

Parallel Programming

0024

Matrix Multiplication

Spring Semester 2010

Gesucht

Übungen von

- Akousz
- Loluca
- Lruosch
- Predalis
- Tobiasho
- Zuestm

Webseite:

<http://www.lantersoft.ch/de/parallelprog/parallelprog.php>

Parallel matrix multiplication

- Data partitioning based on
 - Output matrix C
 - Input matrix A and input matrix B

Output partitioning (2 threads)

Thread 0

```
for (i=0; i<N/2; i++) {  
    for (j=0; j<N; j++) {  
        for (k=0; k<N; k++) {  
            c[i][j] += a[i][k]*b[k][j];  
        }  
    }  
}
```

Thread 1

```
for (i=N/2; i<N; i++) {  
    for (j=0; j<N; j++) {  
        for (k=0; k<N; k++) {  
            c[i][j] += a[i][k]*b[k][j];  
        }  
    }  
}
```

Input partitioning

- Thread 0

```
for (i=0; i<N; i++) {  
    for (j=0; j<N; j++) {  
        synchronized (lock[i][j]) { for (k=0; k<N/2; k++) {  
            c[i][j] += a[i][k]*b[k][j];  
        }  
    }  
}
```

- Thread 1

```
for (i=0; i<N; i++) {  
    for (j=0; j<N; j++) {  
        synchronized (lock[i][j]) { for (k=N/2; k<N; k++) {  
            c[i][j] += a[i][k]*b[k][j];  
        }  
    }  
}
```

Overhead ?

A complete row is locked.

Actual lock contention will be moderate to low.

In practice the slow-down (resp to 1st solution) is moderate (few %)

Input partitioning, fine grain locking

- Thread 0

```
for (i=0; i<N; i++) {  
    for (j=0; j<N; j++) {  
        for (k=0; k<N/2; k++) {  
            synchronized (lock[i][j]) { c[i][j] += a[i][k]*b[k][j];  
            }  
        }  
    }  
}
```

- Thread 1

dto

- Significant overhead

- About 3 x slower than coarse-grain locking





Eine Java Applikation startet in der Methode

`init()`

`main()`

`init(String[] args)`

`main(String[] args)`



Eine Java Applikation startet in der Methode

`init()`

`main()`

`init(String[] args)`

`main(String[] args)`



Um einen Thread `t` parallel zu starten muss welche Methode aufgerufen werden?

`t.run()`

`t.main()`

`t.start()`

`t.init()`



Um einen Thread `t` parallel zu starten muss welche Methode aufgerufen werden?

`t.run()`

`t.main()`

`t.start()`

`t.init()`



Welcher Ausdruck ist in einem throw-Statement erlaubt?

throw null;

throw Throwable();

throw Eiffel();

throw new Object();



Welcher Ausdruck ist in einem throw-Statement erlaubt?

`throw null;`

`throw Throwable();`

`throw Eiffel();`

`throw new Object();`



In welchem State ist ein neuer Thread `t` nach dem Aufruf `t.run()`?

WAITING

BLOCKED

NEW

RUNNABLE



In welchem State ist ein neuer Thread `t` nach dem Aufruf `t.run()`?

WAITING

BLOCKED

NEW

RUNNABLE



Was droht, wenn in einem System `notify()`, statt `notifyAll()` aufgerufen wird?

Deadlock

Livelock

Starvation

Race Condition



Was droht, wenn in einem System `notify()`, statt `notifyAll()` aufgerufen wird?

Deadlock

Livelock

Starvation

Race Condition



Welche dieser Exceptions ist eine Checked Exception?

RuntimeException

InterruptedException

NullPointerException

IllegalThreadStateException



Welche dieser Exceptions ist eine Checked Exception?

RuntimeException

InterruptedException

NullPointerException

IllegalThreadStateException



Kann die main() Methode überschrieben werden?

Ja

Nein

Nur, wenn sie in einer abstrakt. Klasse liegt

Nur, wenn sie in einer statischen Klasse liegt



Kann die main() Methode überschrieben werden?

Ja

Nein

Nur, wenn sie in einer abstrakt. Klasse liegt

Nur, wenn sie in einer statischen Klasse liegt



Welche der folgenden Aussagen ist falsch?

Eine innere Klasse kann von ihrer äusseren Klasse erben

I.d.R. sollte eher Runnable impl., statt Thread erweitert werden

Volatile Variablen können von Threads gecached werden

Java ist auch eine Insel



Welche der folgenden Aussagen ist falsch?

Eine innere Klasse kann von ihrer äusseren Klasse erben

I.d.R. sollte eher Runnable impl., statt Thread erweitert werden

Volatile Variablen können von Threads gecached werden

Java ist auch eine Insel



Was bewirkt `synchronized(this)` { ... }?

Alle Methoden der Klasse werden gesperrt

Alle Methoden mit `synchronized` im Kopf werden gesperrt

Alle non-static Methoden der Klasse werden gesperrt

Alle non-static Methoden mit `synchronized` im Kopf werden gesperrt




Was bewirkt `synchronized(this)` { ... }?

Alle Methoden der Klasse werden gesperrt

Alle Methoden mit `synchronized` im Kopf werden gesperrt


Alle non-static Methoden der Klasse werden gesperrt

Alle non-static Methoden mit `synchronized` im Kopf werden gesperrt



**Wie viele Klassen Arten gibt es?
(private, public, protected,
abstract, final, static, ...)
[ohne strictfp]**

???



**Wie viele Klassen Arten gibt es?
(private, public, protected,
abstract, final, static, ...)
[ohne strictfp]**

34