Part I
Garbage Collection Modeling: Progress Report

Peter Libič
Department of Distributed and Dependable Systems
http://d3s.mff.cuni.cz
CHARLES UNIVERSITY PRAGUE
Faculty of Mathematics and Physics
Last Time (May 2010)

- Collection algorithms overview
- Initial models
  - Two versions
    - “Simple”
    - “Generational”
  - “Ugly equations with lots of sums”
- Based on simulation using lifetimes of the objects
- Simple model has a nice simplification that seems to work well
Today

- More on motivation
  - Why are we doing it?
  - “use cases”
- Current results
- Challenges
  - [possible brainstorming]
Recall: Throughput Collector in HotSpot VM

- Generational collector
  - 2 generations
- Young generation
  - 3 spaces
  - Eden + 2 Survivors
- Allocations in Eden
- Young GC collects all young spaces
- Full GC empties young generation
Hello everyone,

We've recently upgraded our Weblogic webapp to Weblogic 9.2.1 which also involved upgrading from jdk 1.4.2 to 1.5.0_22. Java 6 isn't an option with Weblogic 9.2, and the upgrade to Weblogic 10 isn't until next year, so Java 6 is not an option for me at this point in time.

I had the 1.4.2 tuned to the extreme:

```
JAVA_ARGS=-Xms2816m -Xmx2816m -XX:NewSize=384m -XX:MaxNewSize=384m -XX:CompileThreshold=3000
-Djavax.net.setSoTimeout=20000 \n-XX:LargePageSizeInBytes=4m -XX:+UseMPSS -Xss128k -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:+CMSParallelRemarkEnabled \n-XX:+UseCMSCompactAtFullCollection -Xloggc:gc.out \n-verbose:gc -XX:+PrintGCTimeStamps -XX:+PrintGCDetails -XX:MaxPermSize=92m
```

With our upgrade, the ergonomics of the application changed some, so I just specified Xmx and Xms at 3GB, and let it run to see what happened. Remarkably, things were actually pretty good - minor GC's are faster, and major GC's only occur about once or twice an hour. However, those major GC's are taking from 12-20seconds, which I can't let our website users endure.

Before I go down the road of tuning things, does anyone have any tips for me? I can afford at most 2 or 3 seconds of pause time at once, and would prefer to keep it under 2 seconds if possible. I need the 2GB of tenured to be able to cache all the objects that I need to ensure good site performance.

The ergonomics look kinda cool, but I'm wondering if that large of tenured generation + my low pause requirement is just too much to ask from the throughput collector. Am I destined to go back to hand tweaking the CMS collector?

Justin
Possible use cases / goals

- Suggestion of optimal generation sizes or heap size
- Composition
  - Dependent
    - What happens if we add a component here?
  - Independent
    - Is it better to run two tasks in one JVM or not?
- Port to different platform
  - With stable traces
Simple Model

\[ HS = \left( \sum_{j=n_{i-1}+1}^{n_i} SIZE[j] \right) + \left( \sum_{j \in \{1 \ldots n_{i-1}\} \text{ DEATH}[j] \geq n_{i-1}} SIZE[j] \right) \]

\[ \Delta n = \frac{HS}{OS} - LT \]
# Simple Model I - Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Prediction ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>antlr</td>
<td>0.88-0.95</td>
</tr>
<tr>
<td>bloat</td>
<td>0.95-1.01</td>
</tr>
<tr>
<td>fop</td>
<td>0.94-0.96</td>
</tr>
<tr>
<td>hsqldb</td>
<td>0.75-0.77</td>
</tr>
<tr>
<td>jython</td>
<td>0.79-1.00</td>
</tr>
<tr>
<td>luindex</td>
<td>0.97-1.08</td>
</tr>
<tr>
<td>lusearch</td>
<td>1.55-1.70</td>
</tr>
<tr>
<td>pmd</td>
<td>0.89-1.07</td>
</tr>
</tbody>
</table>

Prediction ratio = time between collections (measured) / tbc (model)

Measured on JikesRVM
Where To Get The Data?

- The simulation needs a lot of data
  - Object trace
    - Size
    - Lifetime
- Not easy to get lifetimes for Java object
- Two approaches:
  - Bruteforce
    - Relatively simple, but very big overhead
  - Merlin algorithm
    - Complex, needs to go deep into VM, but faster
Bruteforce Lifetime Tracer

- Bytecode instrumentation
  - Every NEW (and friends) instruction is instrumented by a call to a static method
  - This static method passes the information to JVMTI agent – via JNI
  - Done by Milan Jovic from USI Lugano

- JVMTI agent traces the JNI calls and logs them
- Postprocessing tool transforms the log into lifetime trace

- http://d3s.mff.cuni.cz/software/jolt
Comparison of predicted and measured GC counts

- Measured value
- Value from simulation

Measured on IBM J9 VM
Generational Model

- Tries to handle HotSpot's throughput collector
- Simulation-based
  - We do have a version in equations
    - Extremely complex
- Relatively precise young GC frequency
- Not very precise for old GC frequency and young GC times
- Partially precise old GC times
Comparison of predicted and measured Minor GC counts

○ Measured value
△ Value from simulation
Generational Model - Old GC Frequency

Comparison of predicted and measured Full GC counts

- Measured value
- Value from simulation

GC count

20 40 60 80 100 120 140

fop-long-12-44
top-long-12-86
top-long-12-176
top-long-16-44
top-long-16-66
top-long-16-88
top-long-16-2044
top-long-20-66
top-long-20-176
top-long-20-28-44
top-long-28-44
top-long-28-long-176
top-long-28-352
top-long-44-44
top-long-44-66
top-long-44-176
top-long-44-352
top-long-76-66
top-long-76-176
Comparison of predicted and measured GC times

- Measured value
- Value from simulation
How useless is this?

- Almost all the absolute numbers are incorrect ...
- Question:
  - How does it behave in situations for which the model is intended?
- Scenario:
  - Fixed total heap size
  - What is the best generation sizing within this heap?
Testing Scenario - GC Frequency

X axis
- execution ID [1=minimum Young Heap size; 31=maximum; Total size fixed]

Y axis
- number of collections
Testing Scenario - GC Times

Axes interpretation the is same as in previous slide.
Why?

- Fragmentation?
- Pretenuring?
- Early GC triggering?
- One of hundreds of other optimizations?

- What to do with that?
Possible Future Challenges

• Composition
  • Model is based on lifetime traces
  • May work for deterministic workloads

• What happens with the traces when new component is added into sequence?
• What happens when the tasks are run in parallel?

• Composition of traces?
Conclusion

- The model still needs a lot of improvement
  - I think it is getting better (but slowly)
- There is still a lot of interesting work :-)

- Shouldn't we introduce delete/free into Java?
Questions?
Part II

USI Lugano: My Experience

Peter Libič

Department of Distributed and Dependable Systems

http://d3s.mff.cuni.cz

CHARLES UNIVERSITY PRAGUE
Faculty of Mathematics and Physics
Lugano, Switzerland
Universita della Svizzera Italiana

- Founded in 1996
- Four faculties
  - Architecture, Communication, Economics, Informatics
- 2157 Students, from which 147 in PhD program
- Mini-campus in Lugano
- Architecture in Mendrisio
Facolta di Szienze Informatiche

- Youngest faculty – 2004
- 59 bachelor students
- 39 master's students
- 87 PhD students
- 21 post-docs
- 21 professors
- Many research topics
  - Distributed systems, computer graphics, computational geometry, performance, ...
How Did I Get There?

- Sciex-NMS\textsuperscript{ch} Program
- Aimed for post-docs and PhD students from new EU member states
- Funds stays within Swiss research teams
- Stays for 6-24 months
- For Czech Republic:
  - Application: end of October
  - Notification: end of March
- http://www.sciex.ch
What Was it Like?

- Friendly environment (almost like here)
- Constructive environment

- Not enough space

- People from all over the world
  - Brazil, Japan, Pakistan, India, USA, Serbia ...
Life in Switzerland

- Very expensive
  - Except for milk products :)
- Rules and regulations everywhere
- Life is rather slow

- Beautiful nature
  - The weekends were almost perfect time for me
• In general I liked it there
  • But I am glad to be back :) 

• Grazie per la vostra attenzione!

• Domande?