

# High Performance Computing

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



Universiteit Leiden  
The Netherlands

Discover the world at Leiden University.

# Groep Computer Systems

- Embedded systems
  - Specifieke software mappen op specifieke hardware.
  - Hardware synthesis.
  - Real-time scheduling.
- High-Performance Computing
  - Optimizing compilers (generieke codes, maar ook specifieke reken kernels).
  - Cluster computing.
  - Parallelisatie/vectorisatie.
  - Zitten vast aan bestaande hardware.

# 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?

# Voorbeelden

- CPU:
  - Zo weinig mogelijk stalls / idle.
  - Cache hits, striding, prefetching, ...
- Memory:
  - Bandwidth, ruimte, ...
- Disk:
  - Seek time, bandwidth, ...
- Interconnect/network
  - Bandwidth, latency, ...
- GPU:
  - 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.
  - Database systems & data integration.
  - Parallel & distributed computing.
- Hardware
  - LGM: cluster van GPUs.
  - DAS4, DAS3.
  - Leiden Zipper.

# Voorbeeld

- Software bestaat vaak uit meerdere “lagen”.
- Bijvoorbeeld een iPhone applicatie:
  - Operating System
  - C library
  - Objective-C runtime
  - OpenGL implementatie
  - CoreGraphics
  - Widget library
  - Eigenlijke applicatie code

# Voorbeeld (2)

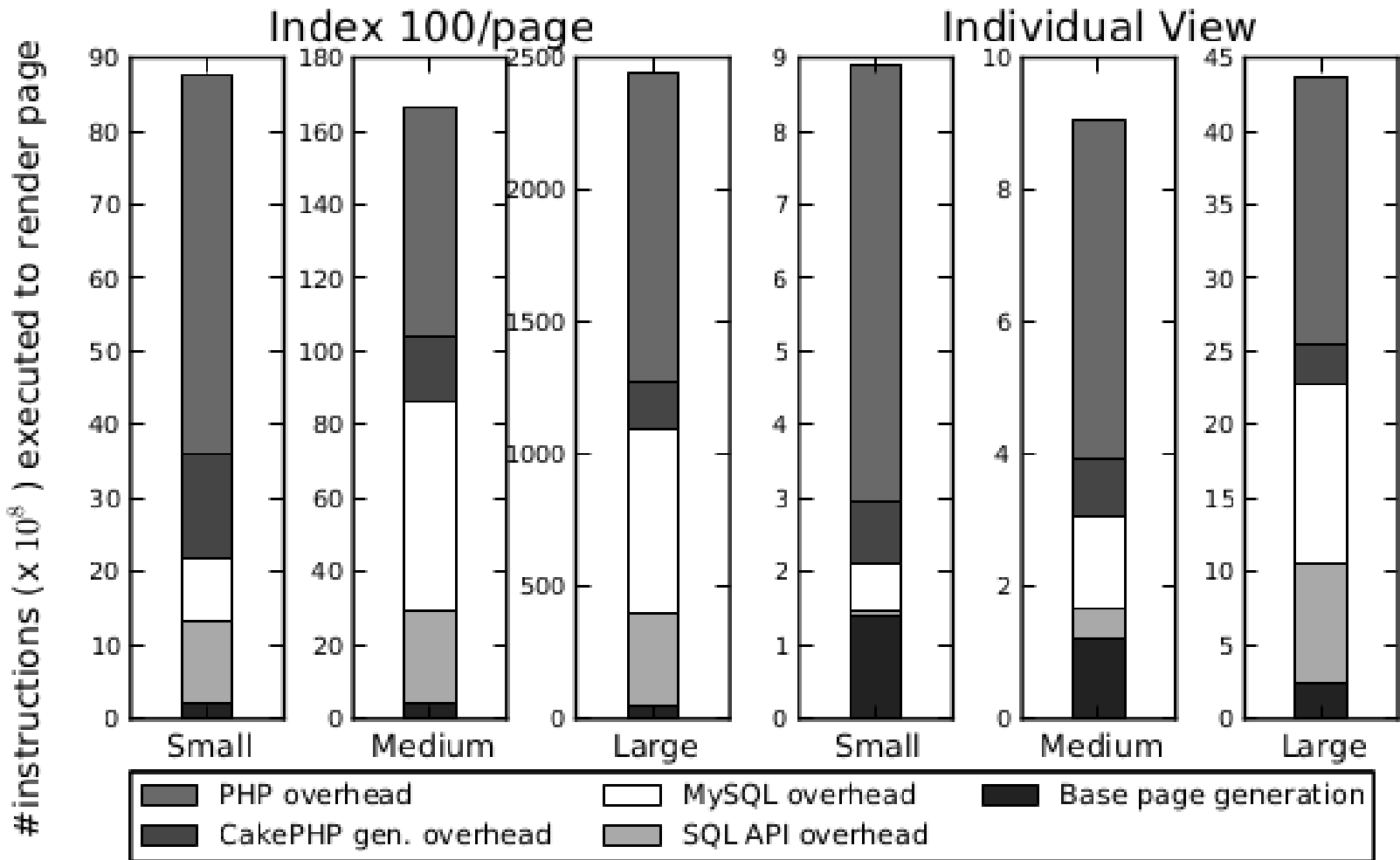
- Web applicatie:
  - Script taal (e.g. PHP)
  - DBMS (e.g. MySQL)
  - Development framework (e.g. CakePHP)
  - Eigenlijke applicatie code



# Voorbeeld (3)

- Web applicatie:
  - Script taal (e.g. PHP)
  - DBMS (e.g. MySQL)
  - Development framework (e.g. CakePHP)
  - Eigenlijke applicatie code
- We hebben bekeken wat de overhead van deze lagen is, door deze lagen te elimineren.

# Voorbeeld (4)



# Voorbeeld (5)

- Vervolgens hebben we compilertechnieken ontwikkeld om deze lagen automatisch te elimineren.
  - PHP: vertaling naar C++, bestaand project.
  - DBMS: compilerplugin om MySQL calls te herkennen en te herschrijven.
  - SQL API: applicatie & data code binnen zelfde intermediate transformer.

# Projecten

- Geen lijst openstaande projecten.
- Kader waarbinnen projecten mogelijk zijn.
- Als je dit een interessante hoek lijkt, kom eens langs om te praten!

# Vakken

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

# Projectgebieden

- Compilers (LLVM, Clang)
  - Optimalisaties
  - Compiler Plug-ins
  - Code generatie
- Computernetwerken
- Operating Systems
- Database systemen
- Parallel & Distributed Computing

# Voorbeeld 1

- Port “SMACK” naar x86 architectuur.
- Lezen & begrijpen technische documentatie.
- Implementatie & debugging.
- In hoeverre kan OS code platform-onafhankelijk/generiek zijn?

# Voorbeeld 2

- Inrichting interconnect “Leiden Zipper”
- LZ verbindt DAS4 en LGM clusters.
- Hoe kunnen deze clusters samen worden gebruikt?
- Bestuderen interconnectie topologie, InfiniBand interconnect.
- Implementatie MPI wrapper.
- Benchmarking.



# In het algemeen

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

Bedankt voor jullie aandacht!

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