Gaurav Tarlok Kakkar - gkakkar7@gatech.edu
Sai Mohith Potluri - spotluri8@gatech.edu

1 Introduction

The following results were obtained by running on a single node with 28 processors on the PACE-ICE cluster.

We decided against using a randomized pivot because it does not ensure load balancing and in the event that the pivot is a global minima or maxima, it does not even reduce the problem size and one round of communication and computation is wasted.

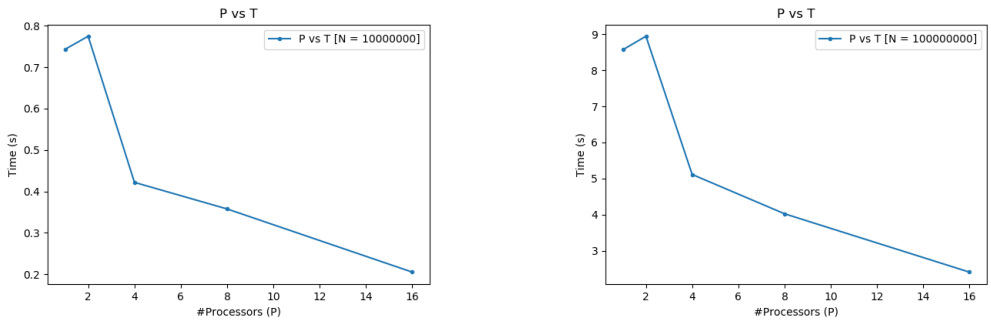
We instead decided to use the median of all local medians as the pivot. The local medians were obtained by each processor by locally sorting their elements. These local medians were then gathered by all processors(using MPI_AllGather) and sorted to obtain the median of all local medians. The advantage of using this as a pivot is that it ensures that each subproblem will be between 0.3 - 0.7 times of the original problem size ensuring load balancing to an extent.

In each round, the time for computation comes mainly from local sorting, which takes $O(\frac{N}{P}\log\frac{N}{P})$ time. The number of elements in each processor remains $O(\frac{N}{P})$ as we ensure load balancing. The communication time in each round comes mainly from the all to all data transfer taking $O(\tau+\mu N)$ time. The algorithm runs for $\log P$ rounds. Another all to all data transfer is done at the end to rearrange the sorted elements such that each processor has exactly the same number of elements as it started with.

$$\text{Computation T(N,P)} = \Theta((\frac{N}{P}\log\frac{N}{P})\log P) = \Theta(\frac{N}{P}\log N)$$

$$\text{Communication T(N,P)} = \Theta(\tau\log P + \mu N\log P)$$

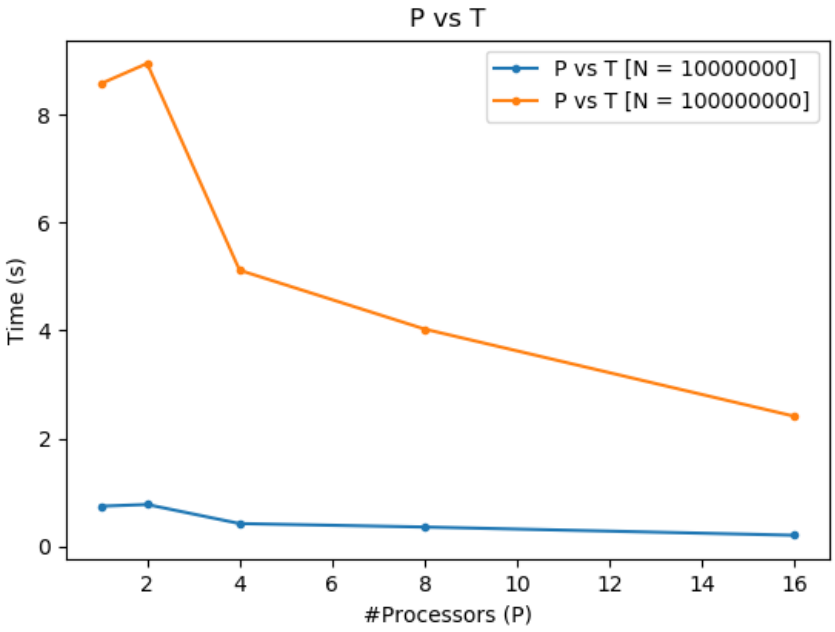
P vs T



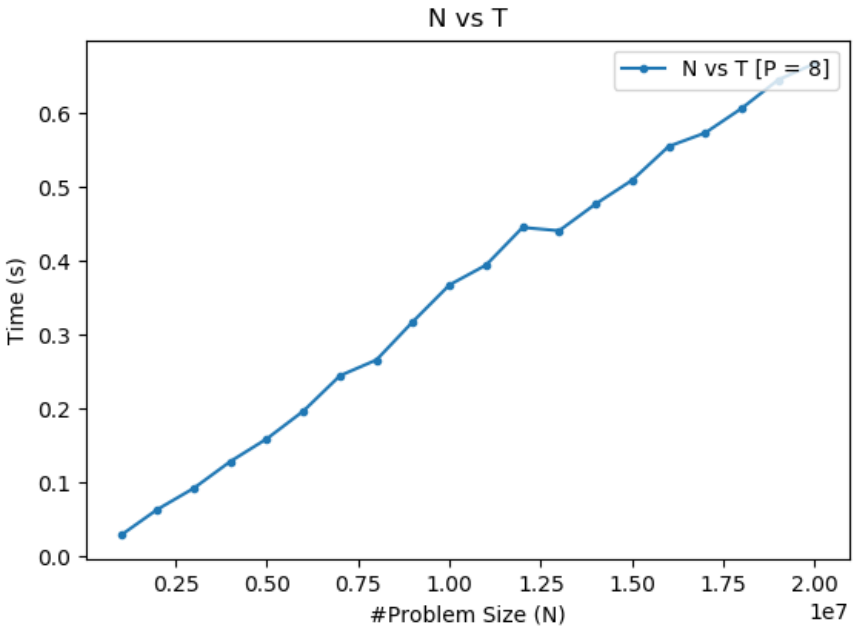
(a) $N = 10^7$ (b) $N = 10^8$

Figure 1: P vs T plots

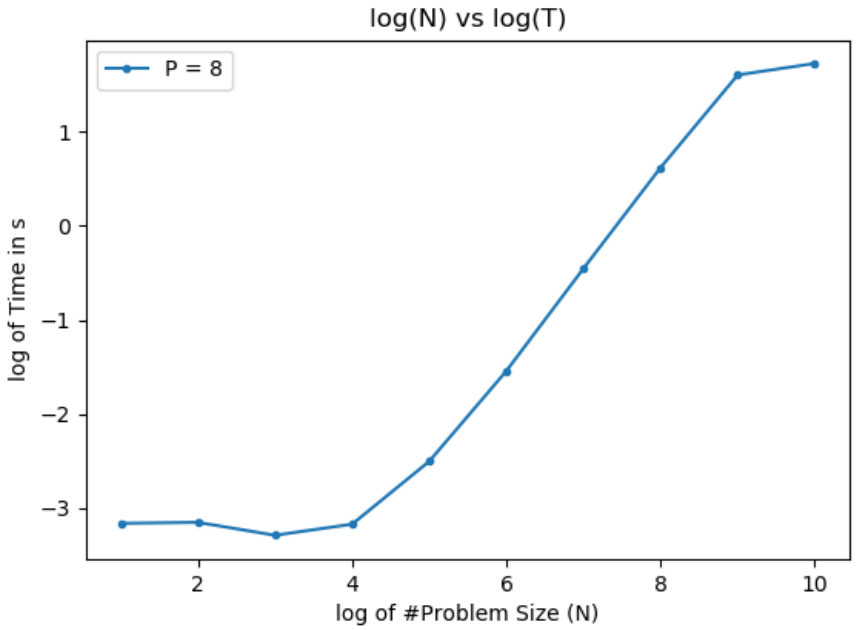
We see from the above plots that the runtime of the algorithm decreases with increase in the number of processors. This is as expected from the theoretical runtime because the term $\frac{N}{P}$ decreases much faster than the increase of $\log P$. The slight increase in time between $T(N,1)$ and $T(N,2)$ can be attributed to the introduction of communication time which was 0 when run sequentially. The parallel overhead in this case was more than the improvement in the computation time.



N vs T



From the theoretical runtime expression, we expect that the runtime increases with N in a similar fashion to $N \log N$. The above plot supports the expectation as the graph looks nearly linear just like $y = x \log x$.



We also plotted a $\log N$ vs $\log T$ graph as shown above. We expect this graph to be linear based on the N vs T graph. But we observe that the trend only gets linear once the problem size reaches 10^4. Before that the communication time is still comparable to the communication time. Once the computation time starts dominating, we get a linear increase of runtime with problem size as good parallel efficiency is maintained. It looks like having at least $10^4/8 = 1250$ elements per processor are needed to achieve good parallel efficiency.