TECHNISCHE UNIVERSITÄT DRESDEN

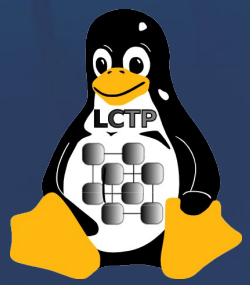
Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH)

Linux Cluster in Theorie und Praxis

Vergleich von C/OpenMP, Go und Chapel am Beispiel 'Quadratisches Sieb'

Ronny Brendel

10. Februar 2012



ronny.brendel@tu-dresden.de

ZIH
Zentrum für Informationsdienste und Hochleistungsrechnen

Inhalt

- Einführung
  - Das Quadratische Sieb
  - C/OpenMP
  - Go
  - Chapel
- Erläuterungen zur Parallelisierung
- Leistungsvergleich
  - Aufbau
  - Performance
  - Weitere Eigenschaften
- Fazit

Einführung - Quadratisches Sieb

- Gegeben: **n**, gesucht: **a** und **b**, sodass gilt: **n = a * b**

  → $n = (x+y)*(x-y)$

  → $0 \equiv (x+y)*(x-y) \pmod{n}$

  → $0 \equiv x^2 - y^2 \pmod{n}$

  → $x^2 \equiv y^2 \pmod{n}$

- x(i) und Faktorbasis wählen
- Siebschritt: Für jedes i: $y(i)^2 \equiv x(i)^2 - n \pmod{n}$ faktorisieren → $x(i), y(i)^2$
- Alle Kombination verschiedener y(i) probieren → y^2

  → $(x(1)*x(2))^2 \equiv y(1)*y(2) \leftrightarrow x^2 \equiv y^2 \pmod{n}$

  → x, y → (x+y)*(x-y) → … → **a * b = n**

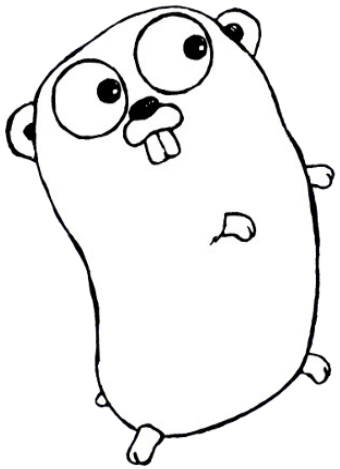
Einführung - C/OpenMP

- Erweiterung für C/C++ und Fortran
- Funktionen + Präprozessordirektiven (Compiler-Unterstützung nötig)
- Ausschließlich Shared Memory
- Loop Parallelism
- Data Parallelism

http://openmp.org

Einführung - Go

- General purpose - / System - Programmiersprache
- Kompiliert, statisch getypt, garbage collected, besitzt umfangreiche Standardbibliothek
- Eingebaute Nebenläufigkeit
  - *Goroutines:* Kooperatives, leicht-gewichtiges multithreading (vgl. OpenMP Tasks)
  - *Channels:* Getypte Pipes



http://golang.org

Einführung - Chapel

- Programmiersprache speziell für HPC
- Übersetzen nach C, danach Kompilieren mit System-Compiler
- Echte Arrays, flexible Indizierung (vgl. Fortran)
- Eingebaute Nebenläufigkeit
  - Data Parallelism
  - Task Parallelism
  - Nested Parallelism

http://chapel.cray.com

Erläuterungen zur Parallelisierung

- Beschränkung auf Shared Memory-Parallelisierung

- Siebschritt:

```
foreach x(i) in [min..max] {
   worked, factors := factorize(x(i)^2 - n,
   base)
   if worked == true {

      results[i] = factors /* „y(i)^2 */
   }
}
```

Vergleich - Aufbau

- Testfokus
  - Absolute Laufzeit
  - Skalierbarkeit

- Testsystem: Atlas
  - Neue Datenauswerte-Komponente des hrsk.tu-dresden.de
  - 64 Cores pro Knoten, 2.2 - 3.1 GHz (Opteron 6274)

Vergleich – Performance

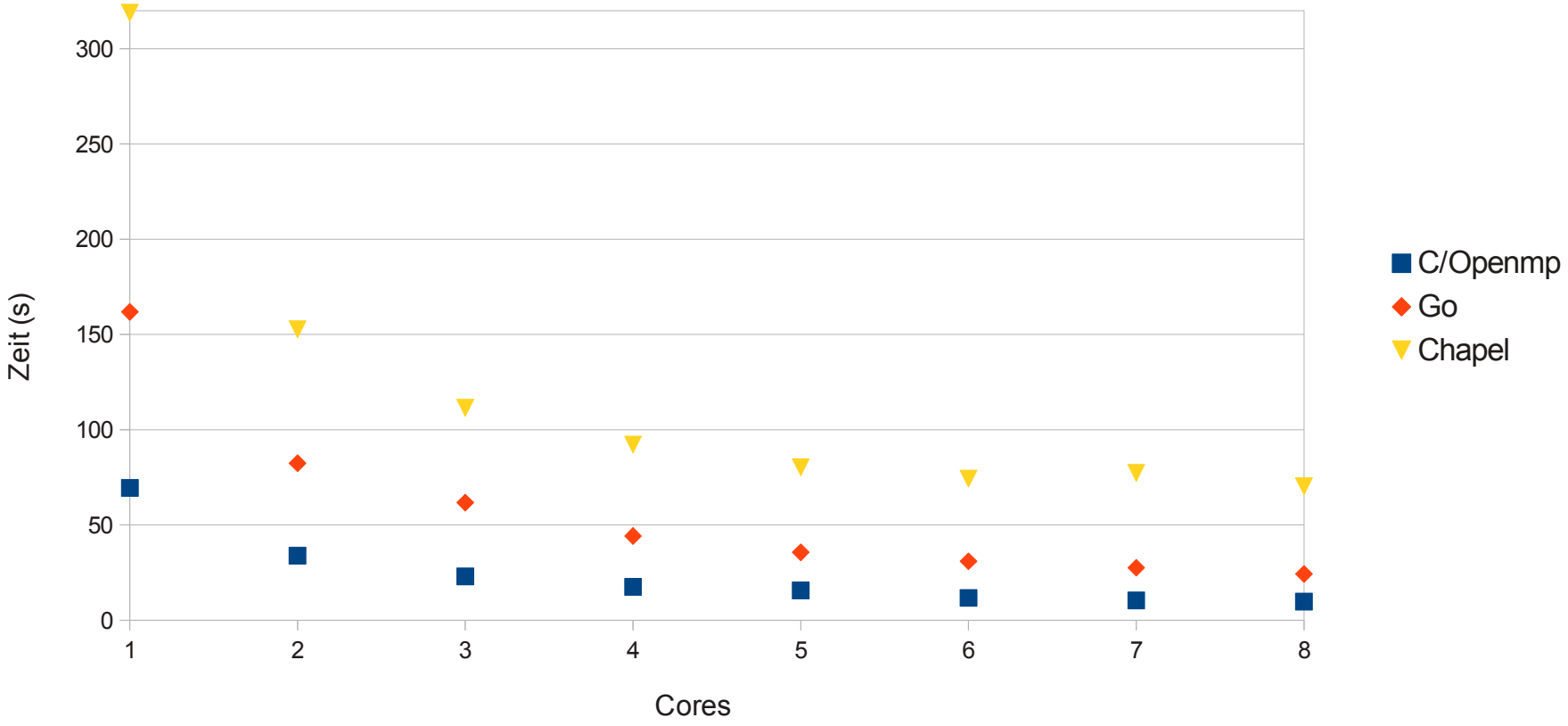


Abbildung 1: Absolute Laufzeit - Siebschritt

Vergleich – Performance

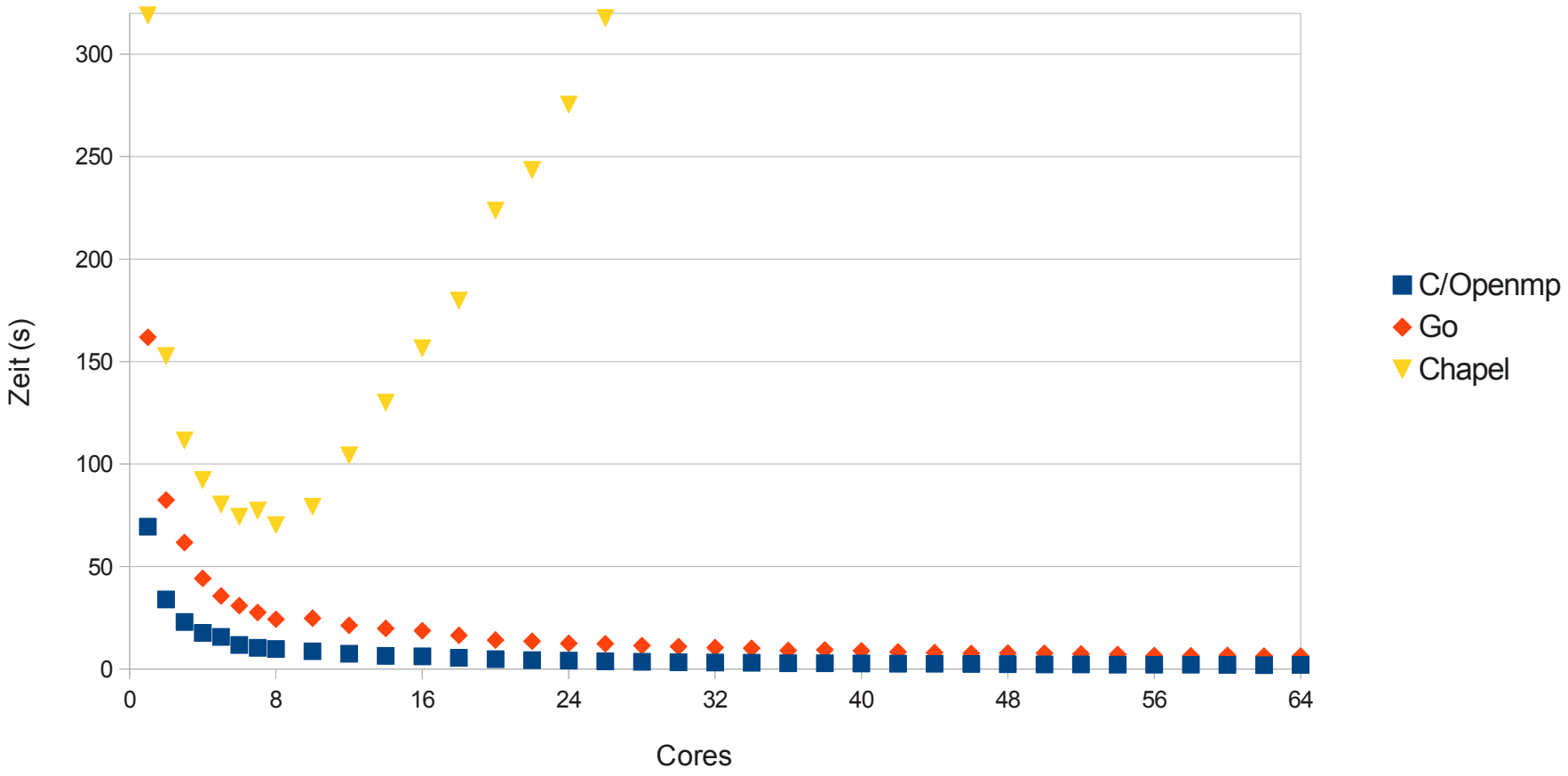


Abbildung 2: Absolute Laufzeit - Siebschritt

Vergleich – Performance

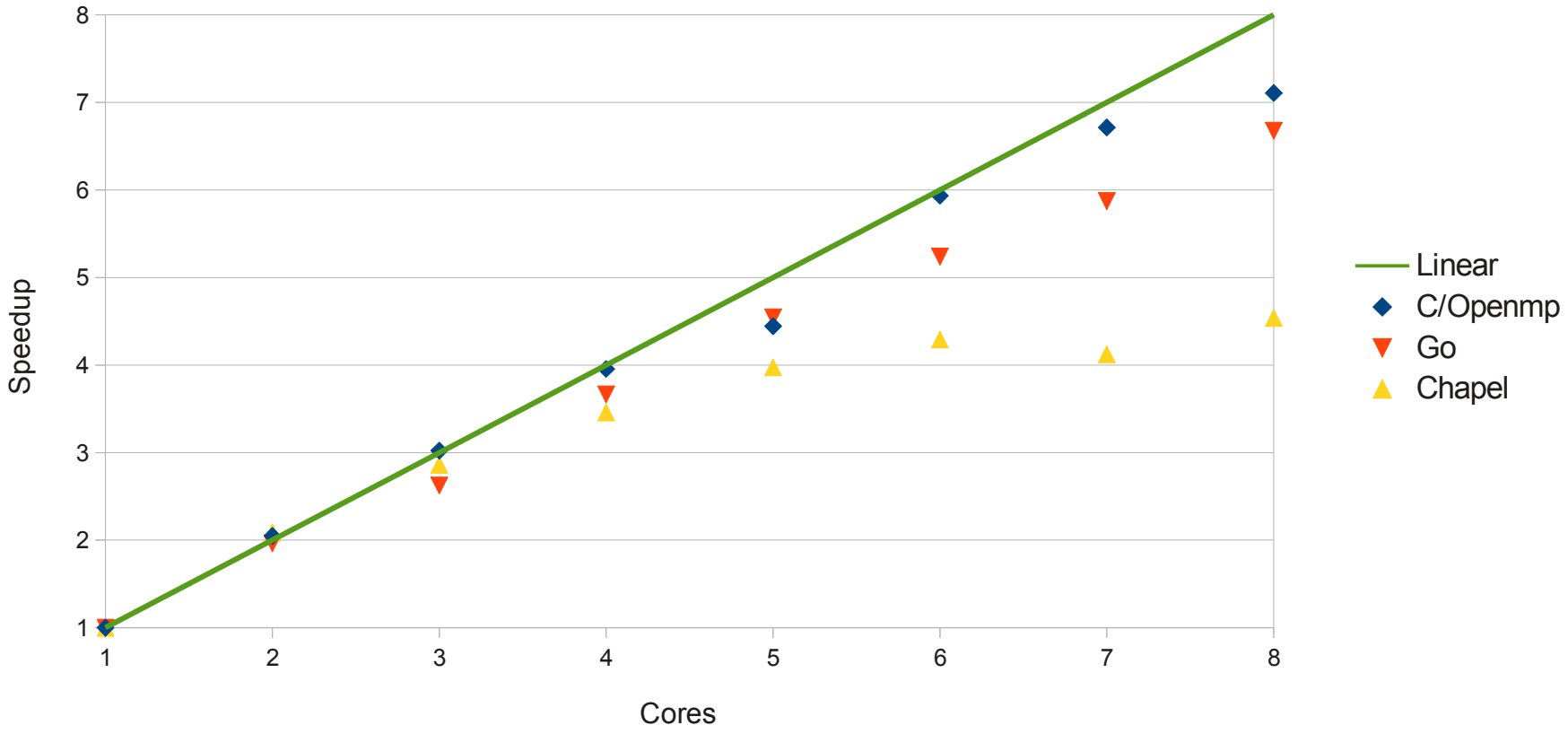


Abbildung 3: Speedup - Siebschritt

Vergleich – Performance

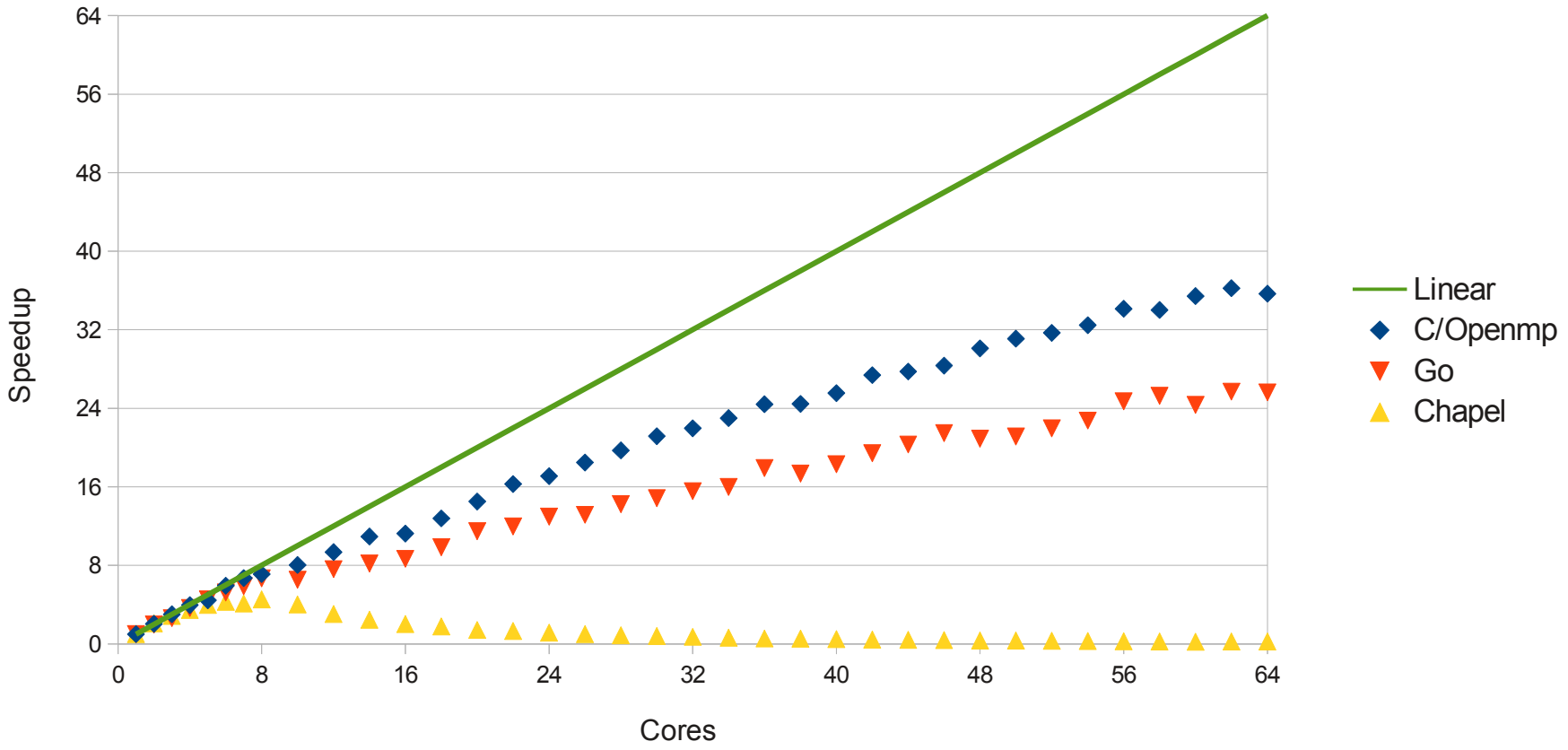


Abbildung 4: Speedup - Siebschritt

Vergleich – weitere Eigenschaften

Eigenschaft		C/OpenMP	Go	Chapel
Quelltext	Zeilen	567	416	471
	Worte	2240	1700	1700
	KiB	20	12	13,5
Parallelisierung	Shared Memory	ja	ja	ja
	Clustering	nein	manuell	ja
Speichervewaltung		manuell	automatisch	gemischt
Kompilierzeit (Sekunden)		0.4	0.4	6.8

Fazit

- C/OpenMP
  - Bestes Ergebnis
  - Größter Quelltext-Umfang (+25% gegenüber Go)
- Go
  - 200% - 500% der Laufzeit von C/OpenMP
  - Kleinster Quelltextumfang
- Chapel
  - 300% - 500% der Laufzeit von C/OpenMP
  - Skalierbarkeit und Geschwindigkeit nicht einschätzbar

Quellen

- [1] OpenMP API 3.1, Webseite, 09-02-12

  http://www.openmp.org/mp-documents/OpenMP3.1.pdf

- [2] Go Dokumentation, Webseite 09-02-12

  http://golang.org/doc/docs.html

- [3] Chapel Tutorials, Webseite 09-02-12

  http://chapel.cray.com/tutorials.html

- [4] Vorlesung Kryptologie (TU Dresden WS10/11), Ulrike Baumann

- [5] Faktorisieren mit dem Quadratischen Sieb, R.-H. Schulz und H. Witten, Webseite 09-02-12

  ftp://ftp.math.fu-berlin.de/pub/math/publ/pre/2011/Pr-A-11-03.pdf