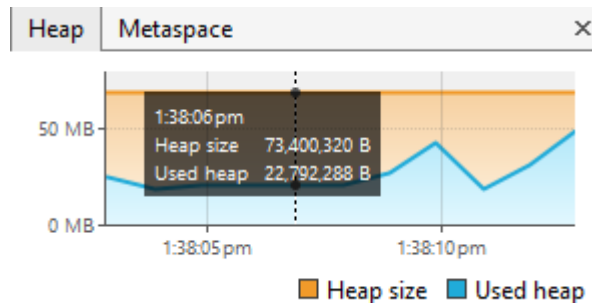


Report for Memory Usage



For both layouts, I have used a max size of 50,000 Logs because this is the highest amount my computer can run without going buggy.

The first spike in memory usage is from the pattern layout. This is a spike of around 23 mb.

The second spike in memory usage is from the velocity layout, This is a spike of around 30 mb.

The pattern spike uses around 7mb less per 50,000 tests which is a significant amount.

The max size for a long variable is over 9 quintillion. This means that at full storage, the velocity pattern would use $(9 \text{ quintillion} / 50,000) * 7 = 1.26e+15 \text{ mb}$ more than the pattern layout.

This would not be suitable at all for larger tests.

My theory as to why this is happening is due to the way I have had to code the velocity layout. By adding the template into a string and then using the string as the layout source, I have added a few extra lines of code that the program will have to run through for every instance, generating a very large backlog.

```
-----  
Test set: org.example.Test4V  
-----
```

```
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.8 s - in org.example.Test4V
```

```
org.example.Test4 - Notepad
```

```
File Edit Format View Help
```

```
-----  
Test set: org.example.Test4  
-----
```

```
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 2.998 s - in org.example.Test4
```

Although Velocity seems to use more memory, looking at the surefire reports, where Test4V is velocity and Test4 is pattern, we can see that for 50,000 tests, it is just under 0.2 seconds faster than pattern.

For the largest size of logs we can get using a long, that would be 3.6e+13 seconds faster using velocity templates.

In conclusion, although velocity templates are a lot faster than pattern, they use a lot more memory. It would be up to the user of the program to decide whether they wanted speed or size.