
Benchmarking von Bilddatenstrukturen

Ralph Guderlei, Timo Tonn

Universität Ulm

- **Benchmark:** "Something which can be used as a standard by which other things are judged or measured"

- **Benchmark:** "Something which can be used as a standard by which other things are judged or measured"
- Objekte müssen vergleichbar sein

- **Benchmark:** "Something which can be used as a standard by which other things are judged or measured"
- Objekte müssen vergleichbar sein
- Objekte müssen auf die selbe Art und Weise verglichen werden

- **Benchmark:** "Something which can be used as a standard by which other things are judged or measured"
- Objekte müssen vergleichbar sein
- Objekte müssen auf die selbe Art und Weise verglichen werden
- Vergleichskriterium bzw.
Metrik: $\frac{Zeit}{Operation}$

- Simuliert wird das Laufzeitverhalten bestimmter Operationen in Abhängigkeit von Datenstruktur und Größe.

- Simuliert wird das Laufzeitverhalten bestimmter Operationen in Abhängigkeit von Datenstruktur und Größe.
- Untersuchte Operationen:
Subrange, Slice (für jede Dimension), Transpose, Flip (um jede Achse)

- Simuliert wird das Laufzeitverhalten bestimmter Operationen in Abhängigkeit von Datenstruktur und Größe.
- Untersuchte Operationen:
Subrange, Slice (für jede Dimension), Transpose, Flip (um jede Achse)
- Wir haben uns auf maximal zweidimensionale Datenstrukturen beschränkt

- Festlegung eines Interface, welches die Funktionalität vorgibt

Vorgehensweise

- Festlegung eines Interface, welches die Funktionalität vorgibt
- Die Datentypen implementieren dieses Interface

Vorgehensweise

- Festlegung eines Interface, welches die Funktionalität vorgibt
- Die Datentypen implementieren dieses Interface
- Der Benchmark basiert auf der Funktionalität des Interface und ist für alle Datentypen gleich.

- Festlegung eines Interface, welches die Funktionalität vorgibt
- Die Datentypen implementieren dieses Interface
- Der Benchmark basiert auf der Funktionalität des Interface und ist für alle Datentypen gleich.
- Der Benchmark durchläuft einen Zyklus von Operationen

- Festlegung eines Interface, welches die Funktionalität vorgibt
- Die Datentypen implementieren dieses Interface
- Der Benchmark basiert auf der Funktionalität des Interface und ist für alle Datentypen gleich.
- Der Benchmark durchläuft einen Zyklus von Operationen
- Die Durchlaufzeit wird dann zu Vergleichszwecken gemessen

Ergebnisse

Die Datenstrukturen unterscheiden sich sehr stark in ihrem Laufzeitverhalten (150 mb Last, also etwa 4500x4500 Pixel):

Typ	Subrange	Slice (1. Komp.)	Slice(2.)
Strides + Offset	0	0	0
Index Array	0.047	0.046	0.046
Multidim. Array	427	0	5

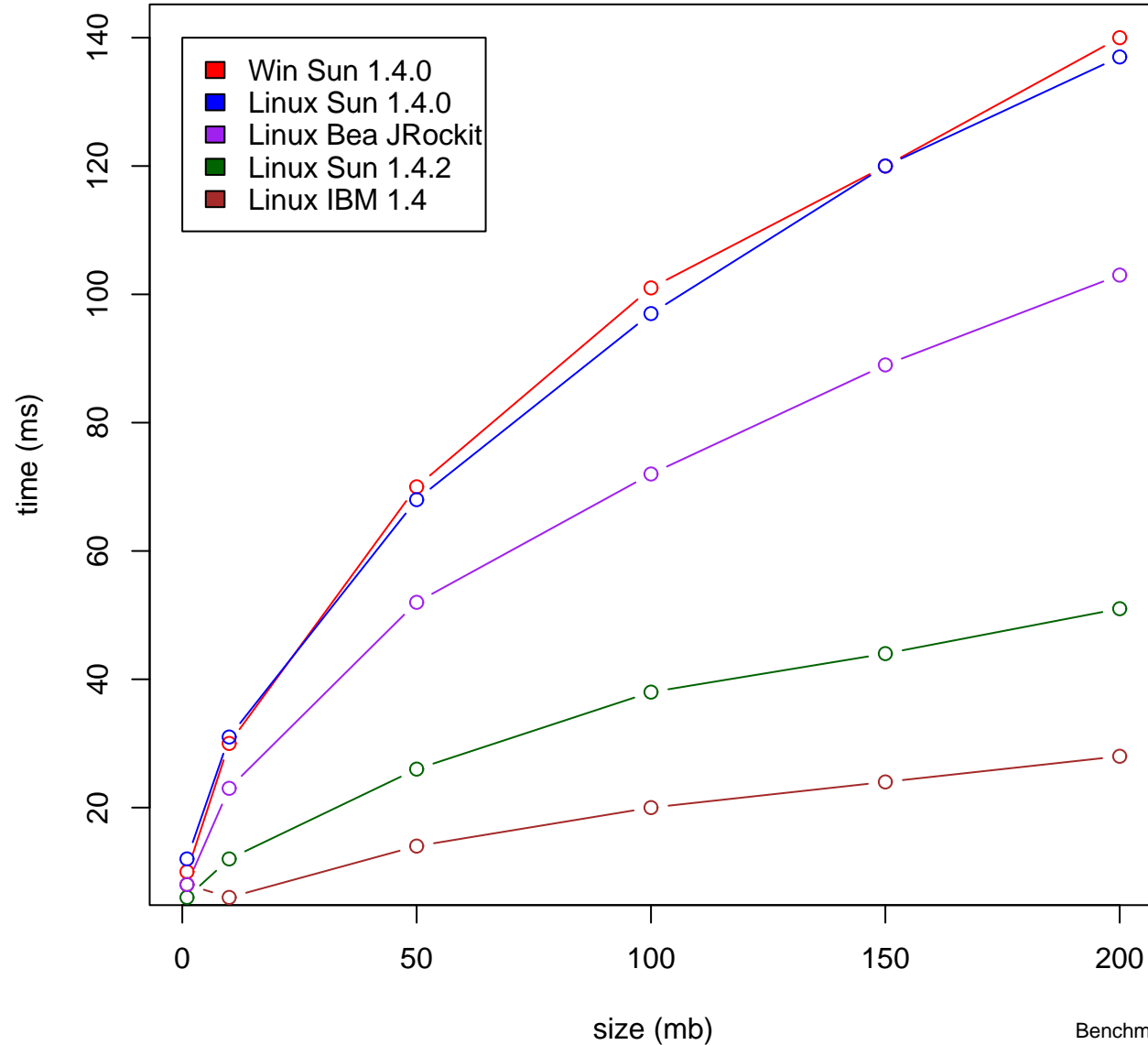
Typ	Transpose	Flip (1. Achse)	Flip (2.)
Strides + Offset	0	0	0
Index Array	0	0.044	0.044
Multidim. Array	10602	472	1

- Das beste Laufzeitverhalten zeigt die "Strides and Offset"-Struktur
- Das schlechteste Laufzeitverhalten das multidimensionale Array
- Insgesamt bestätigten die Versuche die theoretischen Überlegungen.
 - Die "Strides and Offset"-Struktur verhält sich linear in der Anzahl der Dimension
 - Der Index Array linear in der Anzahl der Elemente
 - Der multidimensionale Array schlechtestenfalls quadratisch in der Anzahl der Elemente

- Die eingesetzten Betriebssysteme wirken sich nicht auf das Laufzeitverhalten aus.
- Die getesteten Java-VMs zeigen ein sehr unterschiedliches Laufzeitverhalten, und machen den Hauptunterschied aus.
- Performanteste JVM war IBMs JDK 1.4

JVM Performance

JVM Performance



verwendete Tools

- *Eclipse* als Entwicklungsumgebung
<http://www.eclipse.org>
- *JUnit* als Unit Test-Framework
<http://junit.org>
- *Ant* als Build-Tool
<http://ant.apache.org>

- R. Binder: *Testing Object-Oriented Systems*
- M. Fowler: *Refactoring*
- E. Gamma, R. Helms, R. Johnson, J. Vlissides:
Design Patterns

Addison-Wesley