Step 3 – Load test the servers

Java:

1. threadGroupSize = 10, numThreadGroups = 10, delay = 2

Wall Time: 317 seconds

Throughput: 630.9148264984227 requests/second

1. threadGroupSize = 10, numThreadGroups = 20, delay = 2

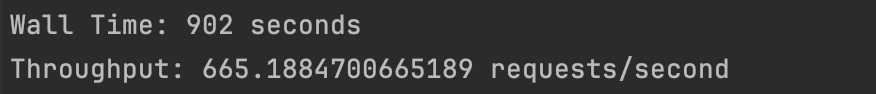
Wall Time: 603 seconds

Throughput: 663.3499170812604 requests/second

1. threadGroupSize = 10, numThreadGroups = 30, delay = 2

Wall Time: 902 seconds

Throughput: 665.1884700665189 requests/second



Go:

1. threadGroupSize = 10, numThreadGroups = 10, delay = 2

Wall Time: 322 seconds  
Throughput: 621.1180124223603requests/second

1. threadGroupSize = 10, numThreadGroups = 20, delay = 2

Wall Time: 622 seconds

Throughput: 643.0868167202573 requests/second

1. threadGroupSize = 10, numThreadGroups = 30, delay = 2

Wall Time: 912 seconds

Throughput: 671.1663607125123 requests/second

### Step 5: Load test the servers again

Java:

1. **threadGroupSize = 10, numThreadGroups = 10, delay = 2**
2. **A screenshot of a computer

   Description automatically generatedthreadGroupSize = 10, numThreadGroups = 20, delay = 2**
3. **A screenshot of a computer

   Description automatically generated threadGroupSize = 10, numThreadGroups = 30, delay = 2**
4. A screenshot of a computer

   Description automatically generated

Go:

1. threadGroupSize = 10, numThreadGroups = 10, delay = 2
2. A screenshot of a computer

   Description automatically generated threadGroupSize = 10, numThreadGroups = 20, delay = 2

A screenshot of a computer

Description automatically generated

1. threadGroupSize = 10, numThreadGroups = 30, delay = 2

A screenshot of a computer

Description automatically generated

### Step 6: Plot Performance

### Java Server:

### A graph showing a graph Description automatically generated with medium confidence

### Go Server:

### A graph showing a barcode Description automatically generated with medium confidence