

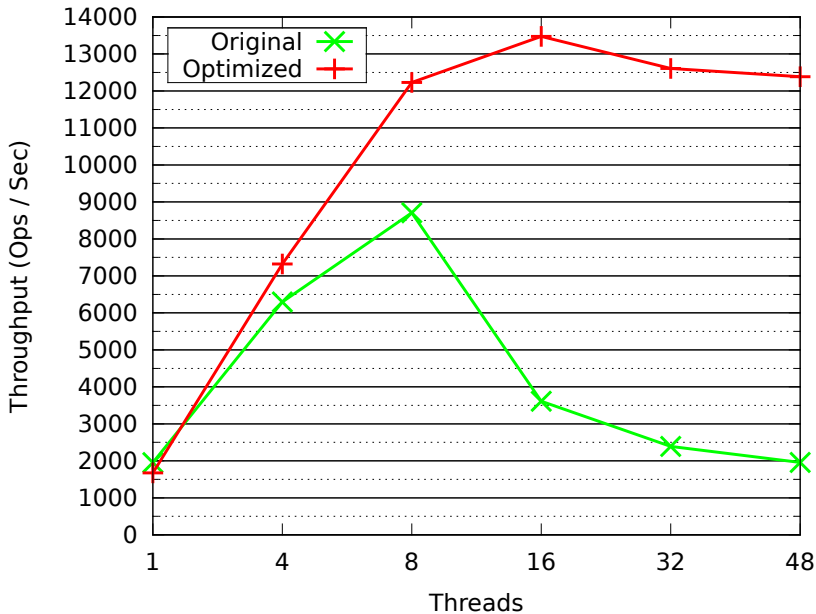
# Profiling and tuning the performance of an STM-based concurrent program

Luís Pina    João Cachopo  
([luis.pina@ist.utl.pt](mailto:luis.pina@ist.utl.pt)    [joao.cachopo@ist.utl.pt](mailto:joao.cachopo@ist.utl.pt))

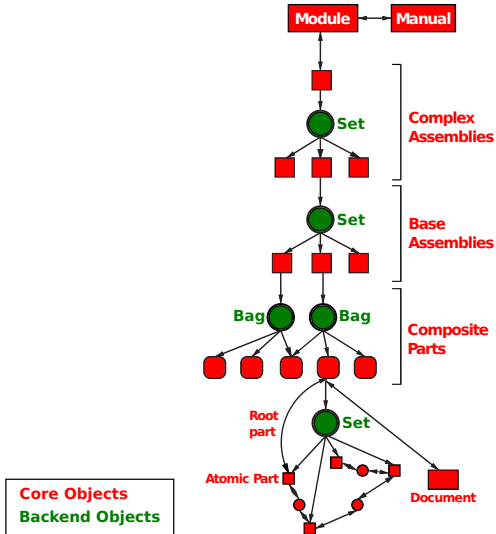
INESC-ID/Technical University of Lisbon

October 23, 2011

Workshop on Transitioning to MultiCore  
(TMC'11)



# STMBench7



# JVSTM

## Java Versioned Software Transactional Memory

```
01 class Counter {  
02   int c = 0;  
03  
04   void inc() { c = c + 1; }  
05   void reset() { c = 0; }  
06   int get() { return c; }  
07 }
```

# JVSTM

## Java Versioned Software Transactional Memory

```
01 class Counter {  
02     int c = 0;  
03  
04     @Atomic void inc() { c = c + 1; }  
05     @Atomic void reset() { c = 0; }  
06     @Atomic int get() { return c; }  
07 }
```

# JVSTM

## Java Versioned Software Transactional Memory

```
01 class Counter {  
02     final VBox<Integer> c = new VBox<Integer>(0);  
03  
04     @Atomic void inc() { c = c + 1; }  
05     @Atomic void reset() { c = 0; }  
06     @Atomic int get() { return c; }  
07 }
```

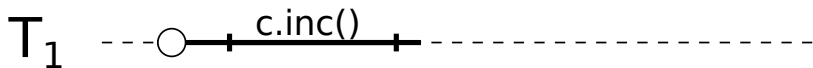
# JVSTM

## Java Versioned Software Transactional Memory

```
01 class Counter {
02     final VBox<Integer> c = new VBox<Integer>(0);
03
04     @Atomic void inc() { c.put(c.get()+1); }
05     @Atomic void reset() { c.put(0); }
06     @Atomic int get() { return c.get(); }
07 }
```

# JVSTM

## Transactions



VBox	(Most Recent) Value	(Most Recent) Version
c	0	0

Read-Set

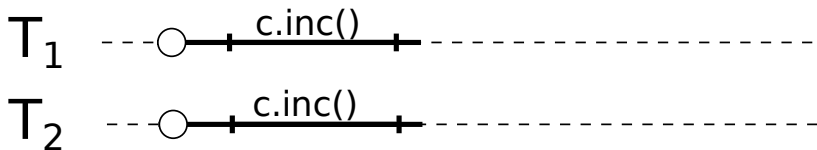
Tx	VBox	Version
$T_1$	c	0

Write-Map

Tx	VBox	New Value
$T_1$	c	1

# JVSTM

## Transactions



VBox	(Most Recent) Value	(Most Recent) Version
c	0	0

Read-Set

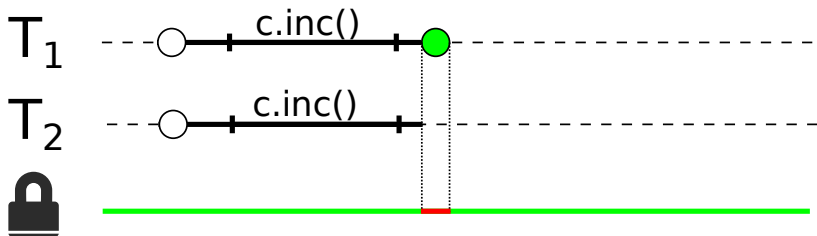
$T_x$	VBox	Version
$T_1$	c	0
$T_2$	c	0

Write-Map

$T_x$	VBox	New Value
$T_1$	c	1
$T_2$	c	1

# JVSTM

## Transactions



VBox	(Most Recent) Value	(Most Recent) Version
c	0	0

Read-Set

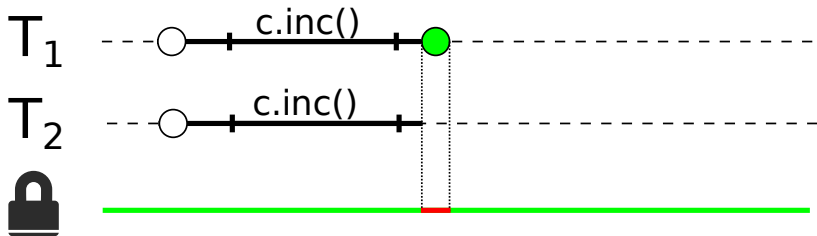
Tx	VBox	Version
$T_1$	c	0
$T_2$	c	0

Write-Map

Tx	VBox	New Value
$T_1$	c	1
$T_2$	c	1

# JVSTM

## Transactions



VBox	(Most Recent) Value	(Most Recent) Version
c	1	1

### Read-Set

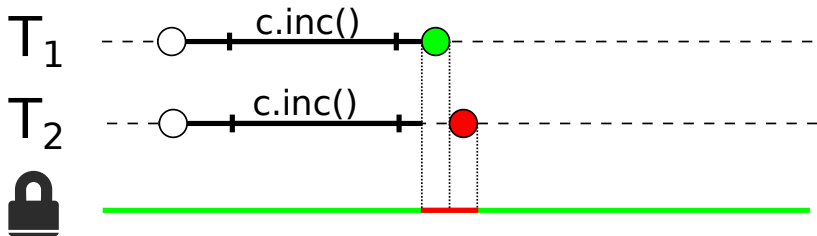
Tx	VBox	Version
$T_2$	c	0

### Write-Map

Tx	VBox	New Value
$T_2$	c	1

# JVSTM

## Transactions



VBox	(Most Recent) Value	(Most Recent) Version
c	1	1

Read-Set

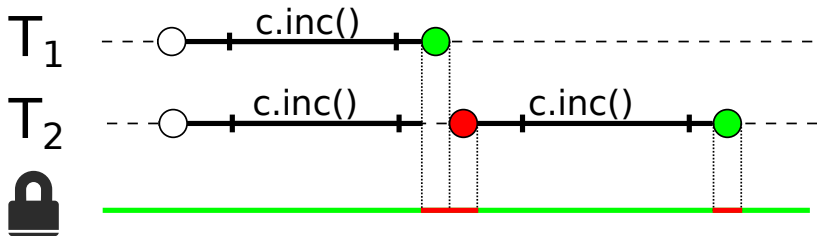
Tx	VBox	Version
$T_2$	c	0

Write-Map

Tx	VBox	New Value
$T_2$	c	1

# JVSTM

## Transactions



VBox	(Most Recent) Value	(Most Recent) Version
c	1	1

Read-Set

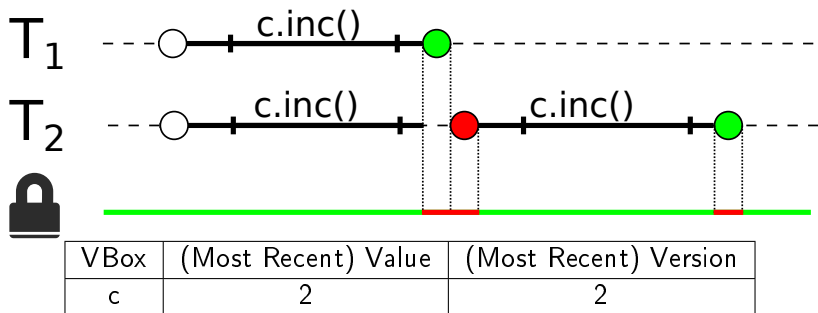
Tx	VBox	Version
$T_2$	c	1

Write-Map

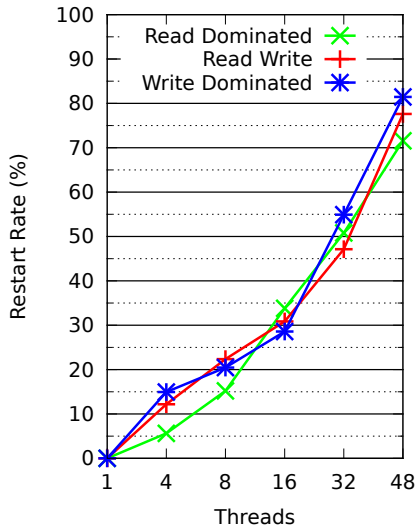
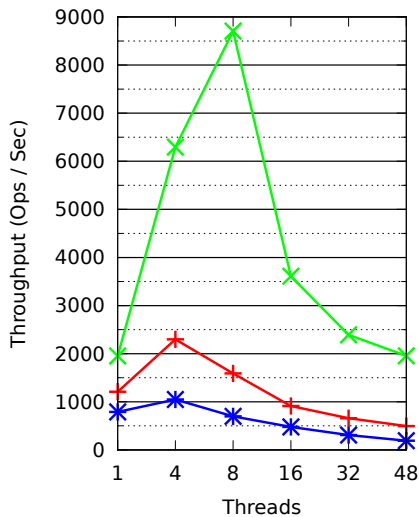
Tx	VBox	New Value
$T_2$	c	2

# JVSTM

## Transactions



## Conflicts and Performance



# Detecting Conflicts

## Number of Conflicts

	8	16	32	48
AtomicPartImpl::y	521.417	1.457.949	18.477.298	64.052.709
AtomicPartImpl::x	521.417	1.457.949	18.477.298	64.052.709
LargeSetImpl::elements	706.930	961.391	882.978	93.991
LargeSetImpl::count	706.587	960.635	870.216	47.154
ManuallImpl::text	15.765	12.144	13.838	16.850
VQueue::front	30.254	39.296	42.617	18.245
VIndex::index	46.363	56.586	50.038	15.210
BuildDate::buildDate	80	861	4.025	5.491
VQueue::rear	2.455	5.377	3.491	879
AtomicPartImpl::to	0	200	200	300

## Reducing Conflicts

- Per Transaction Boxes
- Transactional Futures
- Restartable Transactions

# Reducing Conflicts

Per Transaction Boxes

	8	16	32	48
AtomicPartImpl::y	521.417	1.457.949	18.477.298	64.052.709
AtomicPartImpl::x	521.417	1.457.949	18.477.298	64.052.709
LargeSetImpl::elements	706.930	961.391	882.978	93.991
LargeSetImpl::count	706.587	960.635	870.216	47.154
ManuallImpl::text	15.765	12.144	13.838	16.850
VQueue::front	30.254	39.296	42.617	18.245
VIndex::index	46.363	56.586	50.038	15.210
BuildDate::buildDate	80	861	4.025	5.491
VQueue::rear	2.455	5.377	3.491	879
AtomicPartImpl::to	0	200	200	300

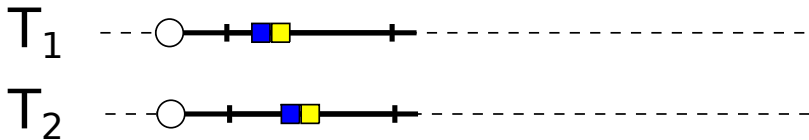
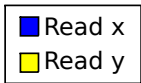
# Reducing Conflicts

## Per Transaction Boxes

```
01 class AtomicPartImpl {
02     final VBox<Integer> x, y;
03
04     void swap() {
05         int tmp = x.get();
06         x.put(y.get());
07         y.put(tmp);
08     }
09
10     int getX() { return x.get(); }
11     int getY() { return y.get(); }
12 }
```

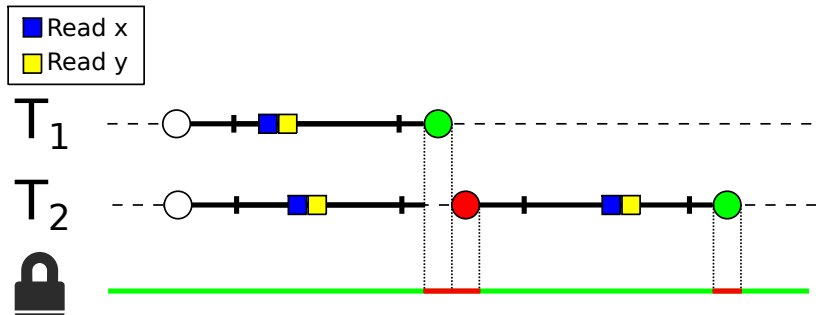
# Reducing Conflicts

Per Transaction Boxes



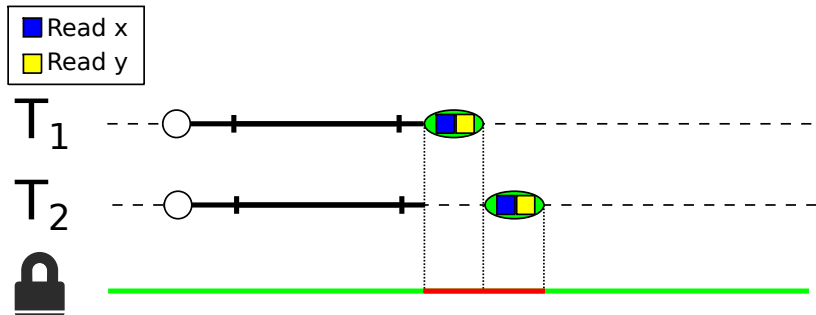
# Reducing Conflicts

Per Transaction Boxes



# Reducing Conflicts

Per Transaction Boxes



# Reducing Conflicts

Per Transaction Boxes

	8	16	32	48
AtomicPartImpl::y	521.417	1.457.949	18.477.298	64.052.709
AtomicPartImpl::x	521.417	1.457.949	18.477.298	64.052.709
LargeSetImpl::elements	706.930	961.391	882.978	93.991
LargeSetImpl::count	706.587	960.635	870.216	47.154
ManuallImpl::text	15.765	12.144	13.838	16.850
VQueue::front	30.254	39.296	42.617	18.245
VIndex::index	46.363	56.586	50.038	15.210
BuildDate::buildDate	80	861	4.025	5.491
VQueue::rear	2.455	5.377	3.491	879
AtomicPartImpl::to	0	200	200	300

# Reducing Conflicts

Per Transaction Boxes

	8	16	32	48
Manuallmpl::text	183.317	230.422	186.706	345.556
VIndex::index	11.021	11.604	34.967	20.342
VQueue::front	18.019	14.612	31.713	21.623
BuildDate::buildDate	517	1.332	4.806	9.744
VQueue::rear	1.050	1.265	3.338	1.513
VLinkedSet::entries	894	711	796	1.659
AtomicPartImpl::to	0	200	200	280
AtomicPartImpl::y	31	104	236	322
AtomicPartImpl::x	31	104	236	322
DocumentImpl::text	16	11	40	161
LargeSetImpl::elements	0	0	0	0
LargeSetImpl::count	0	0	0	0

# Reducing Conflicts

## Transactional Futures

```
01 class ManualImpl {
02     VBox<String> string;
03
04     @Atomic
05     int capitalize() {
06
07
08         int ret = string.get().countChar('a');
09         string.put(string.get().replace('a', 'A'));
10         return ret;
11
12
13     }
14 }
```

# Reducing Conflicts

## Transactional Futures

```
01 class ManualImpl {
02     VBox<String> string;
03
04     @Atomic
05     TransFuture<Integer> capitalize() {
06         return new TransFuture<Integer>(){
07             Integer execute() {
08                 int ret = string.get().countChar('a');
09                 string.put(string.get().replace('a', 'A'));
10                 return ret;
11             }
12         };
13     }
14 }
```

# Reducing Conflicts

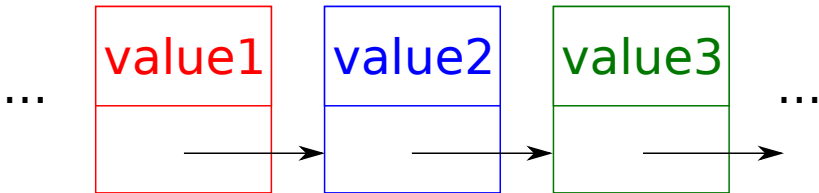
## Transactional Futures

	8	16	32	48
VIndex::index	90.485	73.293	72.378	75.668
VQueue::front	86.066	55.763	56.539	57.300
BuildDate::buildDate	2.044	5.605	11.280	16.307
VQueue::rear	6.365	6.696	6.595	6.620
VLinkSet::entries	6.686	1.513	760	935
BagImpl::elements	48	70	116	186
AtomicPartImpl::y	100	113	262	309
AtomicPartImpl::x	100	113	262	309
BaseAssemblyImpl::components	24	25	44	89
AssemblyImpl::superAssembly	24	42	85	140
ManuallImpl::text	0	0	0	0

# Reducing Conflicts

## Restartable Transactions

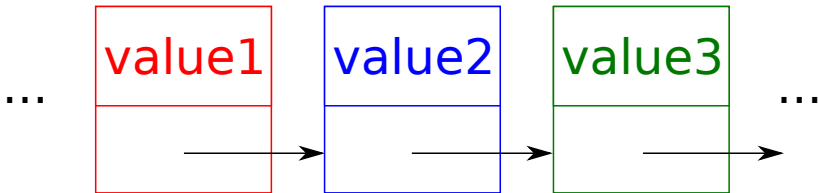
```
01  
02 //Tx1:  
03 if (set.contains(value3))  
04   doStuff();  
05  
06 //Tx2:  
07 set.remove(value2);
```



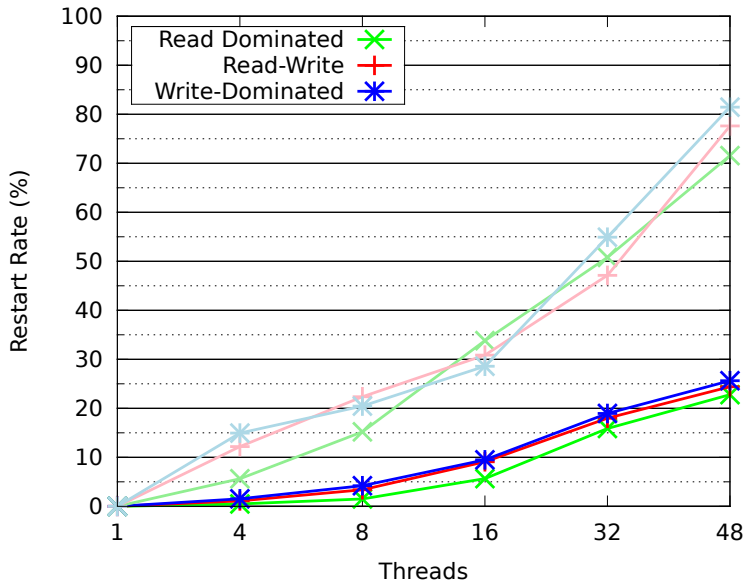
# Reducing Conflicts

## Restartable Transactions

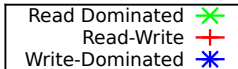
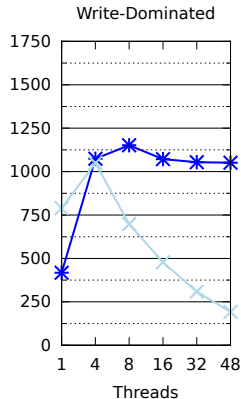
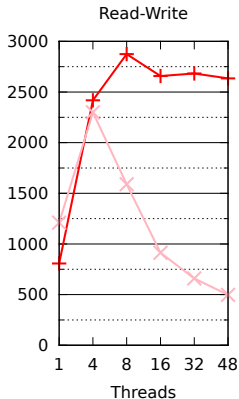
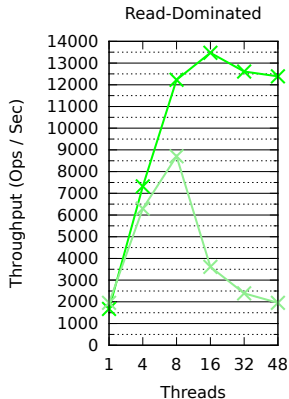
```
01 class LinkedListSet<E> {  
02   ...  
03   @Restartable  
04   boolean contains(E e) {  
05     ...  
06   }  
07 }
```



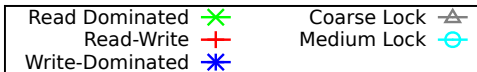
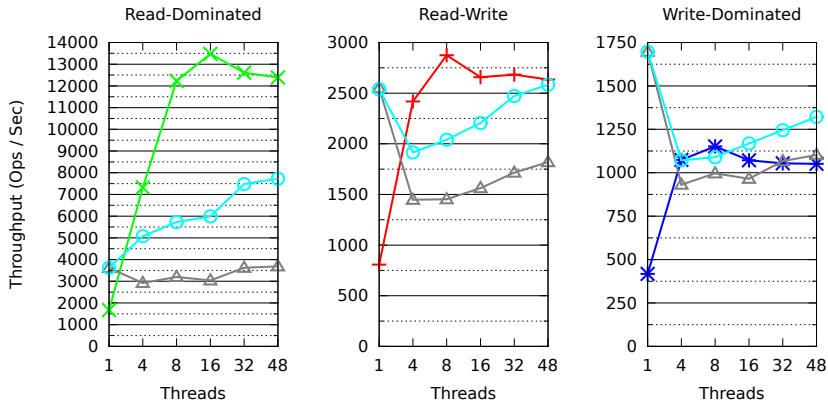
## Results



# Results



# Results



# Conclusion

- Conflict detection technique
- Conflict reduction
  - PerTxBoxes
  - Transactional Futures
  - Restartable Transactions

Thank you!

Q&A