Project 0

Neil P. Parmar

Dept. of Electrical Engineering and Computer Science

Oregon State University, Corvallis, USA

parmarn@oregonstate.edu

1. Introduction

This project discusses about the implementation details for understanding the use of parallel programming with a variable array size and number of tries (NUMTRIES). This report primarily focuses upon the system configurations, the implementation, and performance evaluations & analysis.

1. System Configurations

This project was implemented using the CentOS operating System Release 6.7, at OSU Computer labs. This system has 8 CPU cores, and has an available space of 1.4G. The particular system configuration details and load have been shown in fig 1 and 2.

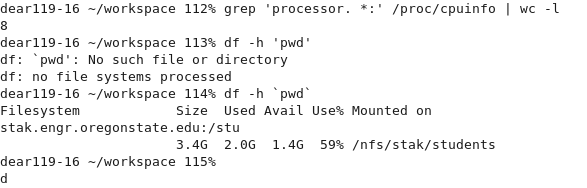


Fig 1. System configurations

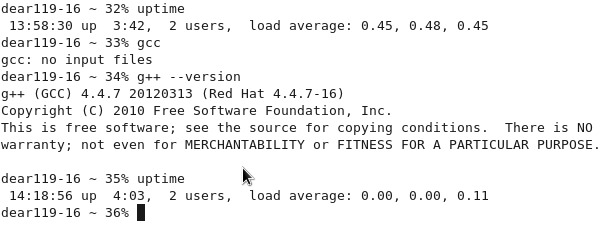


Fig 2. System configurations – System Load

1. Implementation

In this project, “Project0\_t1” evaluates the performance with the use of a single thread and “Project0\_t4” evaluates the performance with the use of four threads. The details of the implementation have been saved in the files Imp1.txt, Imp2.txt, Imp3.txt, and Imp4.txt.

1. Performance Evaluation and Analysis

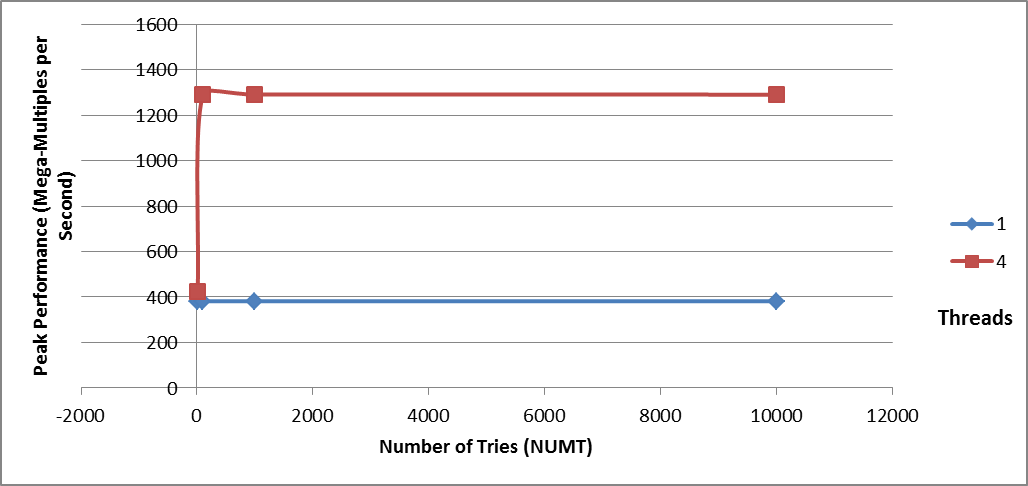
In this project, the performance of the project is evaluated with respect to the Number of Tries (NUMTRIES) and the Array Size measuring Peak and Average Performance in Mega-Multiples per Second. This particular section explains the implementation and analysis of the various tested results.

## Performance evaluation with varying Number of tries

1. Peak Performance Measurement for varying numtries



Considering a single thread with varying number of tries, the peak performance almost remained constant. However, using 4 threads, the peak performance was considerably low with initial NUMTRIES. However, the peak performance sharply increased with an increase in NUMTRIES, as represented in Fig 3.1. However, after reaching its peak, the performance then remained almost constant with an increase in NUMTRIES.

Fig 3.1. PEAK Performance for varying NUMT for threads 1 and 4

Based upon Table I and Fig 3.1, the lower NUMTRIES using a parallel 4 thread environment, would not be utilizing the resources to a considerable extent of its capacity. As the system peak performance remains almost constant for a single thread environment with varying NUMTRIES.

Fig 3.2 below represents the similar results where the number of threads (1 and 2) has been plotted on the horizontal axis with respect to the performance on the vertical axis. Fig 3.2 shows similar graphical representation where the lower NUMTRIES have poor performance for a thread size of 4, however the performance remains almost constant for a higher varying NUMTRIES having a thread size of 1 and 4.

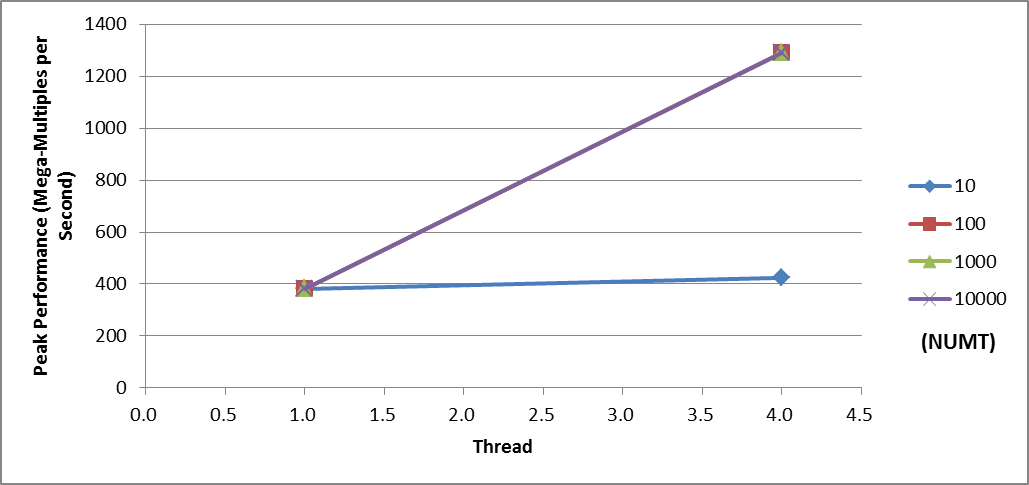


Fig 3.2. PEAK Performance for varying NUMT for threads 1 and 4

Secondly, the Table II represents the Average performance for the similar varying Number of Tries (NUMTRIES).

1. Average Performance Measurement for varying numtries



As being observed from the Table II, Fig 4.1, and Fig 4.2, the average performance shows a similar performance measurement for the varying NUMTRIES. Here, a lower average performance is observed for lower values of NUMTRIES using thread size of 4. However, with a further increase in the NUMTRIES, the average performance after reaching its peak gradually decreases. The particular behavior can be observed using Fig 4.1.

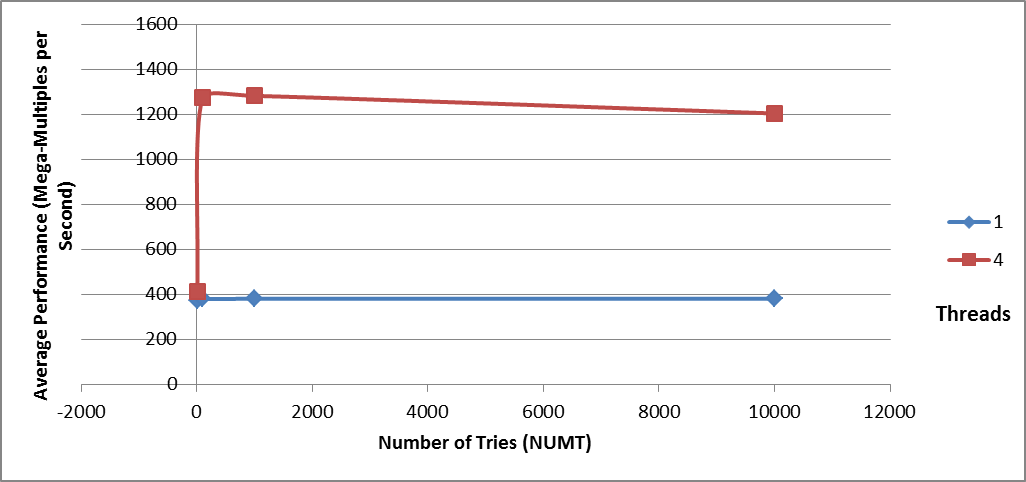


Fig 4.1. AVERAGE Performance for varying NUMT for threads 1 and 4

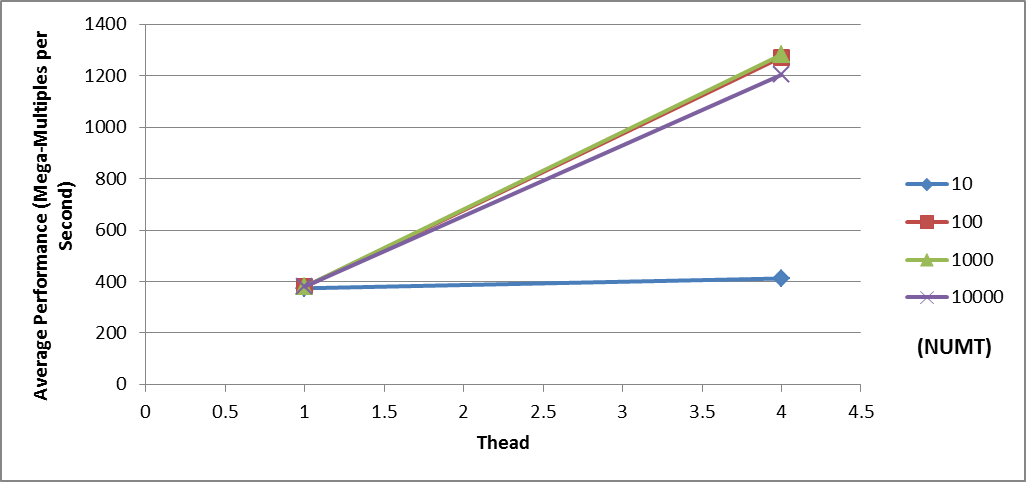


Fig 4.2. AVERAGE Performance for varying NUMT for threads 1 and 4

The gradual decrease in the average performance may not be considerably conclusive as there are certain factors which may affect performance, such as the system load at the moment while evaluating performance.

## Performance evaluation with varying Array Size

1. Peak Performance Measurement for varying array size



Firstly, the highlighted columns in Table III represent the array size which observes a fluctuating performance. The particular behavior can be seen in the implementation file Imp4.txt.

While testing the peak performance, for a single thread a lower performance is observed for a lower array size. However, the performance remains almost constant for higher array size.

However, while using 4 threads, the performance gradually increases with the increase in the number of threads until it reaches a peak. After which, a steady decrease in the performance is observed with an increase in the array size.

The particular result is shown in the fig 5.1 and fig 5.2.

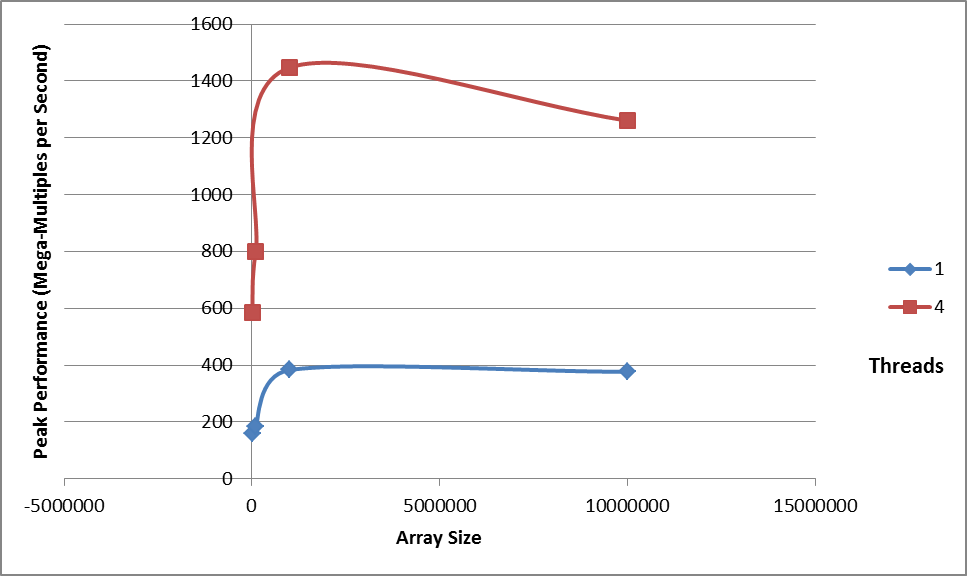


Fig 5.1. PEAK Performance for varying Array Size for threads 1 and 4

Fig 5.2 shows the similar behavior where the performance for thread 1 is almost gradually constant with increase in array size, however the performance degrades with the increase in the array size for 4 threads.

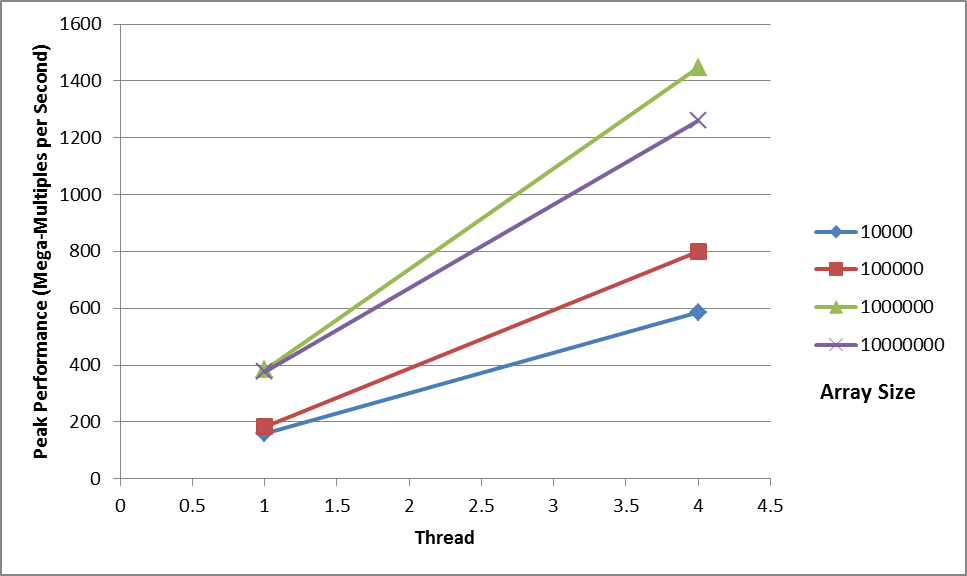


Fig 5.2. PEAK Performance for varying Array Size for threads 1 and 4

Secondly, the table IV below represents the average performance with the similar varying array size. Here, the initial two columns have fluctuating values due to the smaller array size.

1. Average Performance Measurement for varying array size



The average performance, as shown in fig 6.1 and fig 6.2, gradually increases for threads 1 and 4 with the increase in the Array Size. For a single thread, observing fig 6.1, the performance gradually increases with the increase in the array size. However for 4 threads, the average performance gradually increases with the increase in the number of threads. This result is completely opposite to that observed for the Peak Performance in fig 5.1.

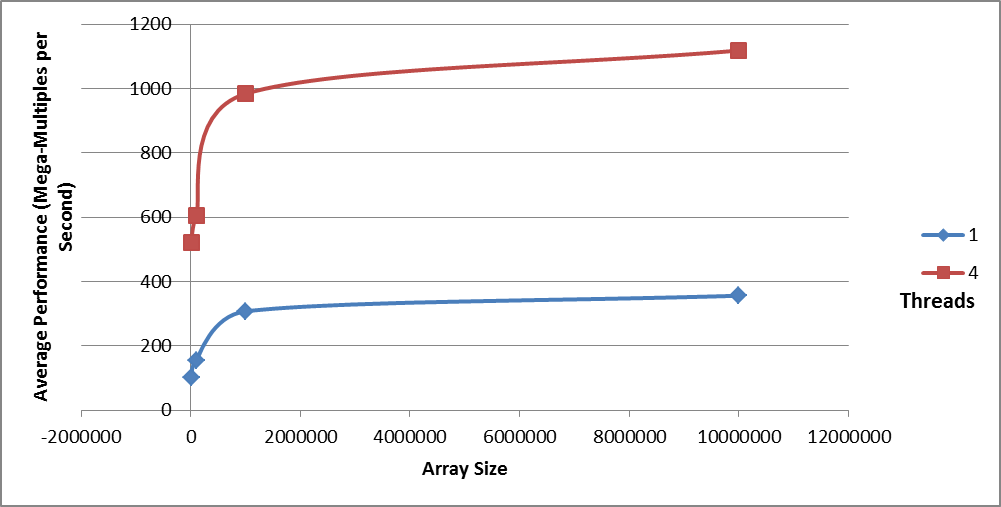


Fig 6.1. AVERAGE Performance for varying Array Size for threads 1 and 4

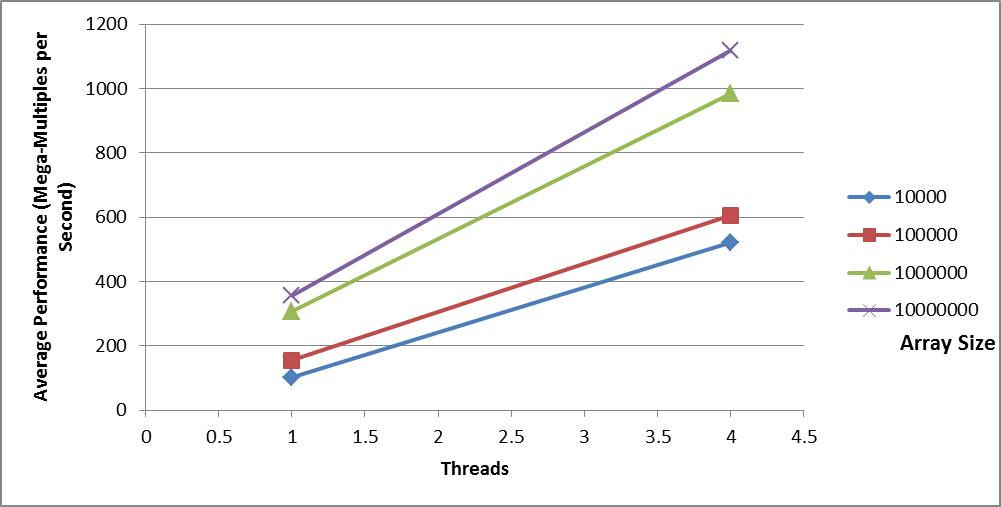


Fig 6.2. AVERAGE Performance for varying Array Size for threads 1 and 4

Fig 6.2 represents a similar increase in performance based upon the threads plotted with respect to the array size.

From the results analyzed using the peak and average performance using varying array size, it can be concluded that the array size should not be considered too large for measurement as the system might not allow much memory usage. Moreover, too lower a value would degrade the performance as the machine would not utilize the assigned capacity.

1. Conclusion

In this particular implementation, an increased system performance was observed with higher NUMTRIES and with standard array size. Moreover, a systems performance may be affected due to various factors such as machine load and its memory capacity.