

Using Java Pathfinder to test Algorithm Implementation

Mohammed-Ali Khan

Department of Computer Science and Engineering, York University,
4700 Keele Street, Toronto, M3J 1P3
Ontario, Canada

m23khan@cse.yorku.ca

Presented on: April 05, 2011

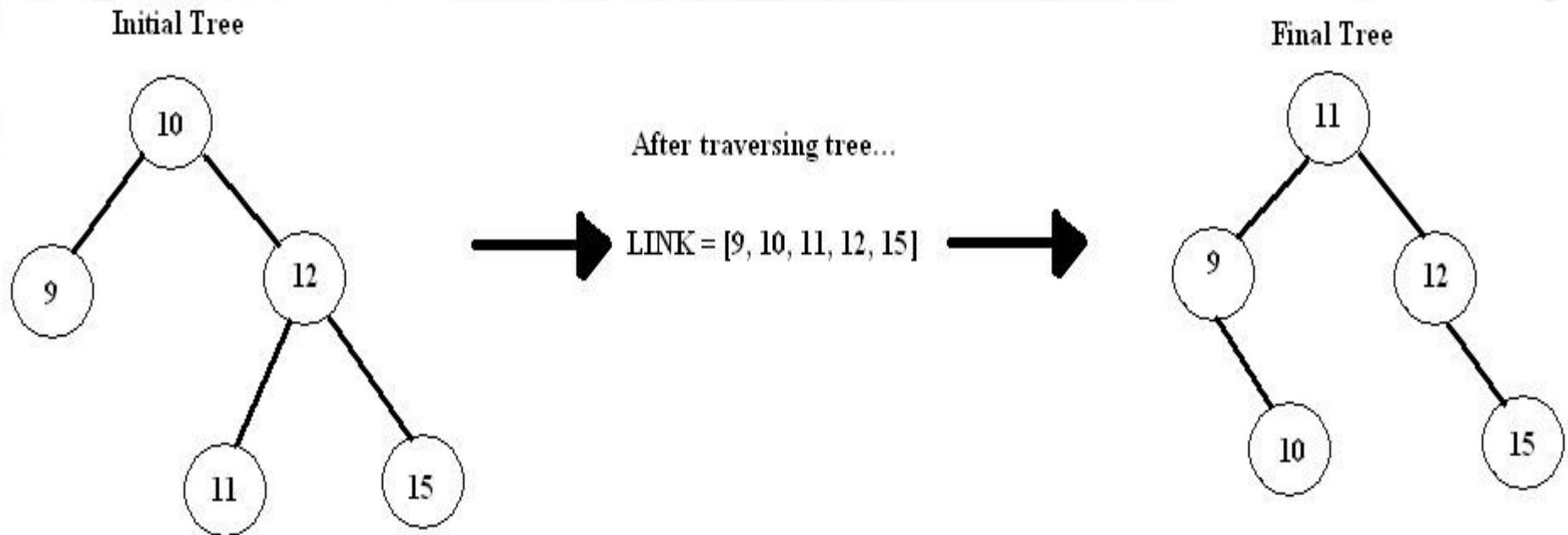
Agenda

- Recap Balancing Algorithms (S and P1)
- Implementation overview - class Diagram
- JPF Testing and results
- Observations



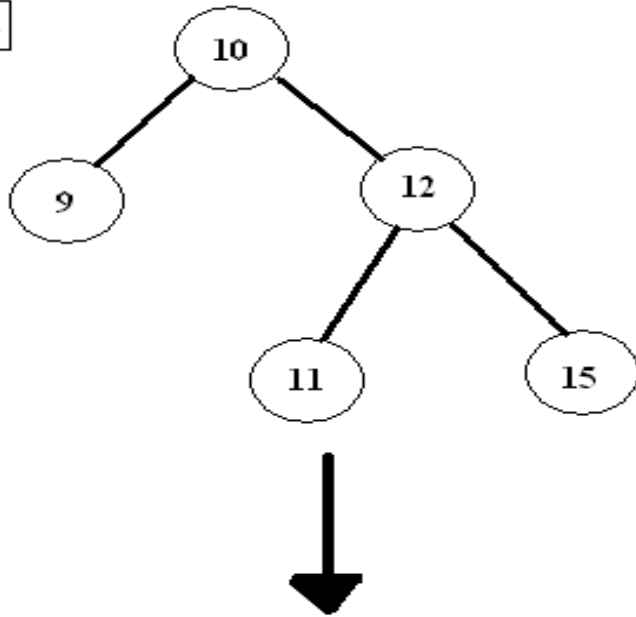
Balancing Binary Tree

- Using a sequential algorithm (algorithm S) or a parallel algorithm (algorithm P1), balance a given binary search tree.

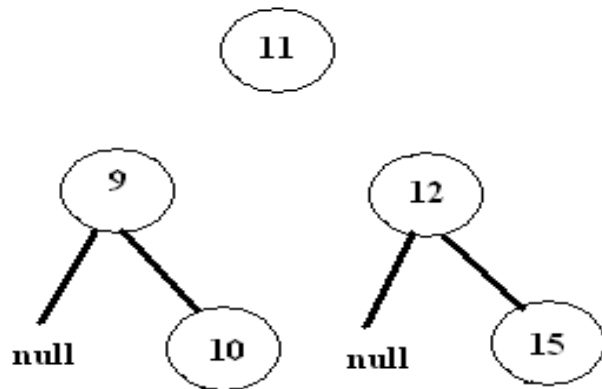


Visualizing Concurrency for P1

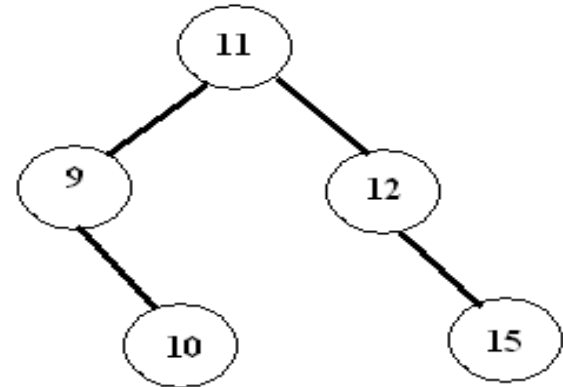
1.



2.



3.



Implementation Overview

Highlights:

- Synchronization done via keyword *synchronized* and volatile variable
- Callable Interface implemented in *runnable* class.
- Thread pool used to control number of threads.

JPF Testing and Results

Tests performed:

- Running program with 3 threads - runtime for JPF
- Running program with 4 threads - runtime for JPF
- Deadlock testing
- Race Condition Detection
- Testing using DFS and BFS Hueristic search

Test 1 - Testing with 3 threads

- Testing with 3 threads took approximately 11 minutes.
- Runtime statistics:

```
states:                new=860500, visited=990286,  
backtracked=1850785, end=64626
```

```
search:                maxDepth=65, constraints hit=0
```

```
choice generators:    thread=860500 (signal=0,  
lock=53967, shared ref=250667), data=0
```

```
heap:                 new=2759486, released=7161690,  
max live=915, gc-cycles=1850786
```

```
instructions:        571619679
```

```
loaded code:         classes=142, methods=1806
```

Test 2 - Testing with 4 threads

- Took approx. 70 minutes to run JPF.
- Likely cause:
 - state space explosion.
- Run time statistics:

```
states:                new=5797424, visited=8970961,  
backtracked=14768384, end=580149  
search:                maxDepth=71, constraints hit=0  
choice generators:    thread=5797424 (signal=0,  
lock=280007, shared ref=1672947), data=0  
heap:                 new=19899451, released=64131976,  
max live=926, gc-cycles=14768385  
instructions:         -575855300  
loaded code:          classes=145, methods=1980
```


Test 3 - Deadlock Testing

- Set Thread pool to 2 threads max.
- JPF results:

```
gov.nasa.jpjvm.NotDeadlockedProperty  
deadlock encountered:
```

```
...
```

```
call stack:
```

```
    at RunnableP1.call (RunnableP1.java:58)  
    at RunnableP1.call (RunnableP1.java:29)
```

```
...
```

```
===== results  
error #1: gov.nasa.jpjvm.NotDeadlockedProperty  
"deadlock encountered:  thread  
index=0,name=main,s..."
```

```
...
```

Test 4 - Race Condition Detection

- Introduce private global variable (counter) in a class with *synchronized method*.
- Increment the variable in the *synchronized* method.
- Remove the keyword *synchronized* from the method.
- JPF Result:
...

```
===== error #1
gov.nasa.jpf.listener.PreciseRaceDetector
race for field AlgorithmP1@23e.sharedVar
  pool-1-thread-1 at
AlgorithmP1.GROW(AlgorithmP1.java:57)
           "(AlgorithmP1.java:57)" : putfield
  pool-1-thread-2 at
AlgorithmP1.GROW(AlgorithmP1.java:57)
           "(AlgorithmP1.java:57)" : getfield
...
```

Test 5 - Testing using DFS and BFS Hueristic search

- JPF returned error messages when program ran with DFS and BFS hueristic search.
- Unfortunately message not very clear about error location...

```
===== search started: 4/4/11 9:52 PM  
java.lang.NullPointerException  
    at  
gov.nasa.jpf.jvm.serialize.CFSerializer.processRefer  
ence(CFSerializer.java:77)  
    at  
gov.nasa.jpf.jvm.serialize.FilteringSerializer.seria  
lizeThread(FilteringSerializer.java:273)  
  
...
```

Conclusion

- Java Pathfinder is an excellent tool to help debug Java applications.
- Has real potential to become one of the hottest Java debug tools in commercial settings.
- For quick hints and tips from experts, please visit Java Pathfinder's Google group webpage at:
<http://groups.google.com/group/java-pathfinder>

References

- Java Pathfinder website: <http://babelfish.arc.nasa.gov/trac/jpf/>
- Java Pathfinder Google Group:
<http://groups.google.com/group/java-pathfinder>

QUESTIONS...

