ECE454 Assignment 2

Max Burstyn (20206120)
Yubin Kim (20239608)

Introduction

This report will discuss Java RMI and passing by value and by reference across multiple machines. The system consists of a server that provides a remote object that provides a simple service. The simple service can return its result either by value or by reference. This reply object supports a single method that emulates some processing, and returns a specified amount of data.

There are five factors that we have control over that will be referred to throughout this paper.

- k: The proportion that the client machine is slower than the server machine in processing.
- t1: The time it takes for the server to perform the simple service, in ms.
- s1: The amount of data the service returns in its reply, in bytes.
- t2: The time it takes for the reply object to process, in ms.
- s2: The amount of data the reply object returns after processing, in bytes.

The other factors that affect the results would be bandwidth, and network communication overhead costs.

Observations

Relating k, t1, and t2

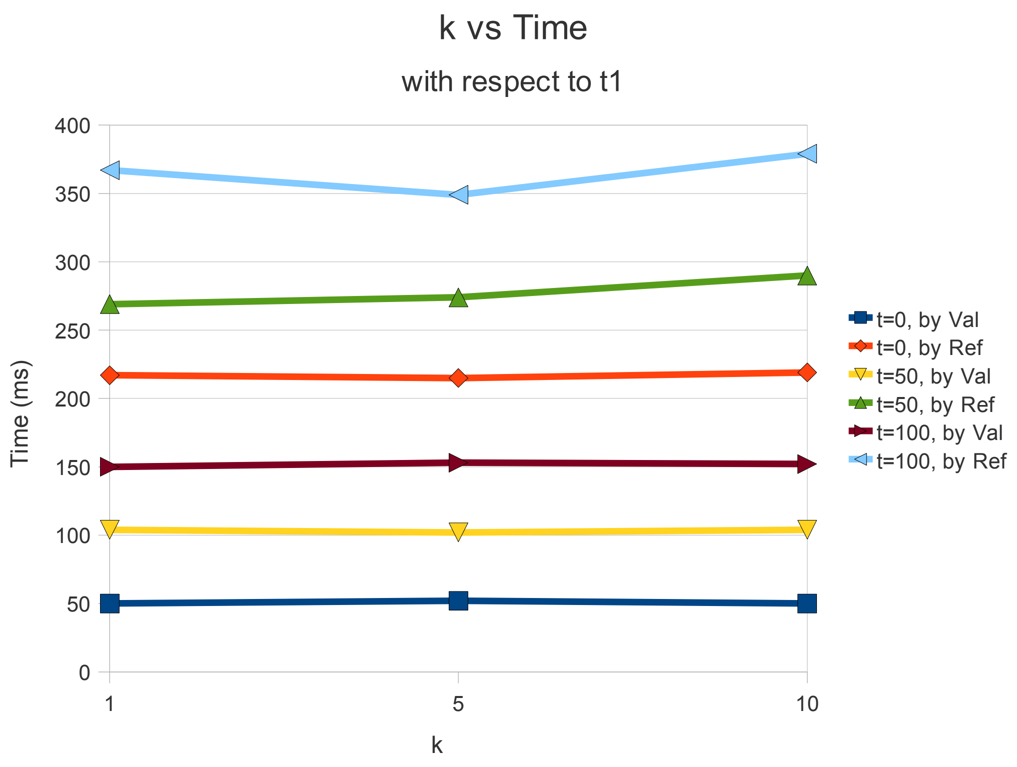

Figure 1

This graph represents the effect of varying k and how it impacts the time for different values of t1. We can see that varying k has **no** effect on the time with respect to different values of t1 and for passing by value and by reference. This can easily be explained because the t1 time is always spent on the server side, which operates independently of k (recall k indicates how much slower the client runs).

We can make one small observation here, and that is that passing by value in general is faster than passing by reference. We can rationalize this be observing that if we were to do pass by value, the first call to the service goes over the network, and then calling process can be handled locally with no problem. However, for pass by reference, this process call need to do all the extra overhead of the remote object lookup for the reply object, and the call has to go over the network, which has quite an impact even if s1 and s2 are small.

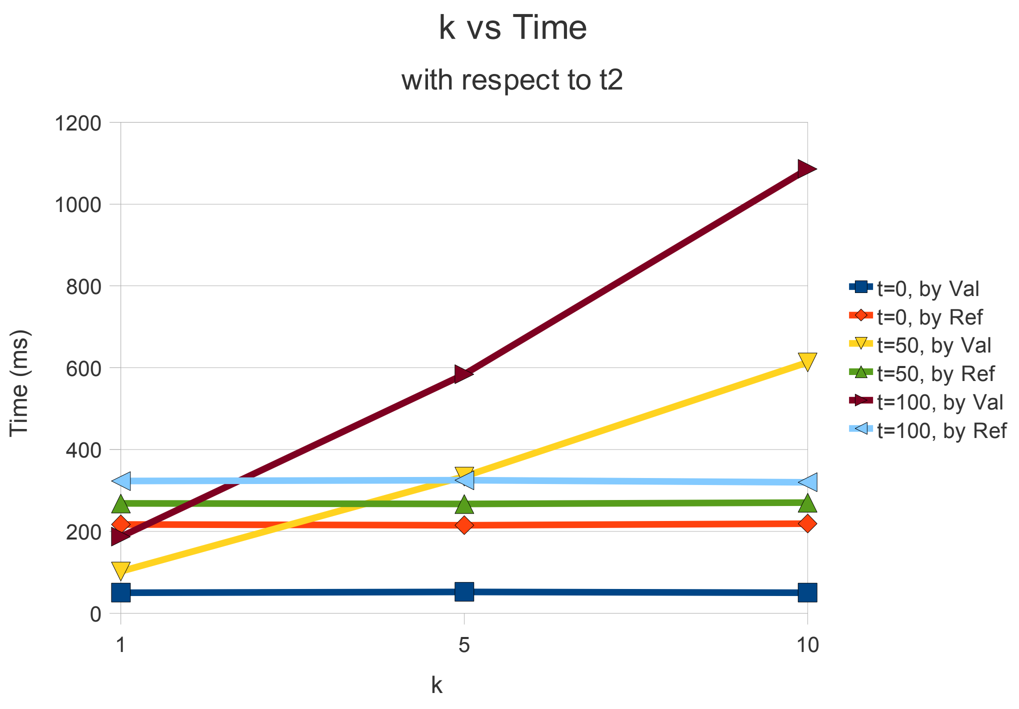

Figure 2

This graph is very similar to the previous one, except that we show the effect k has on t2 instead of t1. Here, we can clearly see that k has a linear impact on the time take for schemes that are pass by value (notice the bottom line, although pass by reference has t2=0, so there should obviously be no impact.) The pass by reference setups however are unaffected when k is varied.

This makes perfect sense since when the simple reply is returned by value, we end up with the object living in the local machine's JVM. Then, when we call process on the reply object, the call is handled locally (in the client) and since the client's speed is affected by the factor k, we see that is has an effect. In the pass by reference case, the process call is delegated back to the object that lives on the server, where the k factor is not used, so there is no noticeable effect in the graph.

We can make some conclusions from this graph. We can look at the intersection of the t=50 data sets, and the intersection of the t=100 data sets. We see that passing by value is faster for t=50 until about k=4. This is the point at which the speed boost from running on the server overtakes the performance boost inherit in pass by value (less network communication needed.) Similarly passing by value is faster for t=100 until about k=2. Obviously, the multiplier has a higher effect if the needed processing time is higher, so for high t2, it is more efficient to do pass by reference. Quantitative analysis will be done below in the conclusion section.

Evaluating s1 and s2

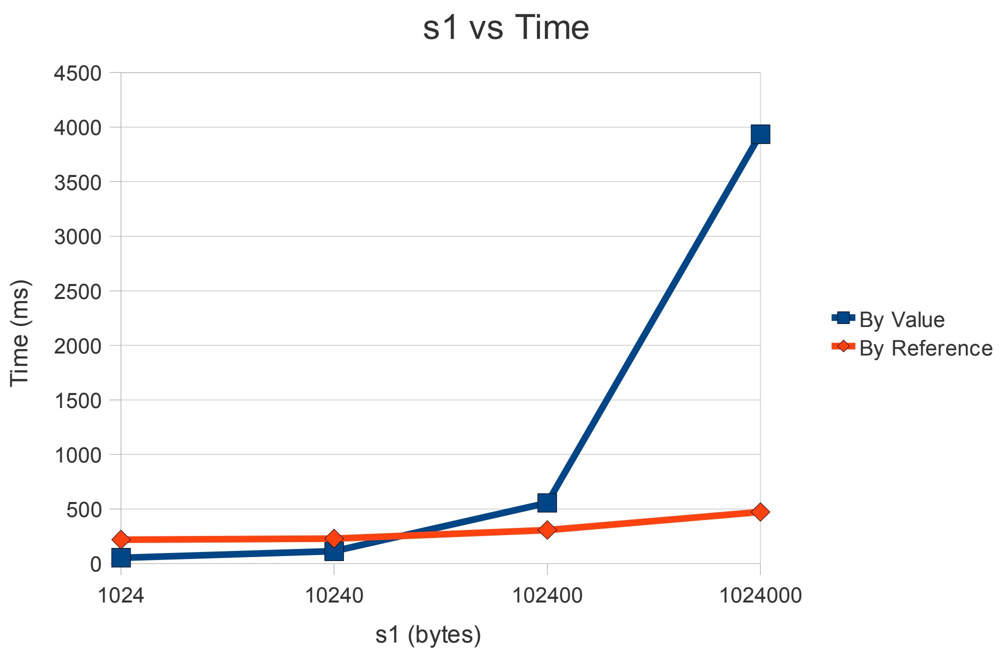

Figure 3

Here we see the effect s1 has on the timing for these setups. We clearly see here that when s1 is large, we see lots of extra time taken for the pass by value system. Pass by reference, however, is largely unaffected by s1. This can be easily explained because s1 is the amount of data returned in the simple service reply. If we use pass by value, that data will have to traverse the network, whereas if we use pass by reference, the data just sits on the server, and only a proxy is returned.

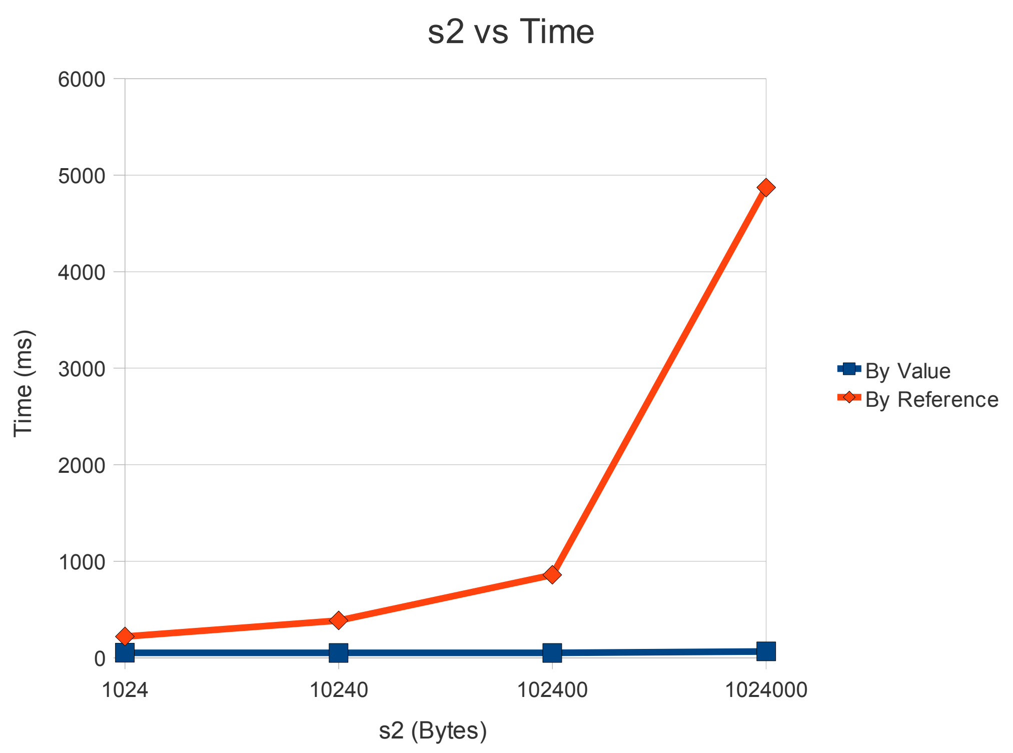

Figure 4

Contrary to s1, we see that when we increase s2, it increases the time it takes for the pass by reference. Additionally, the pass by value system is unaffected by s2. This is also easily explained, since s2 is the data returned from the process call made on the reply object. If we use pass by reference, that reply object lives on the server, so the data returned has to traverse the network. On the other hand, using pass by value, the reply object already lives locally, so returning the data here is very fast and doesn't have to cross the network.

Conclusions

Using the data represented in the four figures above, we will attempt to do some quantitative analysis comparing pass by value and pass by reference for java RMI.

From Figure 1, if we compare data sets with the same t1 values, we can conclude that using pass by reference has an added overhead of approximately 170ms. Additionally, we can see that t1 has the same flat effect on both setups.

From Figure 2, we know that there is a linear relationship with k and t2, but only for pass by value. In pass by reference, k is unused, and only the flat value of t2 is effective.

From Figures 3 and 4, we can determine that the bandwidth for reply objects is approximately 250kb/sec, and for the process call is approximately 200kb/sec by dividing the respective s parameters by total time (using the largest data value to make the overhead as negligible as possible). We'll call these b1 and b2 respectively. Also, recall that only one of b1 or b2 is used depending on whether it is pass by value or pass by reference. Note that these two values are different and may due to the difference in the overhead of serializing the reply object as opposed to a simple character array.

Using these values, we can produce equations for the total time taken for both pass by value and pass by reference systems. For pass by value, we have $T = t1 + k \cdot t2 + s1/b1$. For pass by reference, we have $T = t1 + t2 + s2/b2 + 170$.

Simplifying these equations, we can conclude that we should use pass by value if:

$$(k-1) \cdot t2 + s1/250 < s2/200 + 170$$

and use pass by reference otherwise.