

The Fast and the Furious

Große ASCII-Dateien schnell verarbeitet

Thomas Stecher

thomas.stecher@bgr.de

www.bgr.bund.de

Die User sagen

„Das Programm ist zu langsam“



http://blocati.de/wp-content/uploads/bilder/natur/2000609_stroh_trecker.jpg

Der Chef sagt

„Besorg Dir einen schnelleren Rechner.“



Quelle <http://www.24gb.de/Gallery/Autos/Toyota-Supra-the-fast-the-furious.jpg>

Fragen?



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Bergwerk analog



Bergwerk digital

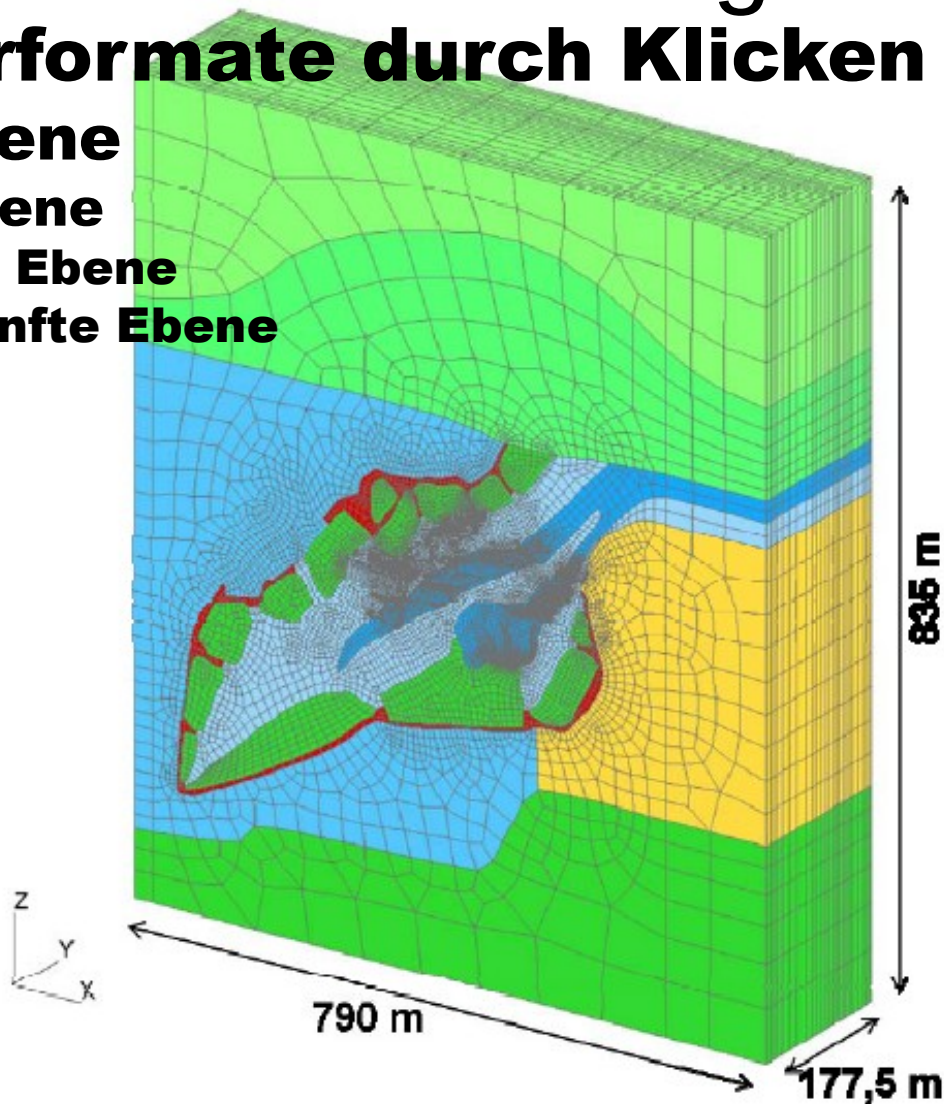
performate durch Klicken bearbeiten
Hexaeder (Würfel)

bene
bene
e Ebene
ünfte Ebene

Materialgruppen

Hohlräume

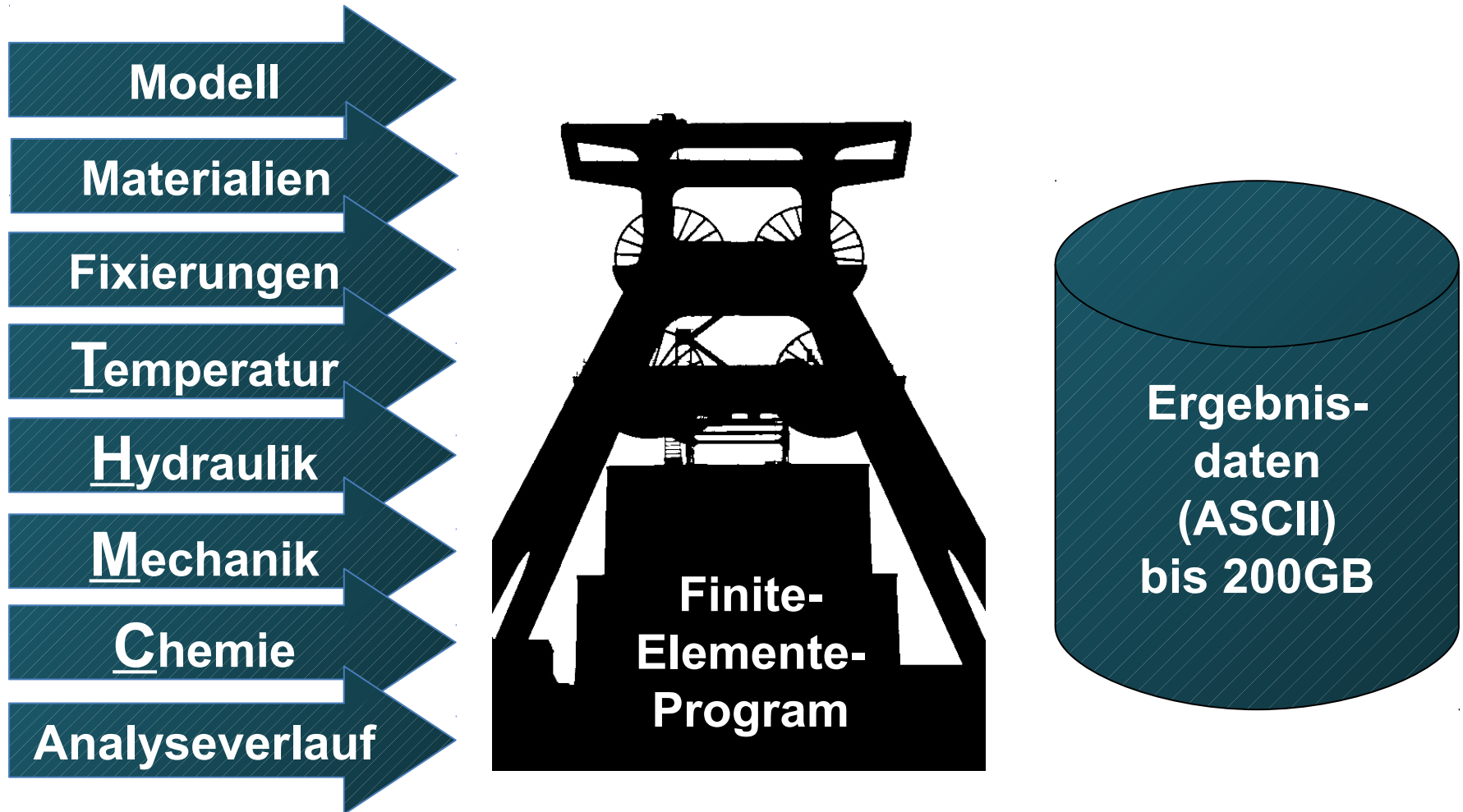
Fixierungen



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Der Datenproduzent

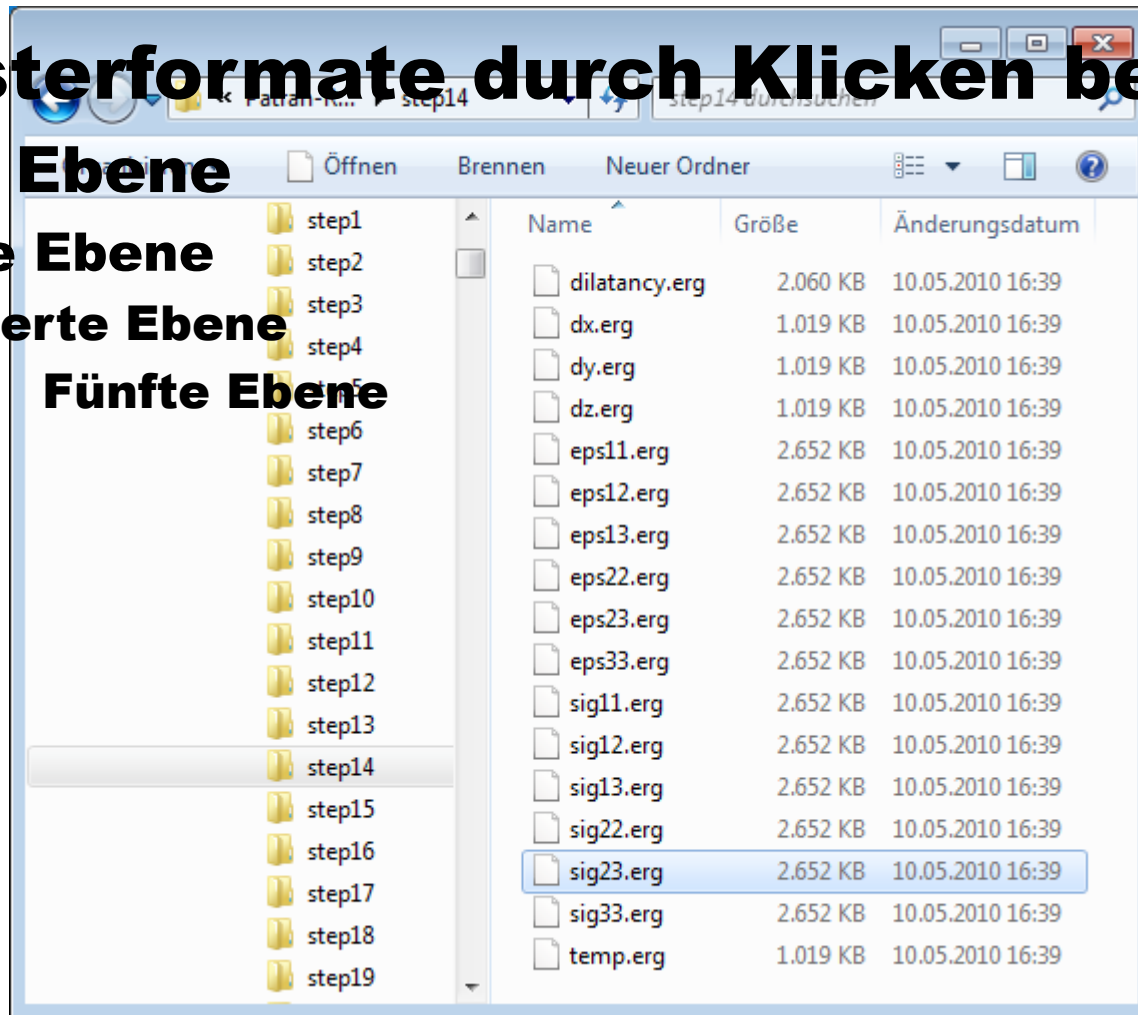


http://blog.mineralium.com/wp-content/uploads/2007/06/bergwerk_foerderturm.jpg

Anwendungsfall: Daten

sterformate durch **Klicken** bearbeiten

Ebene
e Ebene
ierte Ebene
Fünfte Ebene

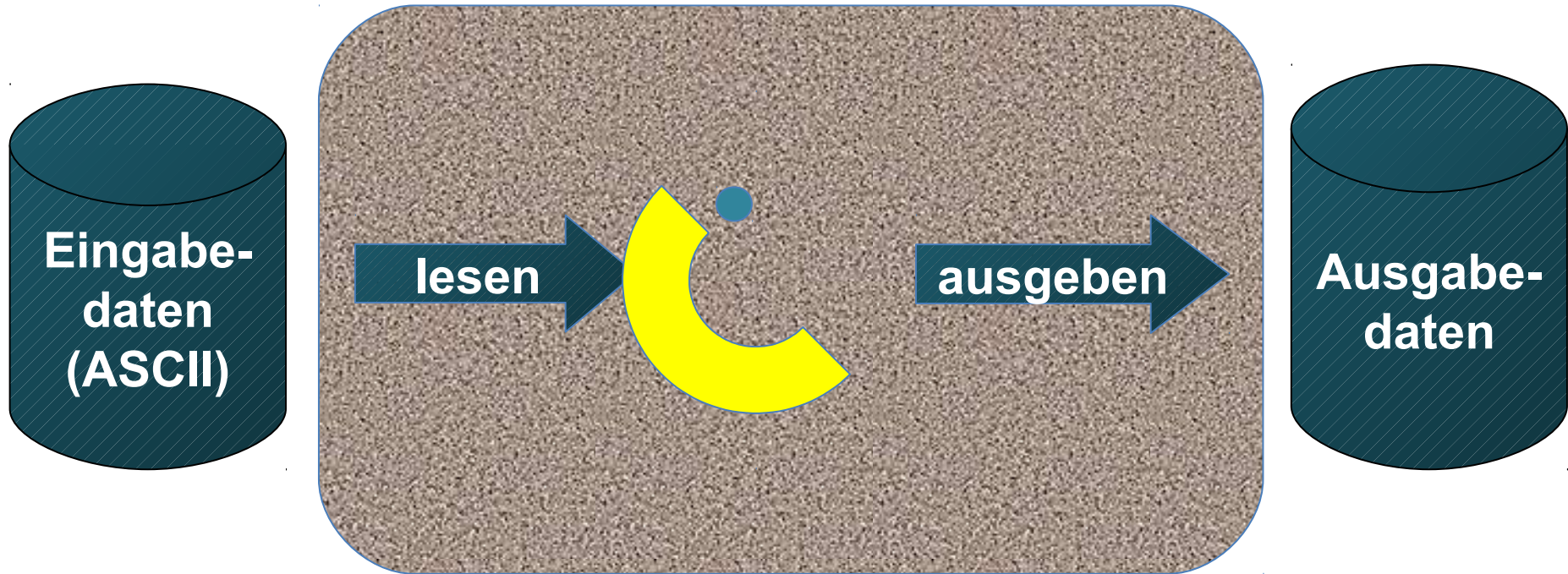


Verzeichnis se	1.160
Dateien	19.704
Größe gesamt	~55 GB

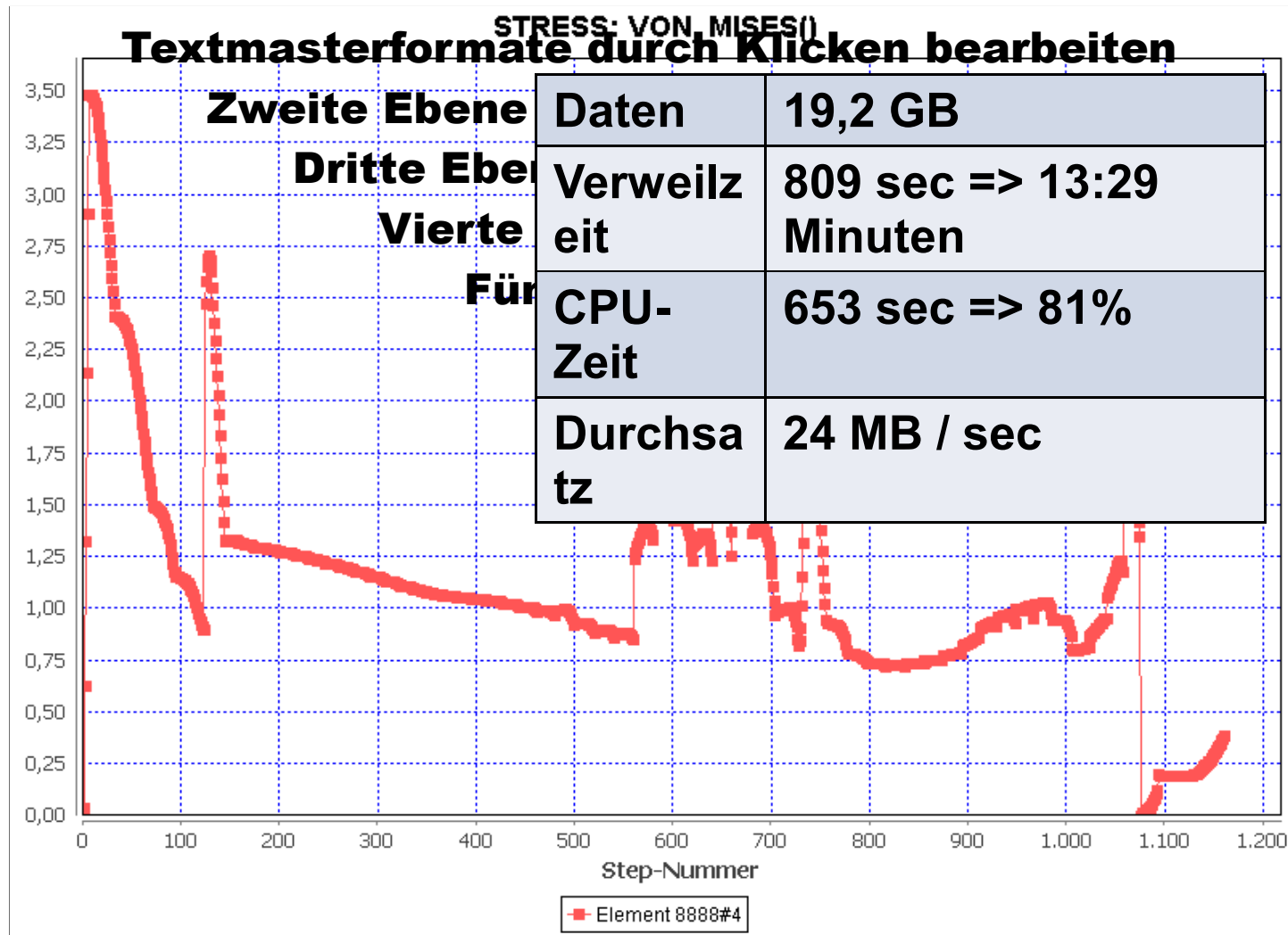
Anwendungsfall: Daten

sig22 19 2379.15900129 89 %I8%8E18%									
1	12	2	2	Textmasterformate durch Klicken bearbeiten					
8560	-4.15419893E+000	-4.27976943E+000	-3.96391879E+000	-4.10480202E+000	-4.15419893E+000	-4.27976943E+000	-3.96391879E+000	-4.10480202E+000	-4.15419893E+000
8561	-5.27328830E+000	-5.22676698E+000	-5.02436222E+000	-4.99285636E+000	-5.27328830E+000	-5.22676698E+000	-5.02436222E+000	-4.99285636E+000	-5.27328830E+000
8571	-5.76963940E+000	-5.59274587E+000	-5.93696511E+000	-5.75358765E+000	-5.76963940E+000	-5.59274587E+000	-5.93696511E+000	-5.75358765E+000	-5.76963940E+000
8572	-3.15327929E+000	-3.26869079E+000	-3.11454493E+000	-3.23010480E+000	-3.15327929E+000	-3.26869079E+000	-3.11454493E+000	-3.23010480E+000	-3.15327929E+000
8573	-2.63985074E+000	-3.00597053E+000	-1.67798916E+000	-2.6648110E+000	-2.63985074E+000	-3.00597053E+000	-1.67798916E+000	-2.6648110E+000	-2.63985074E+000
8574	-4.48718476E+000	-4.30866772E+000	-4.41943805E+000	-4.22852067E+000	-4.48718476E+000	-4.30866772E+000	-4.41943805E+000	-4.22852067E+000	-4.48718476E+000
8582	-8.67827406E+000	-8.61928905E+000	-8.99313712E+000	-8.90226094E+000	-8.67827406E+000	-8.61928905E+000	-8.99313712E+000	-8.90226094E+000	-8.67827406E+000
8583	-6.99685602E+000	-6.83056982E+000	-7.48242647E+000	-7.23891873E+000	-6.99685602E+000	-6.83056982E+000	-7.48242647E+000	-7.23891873E+000	-6.99685602E+000
8584	-1.90180992E+000	-2.12715059E+000	-2.17748744E+000	-2.42266899E+000	-1.90180992E+000	-2.12715059E+000	-2.17748744E+000	-2.42266899E+000	-1.90180992E+000
8585	-3.23080802E+000	-3.12353177E+000	-3.13477190E+000	-3.04022151E+000	-3.23080802E+000	-3.12353177E+000	-3.13477190E+000	-3.04022151E+000	-3.23080802E+000
8586	-3.01579123E+000	-3.57134481E+000	-1.01235508E+000	-2.43972996E+000	-3.01579123E+000	-3.57134481E+000	-1.01235508E+000	-2.43972996E+000	-3.01579123E+000
8587	-5.64213892E+000	-5.26849056E+000	-6.97447954E+000	-6.16028134E+000	-5.64213892E+000	-5.26849056E+000	-6.97447954E+000	-6.16028134E+000	-5.64213892E+000
2	12	2	2	2					
8549	-1.09184287E+001	-1.08289658E+001	-1.04273337E+001	-1.04375660E+001	-1.09184287E+001	-1.08289658E+001	-1.04273337E+001	-1.04375660E+001	-1.09184287E+001
8550	-9.02928099E+000	-9.01305266E+000	-8.59344556E+000	-8.65740754E+000	-9.02928099E+000	-9.01305266E+000	-8.59344556E+000	-8.65740754E+000	-9.02928099E+000
8551	-7.51997242E+000	-7.50404127E+000	-7.18324522E+000	-7.21509615E+000	-7.51997242E+000	-7.50404127E+000	-7.18324522E+000	-7.21509615E+000	-7.51997242E+000
8552	-9.86826808E+000	-9.82266514E+000	-9.79897418E+000	-9.75628945E+000	-9.86826808E+000	-9.82266514E+000	-9.79897418E+000	-9.75628945E+000	-9.86826808E+000
8558	-8.58116962E+000	-8.46464182E+000	-8.59811962E+000	-8.48858763E+000	-8.58116962E+000	-8.46464182E+000	-8.59811962E+000	-8.48858763E+000	-8.58116962E+000
8559	-6.12351050E+000	-6.19330640E+000	-6.18996783E+000	-6.25710052E+000	-6.12351050E+000	-6.19330640E+000	-6.18996783E+000	-6.25710052E+000	-6.12351050E+000
8563	-1.05101411E+001	-1.04083305E+001	-1.05995618E+001	-1.04977139E+001	-1.05101411E+001	-1.04083305E+001	-1.05995618E+001	-1.04977139E+001	-1.05101411E+001
8564	-8.64969318E+000	-8.68084888E+000	-8.77293883E+000	-8.80111274E+000	-8.64969318E+000	-8.68084888E+000	-8.77293883E+000	-8.80111274E+000	-8.64969318E+000
8570	-7.89058966E+000	-7.76242480E+000	-7.98789979E+000	-7.85547586E+000	-7.89058966E+000	-7.76242480E+000	-7.98789979E+000	-7.85547586E+000	-7.89058966E+000
8576	-1.15076759E+001	-1.15362497E+001	-1.19014597E+001	-1.18747049E+001	-1.15076759E+001	-1.15362497E+001	-1.19014597E+001	-1.18747049E+001	-1.15076759E+001
8577	-1.07695764E+001	-1.07384431E+001	-1.10918746E+001	-1.10243529E+001	-1.07695764E+001	-1.07384431E+001	-1.10918746E+001	-1.10243529E+001	-1.07695764E+001
8578	-9.21339060E+000	-9.18680743E+000	-9.51363386E+000	-9.45694530E+000	-9.21339060E+000	-9.18680743E+000	-9.51363386E+000	-9.45694530E+000	-9.21339060E+000
3	12	2	2	2					
8553	-1.14729590E+001	-1.13242143E+001	-1.13173749E+001	-1.11795917E+001	-1.14729590E+001	-1.13242143E+001	-1.13173749E+001	-1.11795917E+001	-1.14729590E+001
8554	-1.16036268E+001	-1.15651936E+001	-1.11101227E+001	-1.11642787E+001	-1.16036268E+001	-1.15651936E+001	-1.11101227E+001	-1.11642787E+001	-1.16036268E+001
8555	-1.13242143E+001	-1.13173749E+001	-1.11795917E+001	-1.11642787E+001	-1.13242143E+001	-1.13173749E+001	-1.11795917E+001	-1.11642787E+001	-1.13242143E+001

Die Weiterverarbeitung



Anwendungsfall: TimeHistory



Beeinflussende Faktoren

Hardware

Prozessor

Architektur

Speichermedium

Betriebssystem

Caches

Compiler / VM

Daten

Anzahl Dateien

Dateigröße

Struktur

Typ

Eigenes Programm

Puffergröße

Lesetechnik



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Die eigentliche Aufgabe

Lesen einer ASCII-Datei

Zerlegen in Zeilen

Zerlegen in Worte

Konvertierung

int / long

float / double



Und das *m ö g l i c h s t*

2	5	1.31875000E+00	0	1
2.41619827E+01	2.41982617E+01	3.71619827E+01	2.41619827E+01	
2.41982617E+02	3.71619827E+01	2.41619827E+01	2.41982612E+01	
3.71619827E+01	2.41619827E+01	2.41982617E+01	3.71619827E+01	
2.41619827E+00	2.41982616E+01	3.71619827E+01	2.41619827E+01	
2.41982617E+01	3.71619827E+01	2.41619827E+01	2.41987616E+01	
3.71619827E+01	2.41619827E-01	2.41982617E+01	3.71619827E+01	
2.41619827E+01	2.41982997E+01	3.71619827E+01	0.71619827E+01	
1.71619827E-01	2.41619827E+01	2.41982617E+01	3.71659827E+01	

Vergleichende Tests

Voraussetzungen

ASCII-Datei: 500MB

Puffergröße: 8kB

10 x Lesen

Vergleichsgrößen

Beste Verweilzeit

Beste CPU-Zeit



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Werkzeuge

Server-VM

System.out.println()

System.nanoTime()

**ManagementFactory.getThreadMXBean()
getCurrentThreadCpuTime()**

getCurrentThreadUserTime()

JVisualVM

**Energieoptionen des Rechners
auf „Vollgas“ stellen!**



Quelle: <http://lovehateadvertising.files.wordpress.com/2010/07/keep-it-simple-stupid-kiss.png>



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

FileReader

```
Reader reader = null;
try {
    reader = new FileReader(ASCII_FILE) ;
    char[] buffer = new char[BUFF_SIZE] ;
    int readCnt = 0;
    while (reader.read(buffer) != -1) {
        readCnt++;
    }
    System.out.println("#Read-OPs=" + readCnt);
} finally {
    if (reader != null) {
        reader.close();
    }
}
```

Übersicht

Test	#read	CPU [millis]	Dauer [millis]
FileReader	64000	2090	2099

LineNumberReader(FileReader)

```
LineNumberReader reader = null;
try {
    FileReader fileReader = new FileReader(ASCII_FILE);
    reader = new LineNumberReader(fileReader, BUFF_SIZE);
    int lineCnt = 0;
    while (reader.readLine() != null) {
        lineCnt++;
    }
    System.out.println("#Lines=" + lineCnt);
} finally {
    if (reader != null) {
        reader.close();
    }
}
```

Übersicht

Test	#read	CPU [millis]	Dauer [millis]
FileReader	64000	2090	2099
LineNumberReader(FileReader)	8192000	2948	2985

LineNumberReader(FileReader) + split()

```
int wordCnt = 0;
String line = reader.readLine();
while (line != null) {
    wordCnt += line.split(" ").length;
    line = reader.readLine();
}
System.out.println("#Words=" + wordCnt);
```

Übersicht

Test	#read	CPU [millis]	Dauer [millis]
FileReader	64000	2090	2099
LineNumberReader(FileReader)	8192000	2948	2985
LineNumberReader(FileReader) + split()	8192000	13353	13526

FileInputStream

```
FileInputStream stream = null;
try {
    stream = new FileInputStream(ASCII_FILE) ;
    byte[] buffer = new byte[BUFF_SIZE] ;
    int readCnt = 0;
    while (stream.read(buffer) != -1) {
        readCnt++;
    }
    System.out.println("#Read-OPs=" + readCnt);
} finally {
    if (stream != null) {
        stream.close();
    }
}
```

Übersicht

Test	#read	CPU [millis]	Dauer [millis]
FileReader	64000	2090	2099
LineNumberReader(FileReader)	8192000	2948	2985
LineNumberReader(FileReader) + split()	8192000	13353	13526
FileInputStream	64000	390	401

LineNumberReader(FileInputStream)

```
LineNumberReader reader = null;
try {
    InputStream stream = new FileInputStream(ASCII_FILE);
    Reader streamReader = new InputStreamReader(stream);
    reader = new LineNumberReader(streamReader, BUFF_SIZE);
    int lineCnt = 0;
    while (reader.readLine() != null) {
        lineCnt++;
    }
    System.out.println("#Read-OPs=" + lineCnt);
} finally {
    if (reader != null) {
        reader.close();
    }
}
```

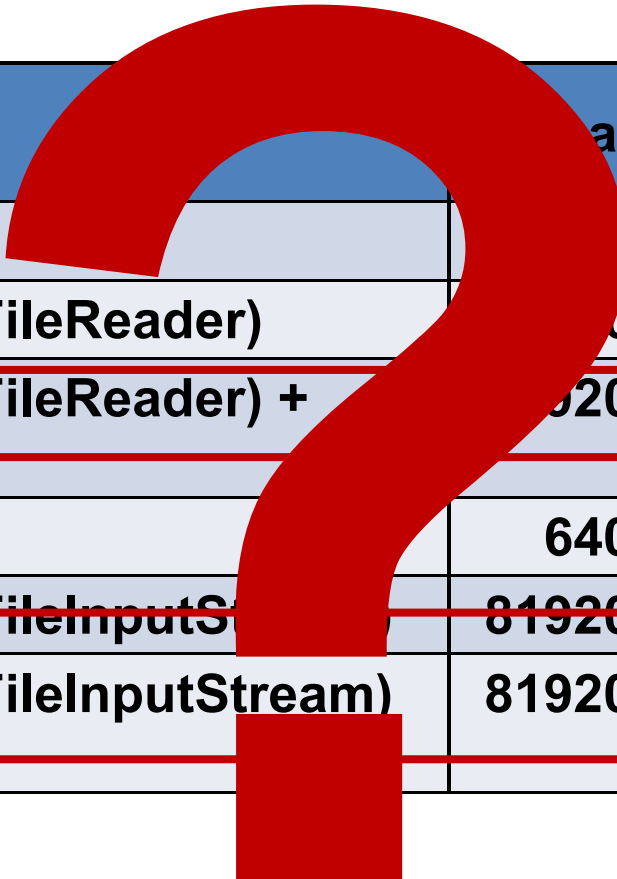
Übersicht

Test	#read	CPU [millis]	Dauer [millis]
FileReader	64000	2090	2099
LineNumberReader(FileReader)	8192000	2948	2985
LineNumberReader(FileReader) + split()	8192000	13353	13526
FileInputStream	64000	390	401
LineNumberReader(FileInputStream)	8192000	2870	2885

LineNumberReader(FileInputStream) + Scanner

```
int wordCnt = 0;
String line = reader.readLine();
while (line != null) {
    Scanner scanner = new Scanner(line);
    while (scanner.hasNext()) {
        scanner.next();
        wordCnt++;
    }
    line = reader.readLine();
}
System.out.println("#Words=" + wordCnt);
```

Übersicht



Test	Zeilenanzahl	CPU [millis]	Dauer [millis]
FileReader	2000	2090	2099
LineNumberReader(FileReader)	2000	2948	2985
LineNumberReader(FileReader) + split()	8192000	13353	13526
FileInputStream	64000	390	401
LineNumberReader(FileInputStream)	8192000	2870	2885
LineNumberReader(FileInputStream) + Scanner	8192000	96283	98430

Scanner „falsch“ bedient?

```
Scanner scanner = null;
try {
    scanner = new Scanner(ASCII_FILE) ;
    int wordCnt = 0;
    while (scanner.hasNext()) {
        wordCnt++;
        scanner.next();
    }
    System.out.println("#Words=" + wordCnt);
} finally {
    if (scanner != null) {
        scanner.close();
    }
}
```

Übersicht



Test	PU [lis]	Dauer [millis]
LineNumberRe split()	353	13526
LineNumberRe + Scanner	283	98430
Scanner	4241	55195

Selbst ist der Entwickler

Basiskonzepte werden durch Klicken b

ungeeignete Ebene

überladen

zu langsam

Dritte Ebene

Vierte Ebene

Fünfte Ebene



Quelle http://1.bp.blogspot.com/_IA5nokOFh84/TQfvxqGRWil/AAAAAAAAAEI0/ouxh5Zvzjmo/s1600/bob-the-builder.jpg



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Was kann ich erreichen?

Test	#read	CPU [millis]	Dauer [millis]
FileReader	64000	2090	2099
LineNumberReader(FileReader)	8192000	2948	2985
LineNumberReader(FileReader) + split()	8192000	13353	13526
FileInputStream	64000	390	401
LineNumberReader(FileInputStream)	8192000	2870	2885
LineNumberReader(FileInputStream) + Scanner	8192000	96283	98430

Was habe ich erreicht?



Test	CPU [millis]	Dauer [millis]
FileInputStream	390	401
AsciiParser.getN	1528	1542
AsciiParser.getNextWo	3198	3208
LineNumberReader(FileReader).split()	8192000	13353
		13526

**4 x
schneller**

Probefahrt gefällig?

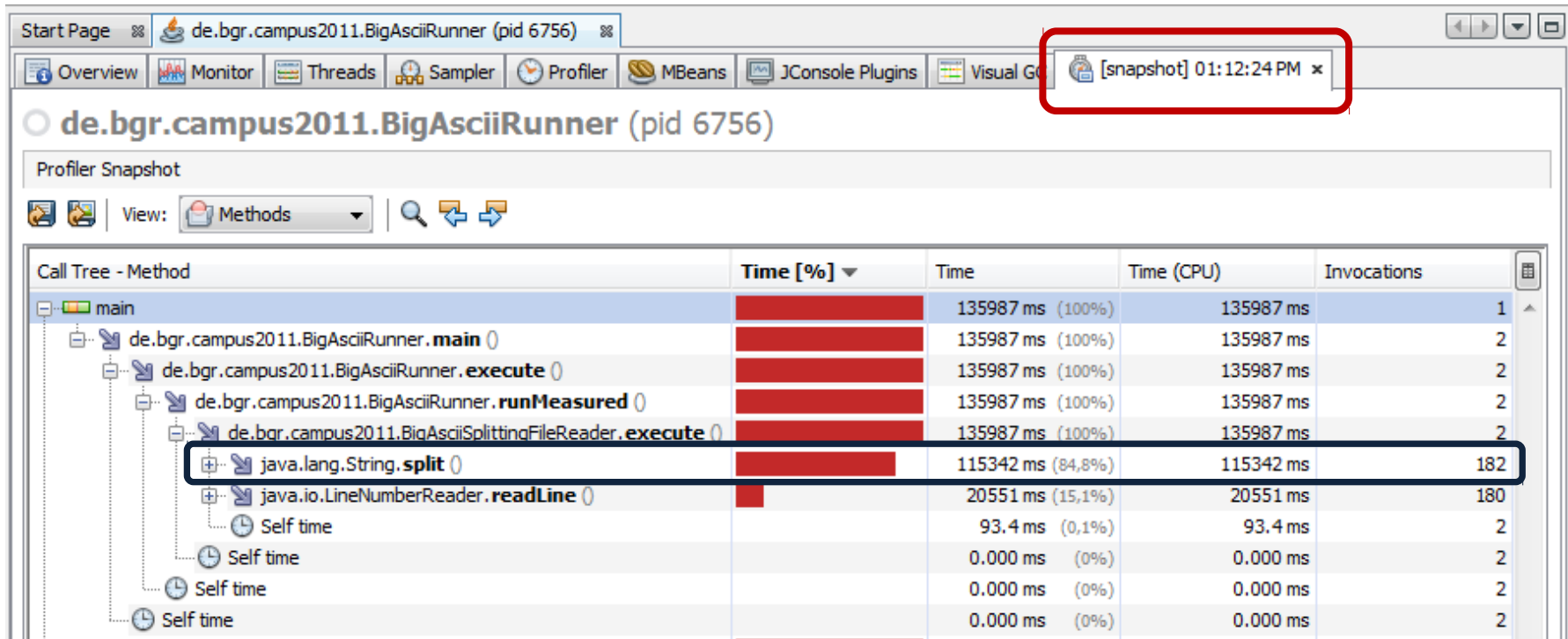


Probefahrt gefällig?

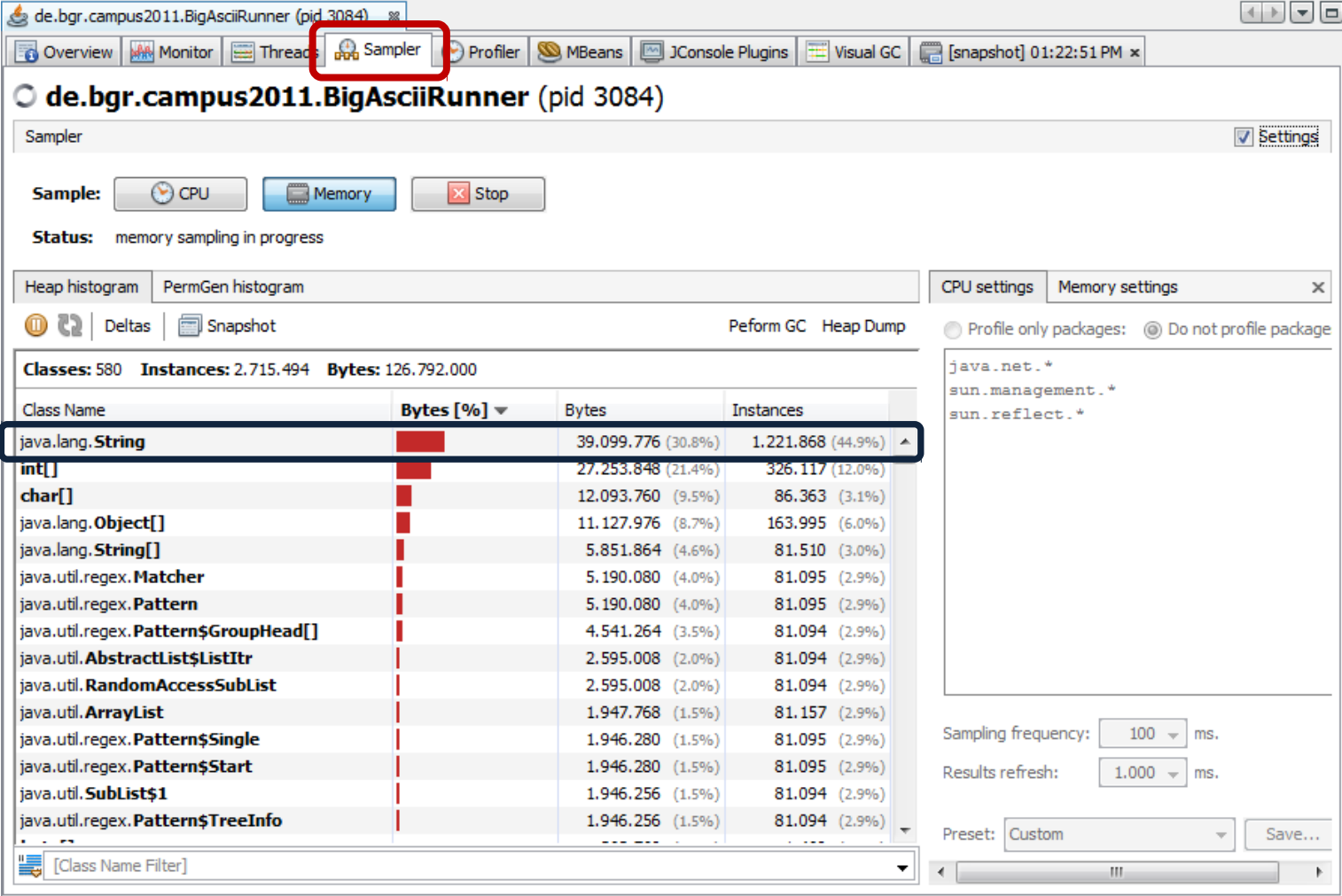
```
AsciiParser parser = new AsciiParser(ASCII_FILE) ;  
try {  
    parser.openFile() ;  
    int wordCnt = 0;  
    while (parser.getNextWord() != null) {  
        wordCnt++;  
    }  
    System.out.println("#Words=" + wordCnt);  
} finally {  
    parser.close();  
}
```

Wo war die Bremse?

JVisualVM: Sampler / Snapshot



Wo war die Bremse?



de.bgr.campus2011.BigAsciiRunner (pid 3084)

Overview Monitor Threads **Sampler** Profiler MBeans JConsole Plugins Visual GC [snapshot] 01:22:51 PM x

de.bgr.campus2011.BigAsciiRunner (pid 3084)

Sampler ☒ Settings

Sample:

Status: memory sampling in progress

Heap histogram PermGen histogram

Perform GC Heap Dump

Classes: 580 Instances: 2,715,494 Bytes: 126,792,000

Class Name	Bytes [%]	Bytes	Instances
java.lang.String	39,099,776 (30.8%)	1,221,868 (44.9%)	
int[]	27,253,848 (21.4%)	326,117 (12.0%)	
char[]	12,093,760 (9.5%)	86,363 (3.1%)	
java.lang.Object[]	11,127,976 (8.7%)	163,995 (6.0%)	
java.lang.String[]	5,851,864 (4.6%)	81,510 (3.0%)	
java.util.regex.Matcher	5,190,080 (4.0%)	81,095 (2.9%)	
java.util.regex.Pattern	5,190,080 (4.0%)	81,095 (2.9%)	
java.util.regex.Pattern\$GroupHead[]	4,541,264 (3.5%)	81,094 (2.9%)	
java.util.AbstractList\$Itr	2,595,008 (2.0%)	81,094 (2.9%)	
java.util.RandomAccessSubList	2,595,008 (2.0%)	81,094 (2.9%)	
java.util.ArrayList	1,947,768 (1.5%)	81,157 (2.9%)	
java.util.regex.Pattern\$Single	1,946,280 (1.5%)	81,095 (2.9%)	
java.util.regex.Pattern\$Start	1,946,280 (1.5%)	81,095 (2.9%)	
java.util.SubList\$1	1,946,256 (1.5%)	81,094 (2.9%)	
java.util.regex.Pattern\$TreeInfo	1,946,256 (1.5%)	81,094 (2.9%)	

[Class Name Filter]

CPU settings Memory settings x

Profile only packages: ☐ Do not profile package

```
java.net.*
sun.management.*
sun.reflect.*
```

Sampling frequency: 100 ms.

Results refresh: 1,000 ms.

Preset: Custom Save...

Die Vermutung

teuer

Strings in Worte zerlegen

Viele String Objekte

billig

Lesen an sich per FileInputStream



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Ein Blick unter die Haube



Der Anlasser

```
public BasicAsciiParser(File file) throws IOException {  
    channel = new FileInputStream(file).getChannel();  
  
    bb = ByteBuffer.allocate(BUFF_SIZE);  
    bb.position(bb.capacity());  
  
    data = bb.array();  
  
    currentLine = new AsciiLine(data);  
  
    isEOF = (file.length() == 0);  
}
```

Der Motor

```
private void readBuffer() throws IOException {  
    int bytesRead = channel.read(bb) ;  
  
    bb.limit(bb.position()) ;  
    if (bytesRead < 0) {  
        isEOF = true;  
    } else {  
        readByteCount += bytesRead;  
    }  
    bb.rewind() ;  
}
```

NIO

`FileInputStream.channel.read()`



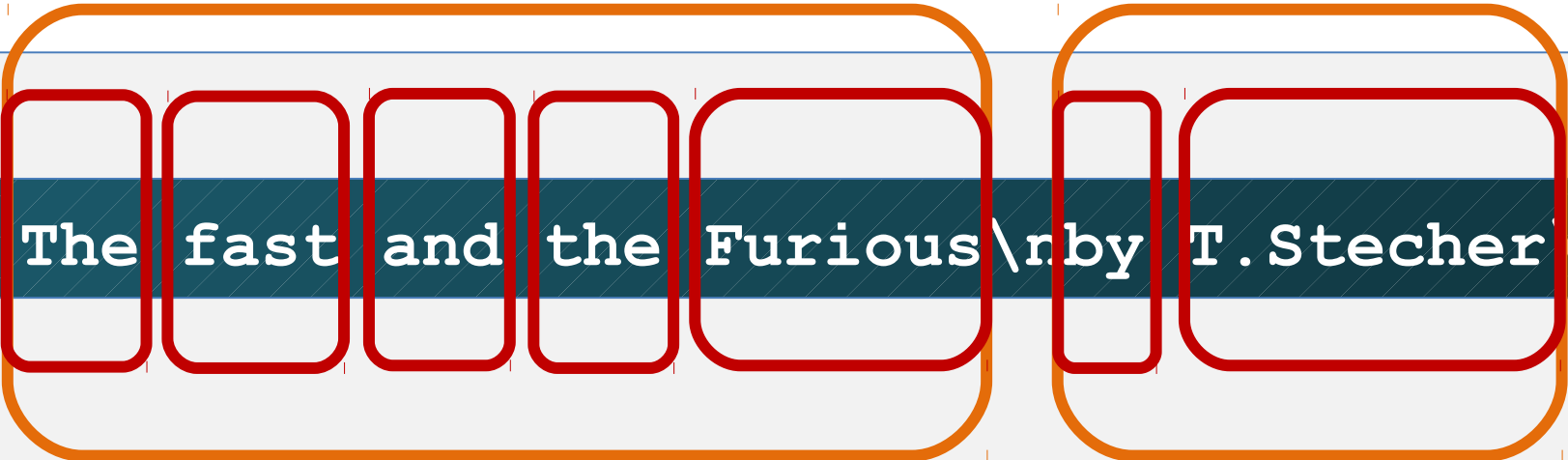
`byte[ ]`

`int position`

`int limit`

ByteBuffer

AsciiLine & AsciiWord



The fast and the Furious\r\nby T. Stecher\r\n

The diagram illustrates a ByteBuffer containing a string. The string is "The fast and the Furious\r\nby T. Stecher\r\n". The text is displayed in a monospaced font on a dark blue background. The string is divided into two parts by a large orange rounded rectangle. The first part, "The fast and the Furious\r\n", is enclosed in a red rounded rectangle. The second part, "by T. Stecher\r\n", is also enclosed in a red rounded rectangle. The entire content is within a larger orange rounded rectangle.

ByteBuffer

Die ASU

NIO statt E10

Zeilen selbst zählen

Zeilen selbst „zerlegen“

Nur EINE AsciiLine

Nur EIN AsciiWord

Lediglich Indizes setzen

Keine Konvertierung in Strings



Äpfel mit Birnen verglichen?



Die faire Gegenprobe

```
AsciiParser parser = new AsciiParser(ASCII_FILE);
try {
    parser.openFile();
    int wordCnt = 0;
    AsciiWord word = parser.getNextWord();
    while (word != null) {
        word.asString();
        wordCnt++;
        word = parser.getNextWord();
    }
    System.out.println("#Words=" + wordCnt);
} finally {
    parser.close();
}
```

Die faire Gegenprobe

Test	#read	CPU [millis]	Dauer [millis]
FileInputStream	64000	390	401
AsciiParser.getNextLine()	64001	1528	1542
AsciiParser.getNextWord()	64001	3198	3208
AsciiParser.getNextWord().asString()	64001	7254	7564
LineNumberReader(FileReader) + split()	8192000	13353	13526

Aber...

Diese Fairness wird in der Praxis nicht benötigt

Haupteinsatz

Konvertierung in long

Konvertierung in double

Suche des x-ten Wertes



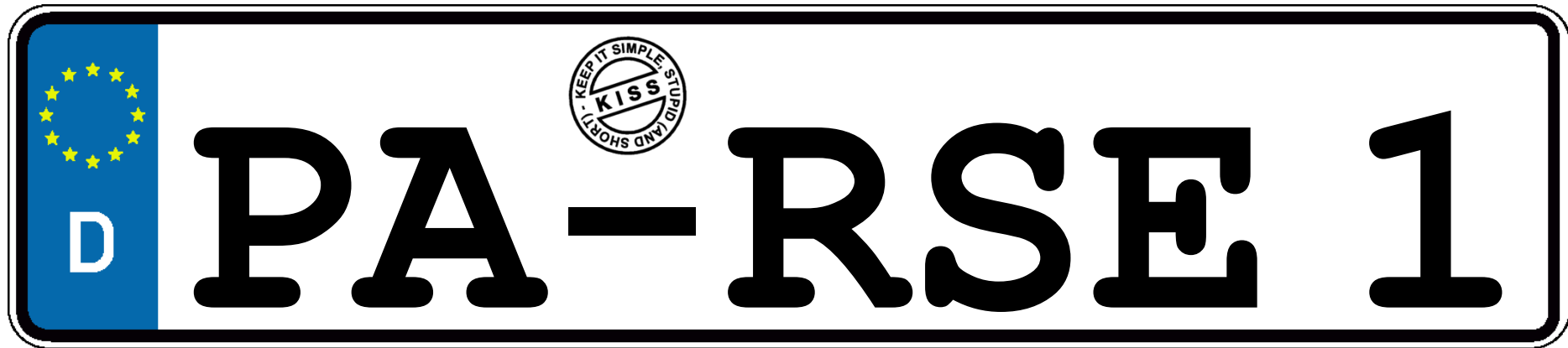
Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Anwendungsfälle

	Elapsed alt	Elapsed neu	Speedup
TimeHistory	13:29	3:37	73 %
ConvertResults	22:46	14:49	37 %
S1-WD	41:56	34:50	17 %

Zulassung



Danke

Rainer Büsch (rainer.buesch@tntsoft.de)

für die tolle Zusammenarbeit

Detlev Müller (deti@ta-deti.de)

für die Ferrari Fotos



Bundesanstalt für
Geowissenschaften
und Rohstoffe

GEOZENTRUM HANNOVER

Fragen?

Vielen Dank!

thomas.stecher@bgr.de