of beide?)

# Projectvoorstel (3)

- Kan gebeuren op verschillende niveaus, je kan kiezen wat je het interessants vindt:
  - Syteemniveau: “low-level” transformaties om de code te optimaliseren; algoritme zelf verandert eigenlijk niet.
  - Algoritmisch niveau: transformaties die het algoritme licht aanpassen (bijv. stappen algoritme op andere manier uitvoeren).
  - Fundamenteel niveau: transformaties in het specificatieraamwerk, waar nog geen code wordt gegenereerd.



# Projectvoorstel (4)

- Verwachte vaardigheden:
  - Programmeervaardigheden en abstract denken afhankelijk van niveau.
  - Systemniveau: veel programmeervaardigheid, weinig abstract denken.
  - Fundamenteel niveau: weinig programmeervaardigheid, veel abstract denken.

# Projectvoorstel (5)

- Opzet project:
  - Deel theorie: omschrijving transformaties.
  - Deel implementatie.
  - Deel experimentatie & conclusies trekken.
- Keuze algoritme & abstractieniveau in overleg.

# Andere Projecten

- Andere projecten ook mogelijk; eigen ideeën zijn welkom!
- Als je dit een interessante vakgebied lijkt, kom eens langs om te praten.

Bedankt voor jullie aandacht!

Interesse? E-mail: [krietvel@liacs.nl](mailto:krietvel@liacs.nl)  
Of kom langs in kamer 138.