Benchmarking Memory and Computational Efficiency of Various Languages

COS 284, Regan Koopmans

July 23, 2016

Testing Methods

In order to test the time taken when running a program, I used the ’time’ bash command. This command has a 1 millisecond accuracy. In order to record the averages and control the test sequences, I constructed the following short bash script:

```
#!/bin/bash
for i in {1..100}
  do
    var=$(./getTime.sh 2>&1)
    string=”$var”
    for i in {1..999}
      do
        var=$(./getTime.sh 2>&1)
        string=”$string + $var”
      done
    echo $(python −c ”print ($string)/1000”)
done
```

In this case getTime.sh is a small script that runs and measures a program, dependent on its specific running requirements (such as invoking Java or Lisp). The results were then sorted and stored in a file for later comparison.

Results

Memory

Language	*Space Occupied on Disk*
Assembly	4 KB
C++	12 KB
COBOL	16 KB
Fortran	12 KB
Lisp	4 KB (clisp binary is 9.5 MB)
Java	4 KB (JVM is approx. 150 MB)

Computation Time

Conclusion

It is obvious that, which concurs with expectation. This is however only a limited test, and languages may be faster or slower than one another in certain environments and tasks.