Spring 2018

Supporting Big Data at the Vehicular Edge

Lloyd Decker
Old Dominion University

SUPPORTING BIG DATA AT THE VEHICULAR EDGE

by

Lloyd Decker
B.S. May 2015, Old Dominion University

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
May 2018

Approved by:

Stephan Olariu (Advisor)

Michele Weigle (Member)

Steven Zeil (Member)

ABSTRACT

SUPPORTING BIG DATA ON THE VEHICLE EDGE

Lloyd Decker
Old Dominion University, 2018
Advisor: Dr. Stephan Olariu

Vehicular networks are commonplace, and many applications have been developed to utilize their sensor and computing resources. This is a great utilization of these resources as long as they are mobile. The question to ask is whether these resources could be put to use when the vehicle is not mobile. If the vehicle is parked, the resources are simply dormant and waiting for use. If the vehicle has a connection to a larger computing infrastructure, then it can put its resources towards that infrastructure. With enough vehicles interconnected, there exists a computing environment that could handle many cloud-based application services. If these vehicles were electric, then they could in return receive electrical charging services.

This Thesis will develop a simple vehicle datacenter solution based upon Smart Vehicles in a parking lot. While previous work has developed similar models based upon the idea of migration of jobs due to residency of the vehicles, this model will assume that residency times cannot be predicted and therefore no migration is utilized. In order to offset the migration of jobs, a divide-and-conquer approach is created. This uses a MapReduce process to divide the job into numerous sub-jobs and process the subtask in parallel. Finally, a checkpoint will be used between the Map and Reduce phase to avoid loss of intermediate data. This will serve as a means to test the practicality of the model and create a baseline for comparison with future research.

This thesis is dedicated to the proposition that learning is a lifetime journey.

ACKNOWLEDGMENTS

There are many people who have contributed to the successful completion of this thesis. This journey would never have been possible without the support of my family. Having a full-time career and working on a Master’s Thesis occupied most of my time. This time was time stolen from my family. Where many could have been resentful of this, my wife, daughter, and son were always supportive and encouraging. This Thesis is just as much theirs as it is mine.

There is one person that stands out as the most influential in my studies. Dr. Olariu has been an inspiration to me. He has been the greatest motivation in my studies. During weekly meetings, I have often felt that I have not met his expectations for that week. Every time he has shifted my negative feelings into positive motivation to continue to strive forward. Without his encouragement, there is a strong chance that I would have put my studies on hold and eventually forsaken them. He will always have my highest admiration and appreciation.

I must also thank the committee members. Dr. Zeil and Dr. Weigle have been kind enough to give their time to help me finish this Thesis. While my lifelong studies have allowed me to acquire multiple undergraduate and graduate degrees, this is the first formal Thesis that I have submitted. The process can be daunting, but their willingness to help has been greatly appreciated. Their understanding and guidance of a novice such as myself to the process has made the submission of this Thesis possible. Their feedback and suggestions have allowed me to take my ideas and share them. My thanks go out to both for giving me their invaluable time and understanding.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION AND MOTIVATION

Internet of Things (IoT) is a broad term encompassing any device that is connected to the Internet. The first devices that come to mind are generally that of smart phones and tablet computers. With the advances in microprocessors, many other devices have been developed such as smart watches, smart glasses, smart meters, connected vehicles, etc. The number of these smart device users is expected to exceed 4 billion by 2019, and Cisco predicts the number of connected IoT devices will reach 50 billion by 2020 [1] [2] [3] [4].

These devices form the periphery of the Internet and are referred to as edge devices. By the 2019 the amount of data generated each month at the edge of the Internet by edge devices will surpass 24.3 exabytes by 2019 [5]. The data generated at the edge is valuable. Due to the gap between available bandwidth and volume of data, much of this data will need to be processed at the edge or it will be lost. Furthermore, the transient nature of this data will require it to be processed in near real-time. Due to the latency costs in moving data between the edge and a datacenter or cloud, cloud-based real-time processing may not be feasible nor economical. Hence, there is a need to process data at the edge.

It is interesting to note that these edge devices that offer computing and storage resources generally remain underutilized. Indeed, it is estimated that the collective computing and storage capacity of smartphones has exceeded that of worldwide servers at the end of 2017 [6]. These devices have the potential to take on the role of servers. For example, StoreDot [7] and uBeam [8] have demonstrated game-changing battery technologies. Implementations of LTE Direct [9], WiFi Direct [10], and WiGig [11] standards will increase peer-to-peer connectivity among edge devices. Finally, container approaches such as Docker [12] on Android will make application portable by addressing security and heterogeneity concerns for edge devices. This is similar to what virtualization have done for servers.

An edge device of interest is the smart vehicle. A smart vehicle is a vehicle that not only has enough processing power and storage to handle the basics of running the vehicle, but it has additional capacity to handle services for the operator of the vehicle. These could be situational awareness, entertainment, and communications. Many studies are underway to use the computing resources and sensor resources of the smart vehicles to create dynamic sensor networks. While the smart phone is currently the driving force in IoT, smart vehicles are quickly coming to prominence.

To any computer engineer or computer scientist, one of the greatest lost opportunities is to have a processor sitting idle. A great deal of effort has been devoted to optimizing the flow of instructions through a processor to minimize wasted clock cycles. As with processors, any computing resource that is left idle is a waste of that resource. Vehicular networks have the potential to become commonplace and many applications have been developed to utilize their sensor and computing resources. This is a great utilization of these resources as long as they are mobile. When the vehicles are parked, these resources are sitting idle. A question to ask is whether these resources could be utilized when the vehicles are not mobile. If the vehicles have connections to a larger computing infrastructure, then they can put their resources towards that infrastructure. With enough vehicles interconnected, there exists a datacenter computing environment that could handle many cloud-based application services. One such service is that of Big Data processing. This thesis will investigate the use of parked vehicles to form a datacenter infrastructure for supporting Big Data processing. This discussion will provide a baseline for this computing infrastructure, and the basis for further investigation on the use of these untapped resources.

The discussion of supporting Big Data at the vehicular edge will focus on a simple case of parked vehicles. A model will be created to evaluate processing Big Data using a datacenter comprised of these parked vehicles. The model will simulate a datacenter implemented on the vehicles in the parking lot of a business that operates twenty-four hours a day, seven days a week. The employees of the business work on staggered eight-hour shifts. This provides a pool of vehicles that can serve as the basis for a datacenter for the business. The vehicles in the parking lot are provided a standard power outlet for charging their vehicles in return for the use of their computing resources.

The challenge facing the implementation of the vehicle datacenter is to determine if it is practical. This is a simplistic model for a specific scenario. It is the desire to expand this model to other broader scenarios. If the simple case is not practical, further research may need curtailed. Furthermore, this model will serve as a baseline for comparison with future research. While this model deals solely with Smart Vehicles, the model could be expanded to deal with heterogeneous devices such as Smart Phone, tablets, and other IoT edge devices.

CHAPTER 2

BACKGROUND OF THE RESEARCH PROBLEM

2.1 THE INTERNET OF THINGS

The expansion of broadband service and the ease at which to connect devices to broadband has created a surge in the number of devices connected to the Internet. No longer are computers the sole devices connecting to the Internet. Smart phones, coffee makers, washing machines, headphones, lamps, wearable devices, and almost anything else that you can think are being connected. The analyst firm Gartner says that by 2020 there will be over 26 billion connected devices [13]. Some think that this number is low. Cisco predicted that the number of connected IoT devices will reach 50 billion by 2020 [3] [4]. Every day it seems that more and more devices are connecting to the Internet.

What is the purpose of all these connected devices? Does IoT constitute an end goal or a means to an end? Being connected allows for a greater efficient use of our time. With greater connectivity comes the ability to efficiently use every waking moment of our day. No longer do you need to make a grocery list. Your refrigerator will keep track and order the groceries for you. No longer do you have down time while driving to work. Your smart vehicle will drive you to work allowing you to start your work day on the road. This seems to be something that makes are lives easier and less stressful. IoT is a tool that will provide the means to an end goal.

On a broader scale, the IoT can be applied to things like transportation networks: "smart cities" which can help us reduce waste and improve efficiency for things such as energy use [13]. At the heart of transportation networks is the smart vehicle. The sensor resources and computing resources of smart vehicles will combine to enable the smart cities of the future. The aggregate computing power of these vehicles will be tremendous and allow for the processing of the enormous volume of data from a variety of sensors.

While not addressed in this discussion, an important issue with IoT is security. There are many stories of smart houses that have been hacked. Hackers yelling at children through baby monitors, constantly changing the temperature on the thermostat, or unlocking the front door are all examples of the concerns in security [14]. Houses are not the only targets. Vehicles have become popular targets the more their onboard processors control more and more of the vehicles system. Hackers have demonstrated the ability to completely take over a vehicle from the driver. This included environmental, entertainment, steering, and engine control [15]. As the IoT grows, so must our vigilance in protecting the multitude of connected devices.

2.2 BIG DATA PROCESSING

Our modern lives involve the collection of large quantities of data. The volume of this data is fueled by the IoT. If one were to doubt this, simply look at social media. It is not uncommon for a single person to create numerous high definition photographs and video on a daily basis. Smartphones have enabled this and they are becoming an integral part of our lives. Smartphones are not the only means of collecting data. IoT includes numerous forms of data collection such as appliances, watches, smart vehicles, and sensor networks [16]. The volume of data being collected and subsequently analyzed is growing exponentially [17]. There are currently estimated over 15 billion devices, and it is estimated that this will increase to 30 billion devices by 2020 and increase to 75 billion devices by 2025. [18] Furthermore, it is estimated that by 2019, monthly data generated by devices such as smartphones, wearable devices of all sorts, and vehicles will surpass

24.3 exabytes [5]. With these devices, comes the opportunity to process the large data sets that are created.

An example of large data sets that need to be processed are those associated with e-commerce applications. In this context, the user's experience is of the utmost importance. Managing searches and shopping carts created by a prospective customer require the ability to efficiently store and recover a customer's information in the form of preferences, purchase histories, and returns. Any delays in presenting requested information to the customer could result in an unhappy customer who likely will not return [19].

Another example is associated with customer searches involving composite services. For example, a customer may need directions to a location along with hotels or restaurants near that location. The searching algorithms need to traverse all available paths and determine the most efficient route based upon current conditions. This requires near real time processing of current sensor data for traffic conditions and processed data for hotels and restaurants. Furthermore, the sensor data and processed data may be located in various locations and must be processed and delivered to the customer in a timely fashion irrespective of how many servers may be down at any moment [20].

Big Data processing involves the processing of terabytes or petabytes of data. The size of the data involved may require new methods from the traditional method of processing data where one application on one computer processes one set of data. With this method, the processing time of Big Data becomes so vast that the results are no longer worthwhile when the processing is done. Data processing at "near real" time is required. Latency is the biggest hurdle to the processing of Big Data. Hardware upgrades in the devices performing the processing are simply not capable of keeping pace with the exponential increase in the volume of data. Different strategies for the processing of data are required.

The idea is not to necessarily change upon what the processing is done. The method of processing the data is the key. As it turns out, emerging Big Data applications involve sophisticated multi-phase data processing [21]. Google's MapReduce [22] [23] and Apache's Hadoop [24] [25] [26] are options that enable the processing of Big Data. The processing performed by MapReduce has two sequential stages, Map and Reduce. In the Map phase, a user-defined function is applied to every logical input record to produce an intermediate result of key-value pairs. The Reduce stage collects all the key-value pairs produced by the Map stage and collapses them using yet another user-supplied function [23]. This method utilizes the idea of distributed computing. By using multiple nodes to process both the map and reduce phases, a large increase of performance can be expected [27].

2.3 CLOUD COMPUTINNG AND THE DATACENTER

Cloud computing has become a driving force in computing and application deployment. Cloud computing is a method of consolidating computing resources into large facilities [28]. This allows the cost to be minimized in infrastructure costs. It also allows the ease of administration. Virtualization of computing resources allows for an abstraction from physical servers. This in turn allows for efficient use of resources. It also allows maintenance and reliability. Virtualized computing resources are simply migrated to physical servers that need maintenance or repair.

Cloud computing allows businesses to provide computing resources to customers. This could range from resources for a single application or resources for large scale database and search engines. Simply put, cloud computing is the delivery of computing services such as servers, storage, databases, networking, software, and analytics [28]. Customers only pay for what they need. This allows for lower costs for the customers since they do not incur overhead costs of physical resources, facilities, and personnel [28]. In general, there are three types of cloud services offered to customers. These are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [29].

IaaS offers its customers full computing resources such as computing and storage. The customer specifies how many processors, how much random access memory (RAM), and how much storage is needed. The cloud computing company provides virtual servers to the customer. An example is Amazon Web Services (AWS). Amazon provides its customers computing resources through the Elastic Compute Cloud (EC2) and storage both Simple Storage Services (S3) and Elastic Book Store (EBS) [30].

PaaS offers its customers a platform on which to develop applications. This frees the customer from maintaining infrastructure needed to develop their applications. A good example of this type of service would be web hosting. Customers design and implement their web sites with nothing more than a web browser. Google AppEngine [31] and Microsoft Azure [32] and examples of this type of cloud service.

SaaS is a “pay-as-you-go” application subscription service. Customers can simply purchase software that they require. This is a benefit to customers that cannot afford the cost of expensive software and the resources required. Google AppEngine [31] is an example of this type of service.

At the heart of the cloud computing is the datacenter [33]. The datacenter is a collection of hardware components such as servers, routers, network switches, and disk libraries [29]. These datacenters can range in size from a single room to that of a large warehouse.

CHAPTER 3

RELATED WORK

A review of previous work with vehicular clouds is in order. The first papers to introduce the idea of vehicular clouds were Eltoweissy *et al.* [34] and Olariu *et al.* [35]. These papers introduced a cluster of vehicles as a means for creating a cloud computing environment. They presented various possibilities and configurations of vehicle clouds. Research has also offered the viability of vehicle clouds with current technology [35] [36].

Arif *et al.* [37] investigated datacenters created from the vehicles parked in a parking lot of a major airport. They presented a stochastic model for predicting the occupancy of the parking lot based upon given time-varying arrival and departure times. They derived a probability distribution for the occupancy of the parking lot as a function of time. They confirmed their model with empirical results.

Vignesh *et al.* [38] investigated services that could be provided by a vehicular cloud. They detailed a master-provider model in which certain vehicles act as controllers (master) and others act as workers (provider). In a Computation as a Service (CaaS) role, the master receives requests for computation from user clients. The master then determines the best available vehicle participant to handle the computational request. In a Storage as a Service (SaaS) role, the master receives storage requests from user clients. The master then determines the optimal vehicle participant to handle the storage. All user client requests and associated data flow through the vehicle masters.

Hussain *et al.* [39] proposed a network consisting of a both a vehicular network (VANET) and a conventional cloud computing environment. Road side gateway terminals

(GT) provided connectivity between the vehicular network and the ground-based cloud computing environment.

He *et al.* [40] proposed services that could be provided by new IoT-based vehicular data clouds. These services include predicting road safety, reducing road congestion, and recommending vehicle maintenance. One useful service that any frustrated driver attempting to find a parking spot would appreciate is that of a service that would direct the driver to the most appropriate parking spot for their needs. They stressed that IoT-based vehicular data clouds need to be efficient, scalable, secure, and reliable. They concluded that existing algorithms and mechanisms are unsatisfactory to meet all these needs simultaneously.

Florin *et al.* [41] investigated a vehicular cloud based on vehicles in parking lot of a medium sized business. They determined that current wireless technology could not efficiently support Big Data applications on a vehicular cloud. They investigated migration techniques to increase the reliability of Big Data processing on vehicular clouds. Their model was based on a medium-sized business with a parking lot containing 2560 parking spaces that are continuous occupied by Smart Vehicles. This model is the basis for the model used in this Thesis.

CHAPTER 4

PROBLEM DEFINITION

A datacenter utilizing a vehicular cloud would be similar to any existing datacenter that supports cloud computing. The major difference is that the physical servers for the cloud architecture are no longer located within server racks in a large building. The physical servers themselves are distributed within a large parking lot. The vehicles are the servers. This discussion does not deal with small parking lots with few vehicles resident. The topic of this discussion deals with large parking lots with many vehicles that are resident for a long period of time. This is the case for airports and medium to large businesses that operate 24 hours a day and 365 days a year. The latter will be the focus of discussions.

The model will simulate a datacenter implemented on the vehicles in the parking lot of a business that operates 24 hours a day and 365 days a year. The employees of the business work on staggered eight-hour shifts. This provides a pool of vehicles that can serve as the basis for a datacenter for the business. The vehicles in the parking lot are provided a standard power outlet for charging their vehicles in return for the use of their computing resources. Wired connections to local access points are provided for all vehicles. The challenge facing the implementation of the datacenter is to maintain high availability and reliability.

The business is a medium-size establishment that employs 7,680 people and operates around the clock, seven days a week. Each employee drives their own vehicle to work. To avoid bottlenecks in the parking lot, the business implements staggered eight-hour shifts. At the top of each hour 320 employees end their workday and leave the plant, only to be replaced by 320 fresh employees that start their eight-hour workday. The parking lot has a capacity to park 2,560 vehicles. The 320 vehicles belonging to departing employees leave the parking lot before the 320 new vehicles pull in. There are no reserved slots and an employee picks a random slot when arriving. In this manner, the parking lot remains full during the entire day excluding the change of vehicles at the top of each hour. For the sake of simplicity, there is no time between the departing and arriving vehicles.

The Vehicle Datacenter offers its users a virtualized instance of their desired hardware platform and operating system bundled as a Virtual Machine (VM). This virtual machine with associated operating system is hosted by a vehicle in the parking lot. The vehicles are assumed to have been preloaded with a suitable Virtual Machine Monitor (VMM) that maps between the virtual machine and the vehicle's resources. Each vehicle can host multiple virtual machines and has ample disk space to accommodate virtual machines and any data being processed. The size of the virtual machines is uniformly 1GB.

The customers of the Vehicle Datacenter run Map-Reduce jobs whose durations are uniformly distributed between 2 hours and 24 hours. The duration of a job is taken to be the amount of time it takes the job to execute in the absence of any overhead. Each customer’s job takes an input of 2GB of raw data and generates final data uniformly distributed between 0.5 GB and 2 GB in size. Specifically, the Map-Reduce job generates the same amount of intermediate data (at the end of the Map stage) and final data (at the end of the Reduce stage).

The network that interconnects the vehicles in the parking lot is organized in a tree architecture (see Figure 1). The root of the tree is a switch called the Datacenter Controller

(DC). The DC has four children, termed Region Controllers (RC). Each RC is a switch and has four children, termed Group Controllers (GC). Finally, each GC is a switch and has four children, termed Access Points (AP). Each AP is a switch in connecting a cluster of 40 parking spots (vehicles). The vehicles in a cluster communicate solely through their designated AP. The links between DC and RCs are 40 Gbps. The links between the RCs and GCs and the links between the GCs and the APs are 10 Gbps. Finally, the links between the APs and the vehicles are 1 Gbps.

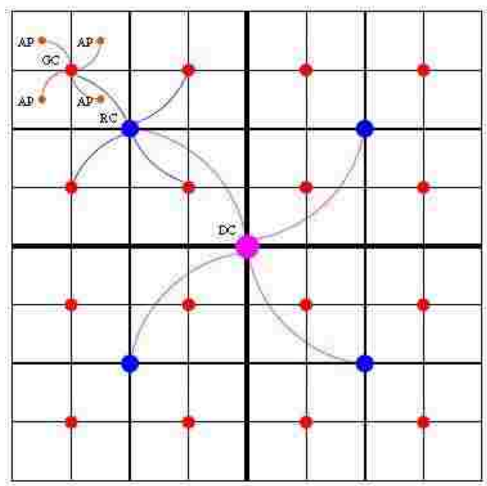


Figure 1: Model Network Depiction

CHAPTER 5

TECHNICAL SOLUTION

The goal of this simulation is to create a vehicle cloud upon the vehicles in a parking lot of a medium sized business. The parking lot is assumed to be constantly full. When a vehicle leaves, there is another to take its place. The emphasis on the simulation is the effect of random residency times, not that of capacity. The simulation consists of a datacenter controller, a resource manager, a job manager, log manager, a network, and vehicles in the parking lot. The simulation is written in C++, and the binary code for this simulation is available upon request.

5.1 DATACENTER CONTROLLER

The datacenter controller is assumed to be ground-based. This means that it is not a vehicle but is a resource that is provided to the vehicular cloud. This model is similar to the model presented by Hussain et al. [39]. It is comprised of a resource manager, job manager, and log manager (see Figure 2).

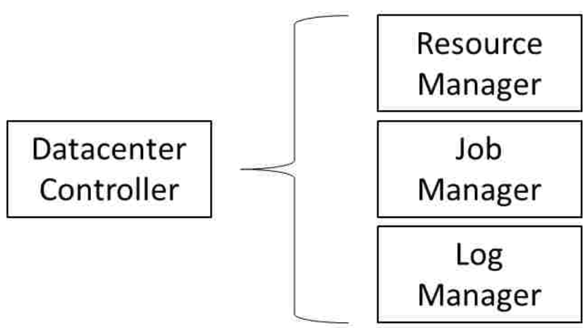


Figure 2: Components of the Datacenter Controller

5.1.1 RESOURCE MANAGER

The resource manager handles the acceptance of user's jobs for processing. It handles the injection of jobs into the system via the job manager. The resource manager keeps track of the number of current jobs being processed. It compares the number of current jobs being processed to the maximum number of simultaneous jobs allowed. If this maximum has not been reached, new jobs are sent to the job manager until the maximum number of simultaneous jobs is reached.

The resource manager is responsible for polling the parking spaces to determine the occupancy of a space. It is further responsible for polling the vehicles to determine if the vehicle is available for task assignment. In other words, it maintains information on the status of the parking spaces and vehicles. The resource manager is responsible selecting available vehicles for job assignment. These assignments can be for job processing or for backups for intermediate data backups. Backups are used to provide a checkpoint during the processing of jobs. Furthermore, it handles all downloads and uploads of data. This could be virtual machine images, raw data, or final processed data.

5.1.2 JOB MANAGER

The job manager controls each job that is submitted by the user for processing. This entails many tasks. The job manager divides the job into sub-jobs for processing. This is the core idea for this simulation, divide the job into smaller pieces and perform parallel processing. The job manager requests resources from the resource manager to perform the job processing. It requests the allocation following vehicles for the user's job: one vehicle for each sub-job processing and two vehicles for each sub-job to act as backups for intermediate data during the checkpoint process. The job manager identifies the virtual machine that is required for the user's job. It directs the resource manager to download the virtual machine to all allocated vehicles. It further directs the resource manager to download the respective sub-job raw data to the allocated vehicles for processing.

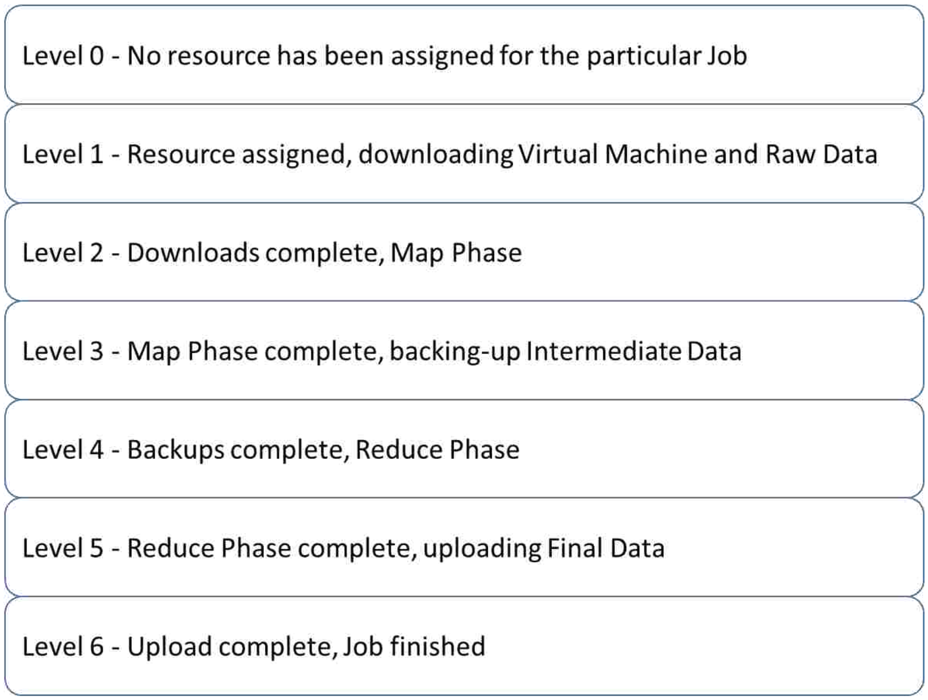


Figure 3: Levels of Job Completion

The Job object keeps track of the progress of the overall job and all sub-jobs by the means of seven designated levels (see Figure 3). Before the overall job progresses from one level to the next, all the sub-jobs need to have completed the current level. Level 0 is the assignment of vehicles to handle the processing of the job. Level 1 is the downloading of the virtual machine and raw data to allocated vehicles. Level 2 is the map phase of the data processing. Level 3 is the collection and backing up of intermediate data to vehicles allocated to handle the backups. Each vehicle that is assigned raw data to process will be assigned two vehicles as backups for the intermediate data that is produced. At level 3 is where the checkpoint is achieved. This allows for level 4 to be a return point in case a sub-job is later interrupted by a leaving vehicle. This Level 4 is the reduce phase. Level 5 is the uploading of the final processed data to the datacenter. Level 6 designates the job as being complete.

5.1.3 LOG MANAGER

The log manager is responsible for logging the statistics of the user's jobs for the entire simulation. Once a simulation has reached its prescribed number of time intervals, all statistics for evaluation are logged by the Datacenter Controller. These are recorded to a file for the specifics of the completed simulation. A separate running file is used to record the statistics for all the simulations being run.

5.2 NETWORK

The parking lot consists of a set number of parking spaces. Each of these parking spaces keeps track of the occupancy of a vehicle. The vehicle maintains information on whether it is currently running a job. It also keeps track of when it arrives and leaves the parking lot. It is assumed that the vehicles in the parking lot are resident for eight consecutive hours. While this knowledge would facilitate the migration of working jobs in a preemptive manner, the intent of this thesis is to investigate the viability of the vehicle datacenter to perform with no knowledge of residency and therefore not perform any preemptive migrations. The vehicles will form a network node on the network. This will be done via the network interface associated with each parking space.

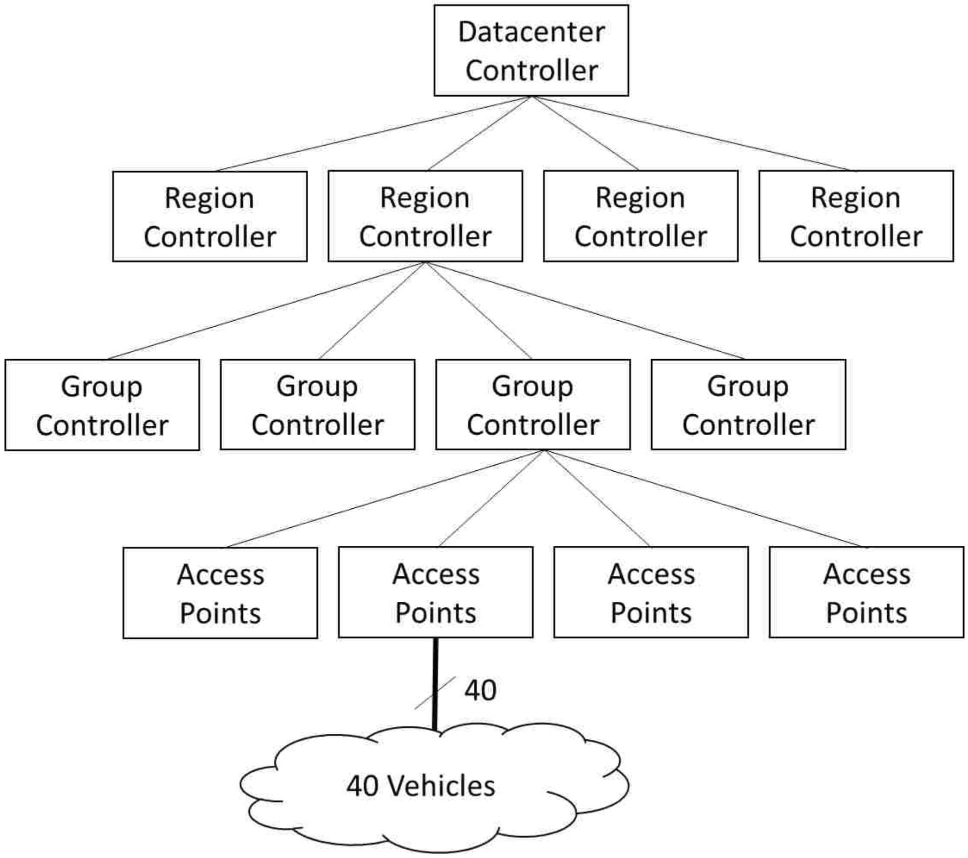


Figure 4: Network Tree Hierarchy

The network consists of a tree structure with the vehicles in the parking spaces being the leaf nodes (see Figure 4). The root of the network tree is a network switch that comprises the core layer of the network. The datacenter attaches directly to the core and forms the datacenter controller. There exist two levels of network switches comprising the distribution layer of the network. These are the region controller and the group controller. There are four region controllers directly connected under the datacenter switch. Under each region controller there are four group controllers. The access layer of the network is comprised of the access controllers. They are either wireless access points in the wireless model or network switches in the wired model. There are four access controllers connected to each group controller. Each access controller can support 40 vehicles.

The network switches are those that may be found in a current high-performance network. All connections are wired. The throughput of the connections between the Datacenter Controllers and the Region Controllers are 40 Gbps. The throughput of the connections between the Region Controllers, Groups Controllers, and Access Points are 10 Gbps. The last mile connections between the Access Points and the vehicles is 1 Gbps.

The simulation of the complexity of a packet network is accomplished by using average throughput over a time interval. Since greater time intervals create a larger error in throughput simulation, smaller time intervals are utilized. In the case of this simulation, one second time intervals are used. The simulation of network traffic is a two part process. The first counts the number of connections across each link between nodes. The second calculates the bandwidth for an entire communication path between two nodes.

The first part involves calculating all traffic paths for all communications that will occur in the next time interval. Every link is marked with the number of communication paths that will traverse it. If the link is a full duplex link, as in the case of most wired links, just one communication path is added to the link for communications between two nodes. This is done since transmitting and receiving can be accomplished simultaneously on a full duplex link. If the link is half duplex, as is the case for most wireless links, two communication paths are added to the link for communications between two nodes. For this simulation, multicast traffic is not simulated. All traffic is unicast traffic. Furthermore, the 80% threshold of half duplex connections is ignored. This means that 100% of a links bandwidth is assumed to be used.

The second part involves calculating the bandwidth for each link that will be available in the next time increment. This is accomplished by dividing the link's bandwidth by the number of connections utilizing that link in the next time increment. This implies there is no priority of service and every communication is allocated equal bandwidth on all links. Then each communication path is evaluated to determine the bandwidth for the entire path. This is done by finding the link with the lowest bandwidth along the communication path for each communication path. This negates any possibility of buffer overruns on network devices and the associated retransmits that occur due to the buffer overrun.

All communication is assumed to be Internet Protocol (IP). There will be three kinds of communications. The first communication is the downloading of guest operating system and raw data to the vehicles during level 1. The next is the backing up of intermediate data from one vehicle to another vehicle at level 3. The final communication is the uploading of final data from the vehicles to the datacenter at level 5.

5.3 VEHICLES

The vehicles are assumed to have a virtual machine manager pre-installed prior to parking in the parking lot. This will allow them to host a virtual machine with the user preferred operating system that will be used as a node in the vehicle cloud. This node will be used in the processing of a user's Big Data job. In essence, these components can be seen as stacking upon one another. As Figure 5 shows, the Virtual Machine Manager is installed on the Vehicle Hardware. The Virtual Machine with the user's operating system is installed on the Virtual Machine Manager. The Virtual Machine is then able to handle user jobs. The jobs are assigned by the Datacenter Controller.

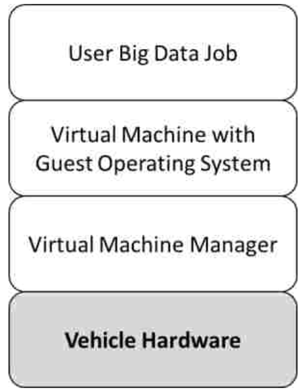


Figure 5: Virtual Machine Hierarchy

CHAPTER 6

EVALUATION OF DEVELOPED SOLUTION

In accordance to Design of Experiment (DOE) techniques, the variables for this simulation are grouped into three categories: constants, factors, and response variables. Constants are static variables that are not changed between simulations. Factors are those variables that are considered to be the independent variables that are changed in order to test the performance of the system. Response variables are the dependent variables that are recorded to investigate the performance of the system. The purpose of this simulation is to determine if a vehicular datacenter is a viable mechanism for the processing of Big Data. The important aspect of this model is that no migrations of jobs are allowed. This model relies on dividing jobs into smaller sub-jobs to compensate for not performing migration. This simple model serves to baseline a model for further study. It is sufficient to find a configuration that proves the viability of processing Big Data at the vehicular edge. Only if this is the case would further study be practical. Furthermore, future innovations to the model can then be compared with the baseline to form a tradeoff analysis between cost and performance of the innovation.

6.1 SIMULATION FACTORS

Many variables will affect the viability of the simulation results. To simplify the model, as many variables as possible are made static. Static and varied variables are listed in Table 6.

Simulation Factors	Method	Values
Size of Parking Lot	Static	2560 vehicles
Residency Time of Vehicles	Static	8 hours
Network Configuration	Static	Tree
Network Throughput	Static	40Gbps-10Gbps-1Gbps
Percentage of Vehicles Tasked	Static	100%
Number of Simultaneous Jobs	Varied	100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 jobs
Number of Worker Objects	Varied	3, 5, 7, 9, 11 workers
Size of Jobs	Varied	3600, 7200, 10800, 14400, 18000, 21600, 25200, 28800, 32400, 36000, 39600, 43200 seconds

Table 1: Simulation Factors

6.1.1 SIZE OF PARKING LOT

As previously described the simulation will model a medium sized business with a 2560 space parking lot. It was decided to utilize a scenario with a set size parking lot with a guaranteed full occupancy. This is the model that is reflected with the medium sized business. This model is the same as that used by Florin *et al.* [41]. Varying parking lot sizes will be a topic for later study.

6.1.2 RESIDENCY TIME OF VEHICLES

Many different models could be used for the residency of vehicles in the parking lot. Stochastic models have been developed to model the residency time of vehicles in airport parking lots [37]. These same models could be used to predict arena or shopping mall parking lot residency. While migration is not considered to be an option for this particular model, static residency times are used to create a simplified model for a baseline case. In this thesis, we assume that the residency time for each vehicle is eight hours. This being said, the simulation does not allow any prediction to time remaining for each vehicle. When the vehicles leave the parking lot, it is as though they randomly left the parking lot. Truly random residency will be left for further investigation.

6.1.3 NETWORK CONFIGURATION

The network consists of a tree structure with the vehicles in the parking spaces being the leaf nodes (see Figure 4). The root of the network tree is a network switch that comprises the core layer of the network. There exist two levels of network switches comprising the distribution layer of the network. These are the region controller and the group controller. There are four region controllers directly connected under the datacenter switch. Under each region controller there are four group controllers. The access layer of the network is comprised of the access controllers. They are either wireless access points in the wireless model or network switches in the wired model. There are four access controllers connected to each group controller. Each access controller can support 40 vehicles.

6.1.4 NETWORK THROUGHPUT

Current network technologies were used as the basis for the network throughput model. The throughput of the connections between the Datacenter Controllers and the Region Controllers are 40 Gbps. The throughput of the connections between the Region Controllers, Groups Controllers, and Access Points are 10 Gbps. The last mile connections between the Access Points and the vehicles is 1 Gbps. All traffic is unicast IP datagrams. The traffic consists of the guest operating system, raw data, intermediate data, and final data. An evaluation of using multicast traffic is saved for future investigation.

6.1.5 PERCENTAGE OF VEHICLES TASKED

Since migration is not allowed for this model, it is assumed that all available vehicles will be tasked. This enables the full utilization of the parking lot. In other words, 100% of vehicles are available for tasking. If a job requires a vehicle and none are available, then the job must wait until a vehicle becomes available. Sub-jobs are only assigned to one vehicle at a time.

6.1.6 NUMBER OF SIMULTANEOUS JOBS

A simple job injection model is used for this simulation. It simply creates new jobs until a specified number of jobs is reached. This is considered to be the number of simultaneous jobs. These values range from 100 to 1000 in increments of 100 simultaneous jobs. When the simulation is first started, all jobs up to the number of simultaneous jobs are injected at once. As the jobs are completed, new jobs are inserted into the simulation to maintain the number of simultaneous jobs.

6.1.7 NUMBER OF WORKER OBJECTS

A worker is a vehicle that has been assigned to process a sub-job. The simulation will be run with a range of different number of workers. The number of workers corresponds to the number of sub-jobs that each job is divided. With each sub-job being processed in parallel, the job completion time can be reduced. The simulation will be run with 3, 5, 7, 9, and 11 workers.

6.1.8 SIZE OF JOBS

Since the residency for any vehicle is only 8 hours, the dividing of jobs into sub-jobs is essential to completing jobs longer than 8 hours. The simulation will be run having random job sizes ranging from 2 to 24 hours with 2-hour increments. To better understand the impact of job sizes on the efficacy of the system, the simulation will also be run with set job sizes ranging between 2 and 24 hours with 2-hour increments.

6.2 RESPONSE VARIABLES

There are two response variables for this simulation. They are the number of jobs completed during a simulation run and the average time to compete a job. These will be used to evaluate the viability of the vehicle datacenter.

6.3 RESULTS

To understand the impact of factors on the response variables, some factors are constant while others are varied to find a viable model. This becomes the baseline for future research. Towards this end, a step by step refinement process is used. The first step is to determine the optimal number of workers for each job. The next is to evaluate the steady state of job completion times for set job sizes. Then a performance comparison is conducted between job completion times for models with set job sizes compared to that of the job completion times of models with random job sizes. Then a comparison is made between a wireless vehicle network model and a wired vehicle network model. Finally, the efficacy is evaluated with a comparison to a traditional datacenter network. This is a model with infinite residency times.

6.3.1 DETERMINATION OF NUMBER OF WORKERS

A worker is a vehicle that has been assigned to process a sub-job. A series of simulations are run with varying sizes of workers. These workers allow a job to be broken into smaller sub-jobs that can then be run in parallel. A worker can also be a vehicle used during checkpointing that serves as a backup for intermediate data at level 3 of job processing. Each sub-job has two backup workers that are assigned to it for redundancy. With numerous vehicles serving as workers for a single job, there is a tradeoff between job completion and resources. Jobs can be completed in a shorter period at the cost of numerous vehicles. For example, a single job dived into 5 sub-jobs will use 5 workers for processing the sub-jobs and 10 workers as backup workers for a total of 15 workers. For a vehicle datacenter running 100 simultaneous jobs with five sub-jobs per job will require 1500 vehicles.

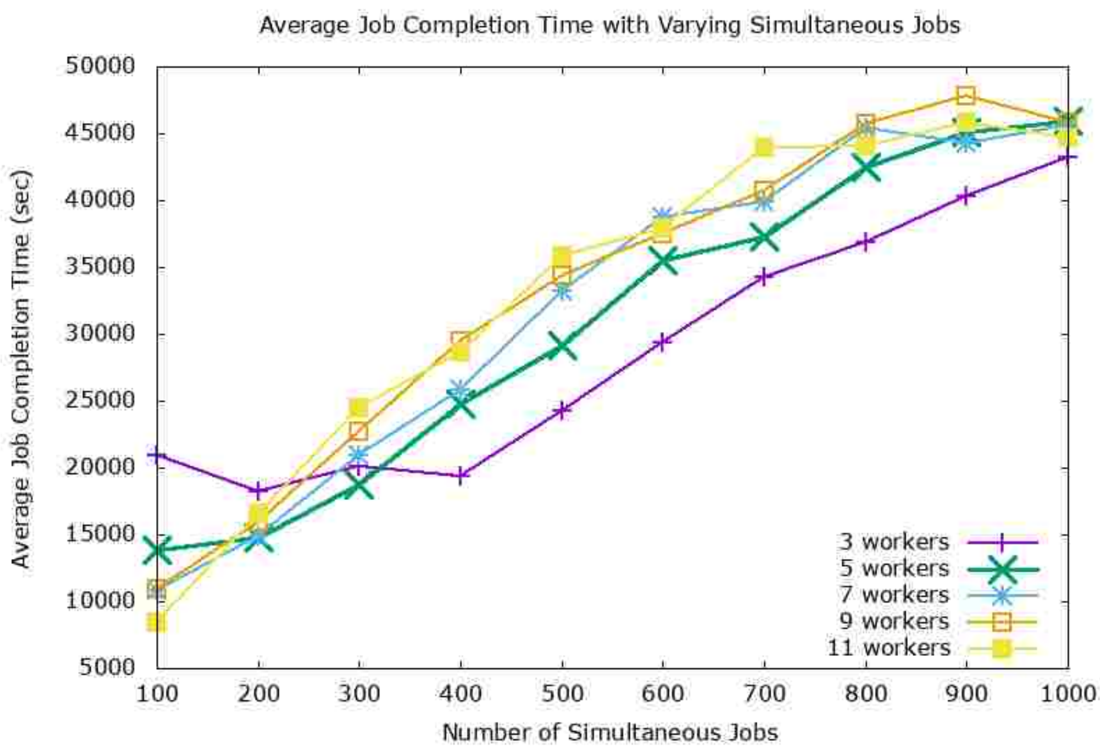


Figure 6: Average Job Completion Time with Varying Simultaneous Jobs

50 Simulations were conducted with worker sizes of 3, 5, 7, 9, and 11. Figure 6 displays the average job completion times for each simulation. Figure 7 displays the number of completed jobs for each simulation. For the average job completion times, the higher number of workers performs better than lower number of workers until there is a contention of resources (around 200 to 330 simultaneous jobs). Once resource contention is reached, the results are completely opposite with lower number of workers performing better than higher number of workers. The average completion times for 3 workers appears to be overall better than the others with 5 workers being next. It is interesting to note the 5 workers perform better than the 3 workers with lower number of simultaneous jobs. For the number of completed jobs, again the higher number of workers performs better than the lower number of workers until resource contention is reached. The number of completed jobs shows that the 5 workers seems to perform better overall. Taking average completion time and number of completed jobs into consideration, 5 workers appears to be a slightly better choice than the others. Furthermore, there seems to be no benefit in choosing higher number of workers as is evident by the overlapping performance of the 7, 9, and 11 workers in both average completion time and number of completed jobs. However, no number of workers seems best. 5 workers appear to perform good for both average job completion time and number of completed jobs. It is for this reason that the 5 workers simulation is chosen for continued testing.

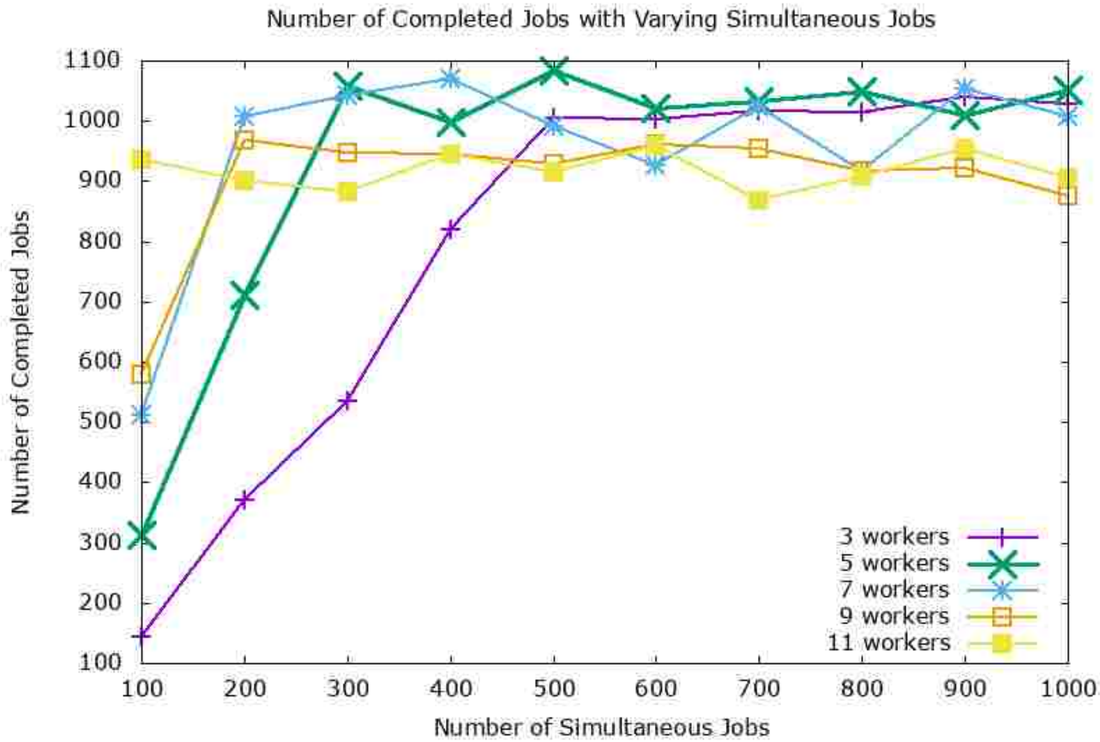


Figure 7: Number of Completed Jobs with Varying Simultaneous Jobs

6.3.2 STEADY STATE OF JOB COMPLETION TIMES

A series of simulations are run for set job sizes and number of simultaneous jobs. The goal of this is to determine if for each combination of job size and number of simultaneous jobs, the completion times will reach a steady state over time. In other words, the system will become stable over time and not have completion times increase without bound. If the completion times do not reach a steady state, then that combination of job size and number of simultaneous jobs would have to be considered not viable. Job sizes are chosen from 2 hours to 24 hours with 2-hour intervals. The number of simultaneous jobs is chosen from 100 to 1000 jobs with 100 job intervals. For the 12 different job sizes and 10 different number of simultaneous jobs, there are 120 combinations tested. Graphs of cumulative average completion time is plotted for each combination. Three steady state patterns are evident in the graphs: bound, trend, and no.

Bound refers to the cumulative average reaching a steady state. Trend refers to the cumulative average beginning to approach a steady state, but not reaching a steady state within the test period. No refers to the cumulative average continuing to increase throughout the test period. The job size is also graphed with the cumulative average completion time to serve as a comparison.

Figure 8 represents a case of a bound cumulative average completion times. As a reference, the red line indicates the job size (processed in a non-parallel manner). As the simulation starts there is a ramp up of completion times. The important thing to note is that the cumulative average completion times settle into a steady state. This is evident in the horizontal line of cumulative average completion times. This case would represent a viable vehicle datacenter model for the given job size and number of simultaneous jobs.

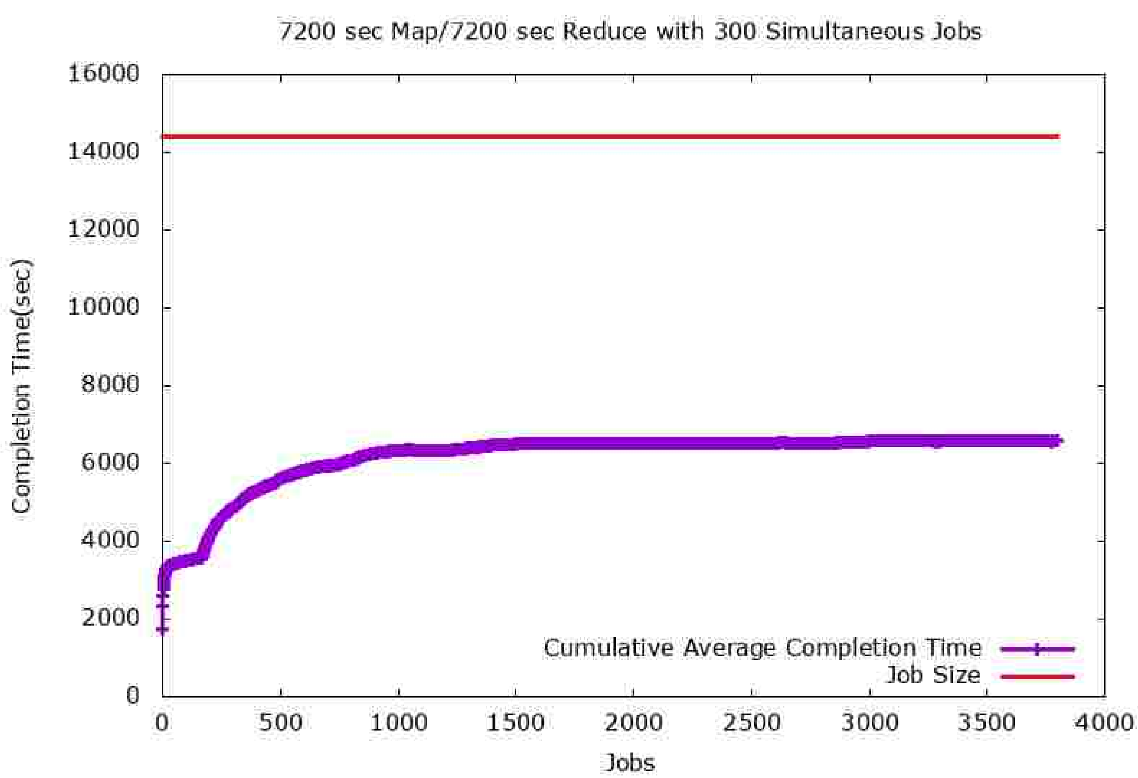


Figure 8: Example of Bounded

Figure 9 represents a case of trend cumulative average completion times. As the simulation starts there is a ramp up of completion times. After the ramp up, the cumulative average completion times begin to curve towards a steady state within the test period. This is evident in the curve approaching a horizontal line of cumulative average completion times. This case implies that a viable vehicle datacenter is possible for the given job size and number of simultaneous jobs.

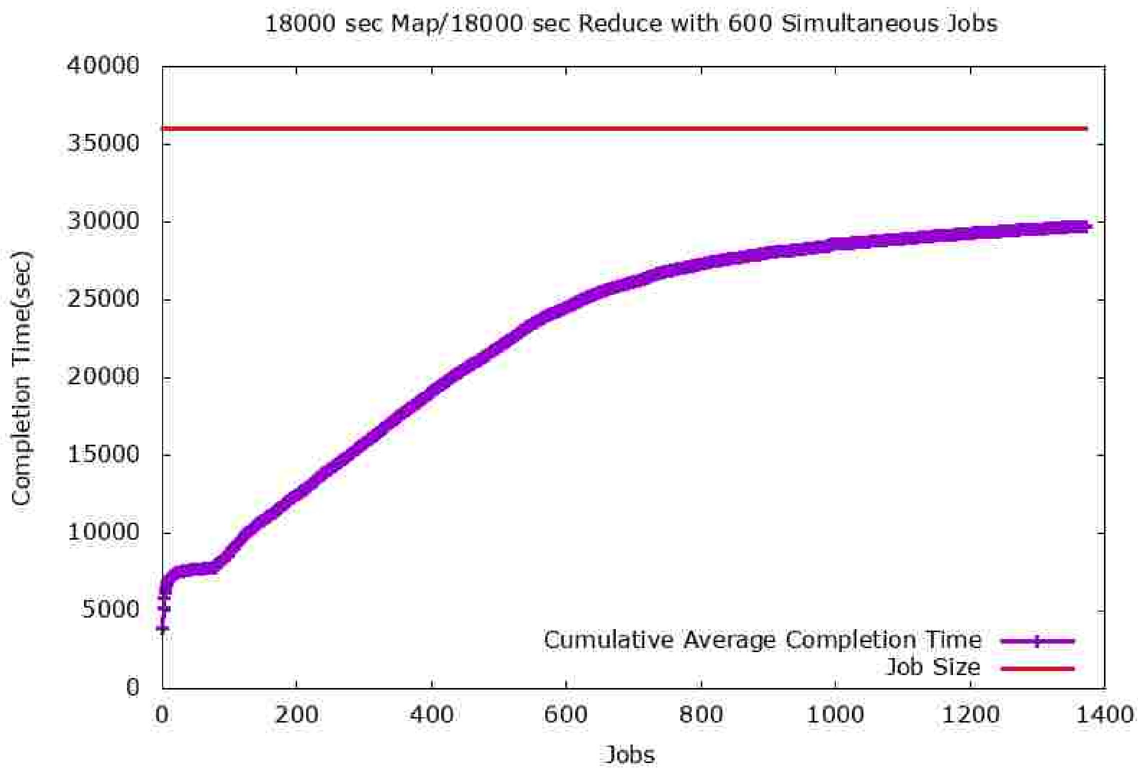


Figure 9: Example of Approaching

Figure 10 represents a case of no steady state in the cumulative average completion times. As the simulation starts there is a ramp up of completion times. The important issue is that the cumulative average completion times form a line that continues to increase throughout the test period. This case represents that a vehicle datacenter is not likely viable for the given job size and number of simultaneous jobs.

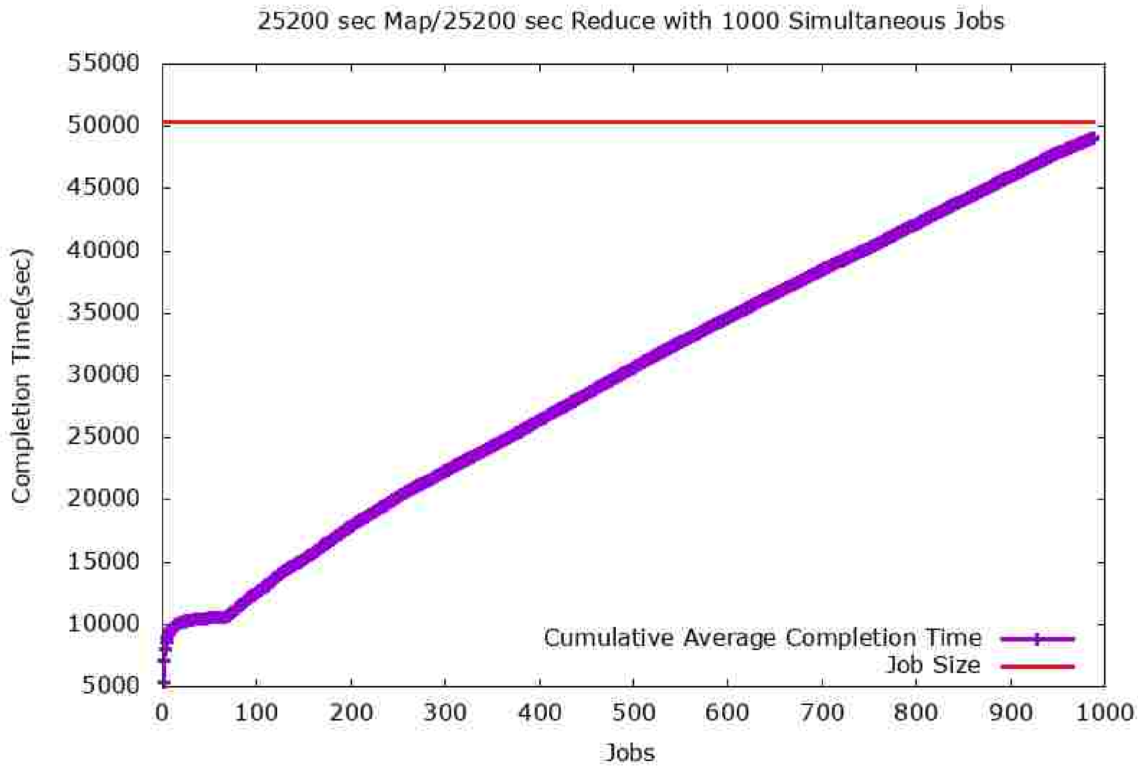


Figure 10: Example of Increasing

The related graphs for all 120 combinations are displayed Appendix A: Job Completion Times. Table 2 summarizes the results of all 120 simulations. The table is color coded to reveal if the cumulative average completion times reached or trended towards a steady state completion time that is less than the job size. Green refers to being less than the job size. Orange refers to a bound or trend case that is not less than the job size. Red refers to a case where the cumulative average completion times continues to increase. The results reveal the vehicle datacenter performs well for simultaneous jobs less than 500. An interesting point is that small job sizes in conjunction with large numbers of simultaneous jobs do not perform as well as medium sized jobs in conjunction with large numbers of simultaneous jobs. Finally, large job sizes in conjunction with large numbers of simultaneous jobs does not appear to be viable solutions. As the number of simultaneous jobs increases, the available resources are overwhelmed. This causes a great deal of contention for resources and results in the model not being viable for large job sizes with large numbers of simultaneous jobs.

Table 2: Summarization of Correlations

Job Size (sec)	Number of Simultaneous Jobs									
	100	200	300	400	500	600	700	800	900	1000
3600	bound	bound	bound	bound	bound	trend	trend	trend	trend	trend
7200	bound	bound	bound	bound	trend	trend	trend	trend	trend	trend
10800	bound	bound	bound	bound	bound	trend	trend	trend	trend	trend
14400	bound	bound	bound	bound	trend	trend	trend	trend	trend	trend
18000	bound	bound	bound	bound	trend	trend	trend	trend	trend	trend
21600	bound	bound	bound	bound	bound	trend	trend	trend	trend	trend
25200	bound	bound	bound	bound	trend	trend	trend	trend	trend	no
28800	bound	bound	bound	bound	bound	trend	trend	trend	no	no
32400	bound	bound	bound	bound	bound	trend	trend	no	no	no
36000	bound	bound	bound	bound	trend	trend	no	no	no	no
39600	bound	bound	bound	bound	trend	no	no	no	no	no
43200	bound	bound	bound	bound	trend	no	no	no	no	no

6.3.3 PERFORMANCE BETWEEN RANDOM AND SET JOB SIZES

Now that a baseline of simulations has identified the behavior of set job sizes, simulations are run to identify the behavior with random job sizes. The series of simulations that are run for set job sizes are compared to the simulations run for random job sizes. The intention is to see if each job size within the random job sizes will follow the average completion time as for the set job size simulation runs. In other words, determine if vehicle datacenter will be able to process numerous different size jobs simultaneously with the same performance as handling only set sized jobs.

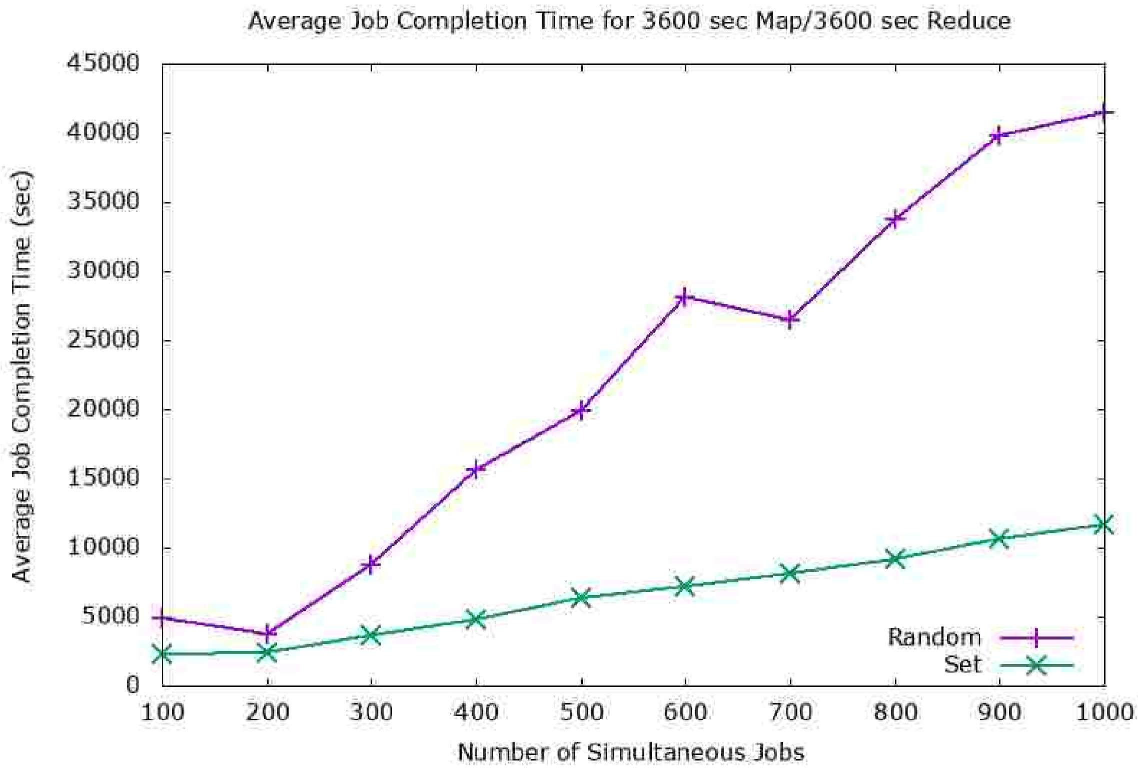


Figure 11: Small Job Size Comparison

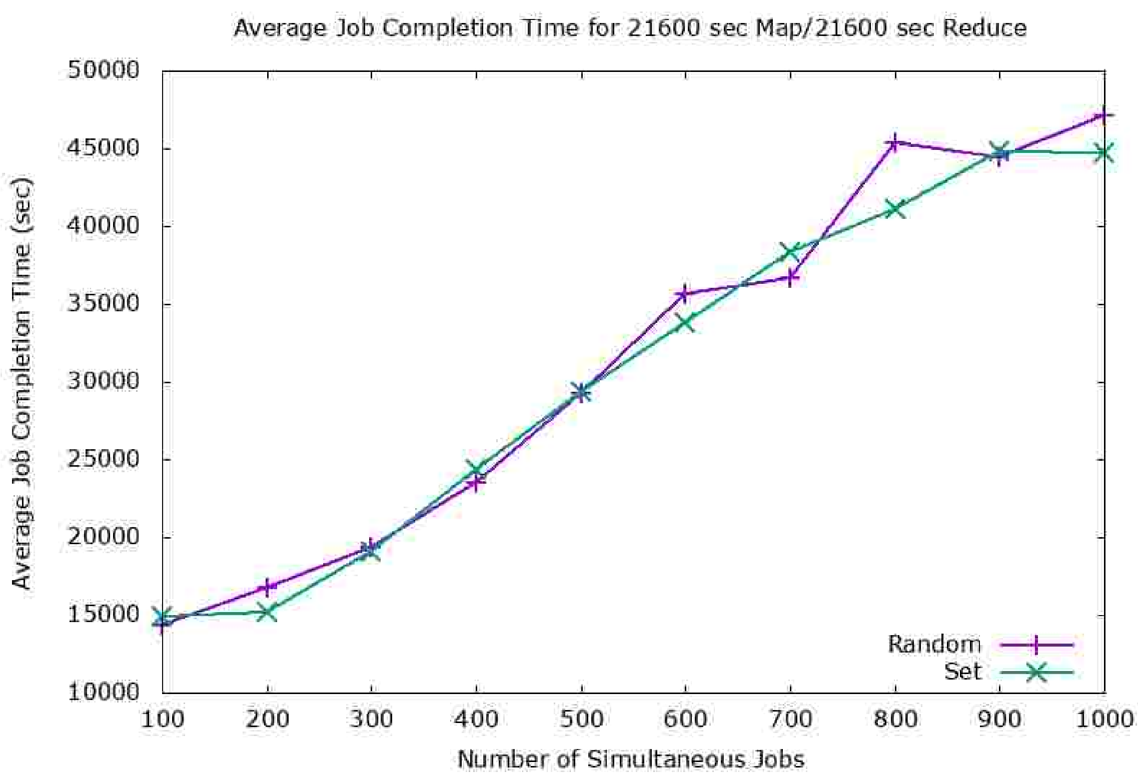


Figure 12: Medium Job Size Comparison

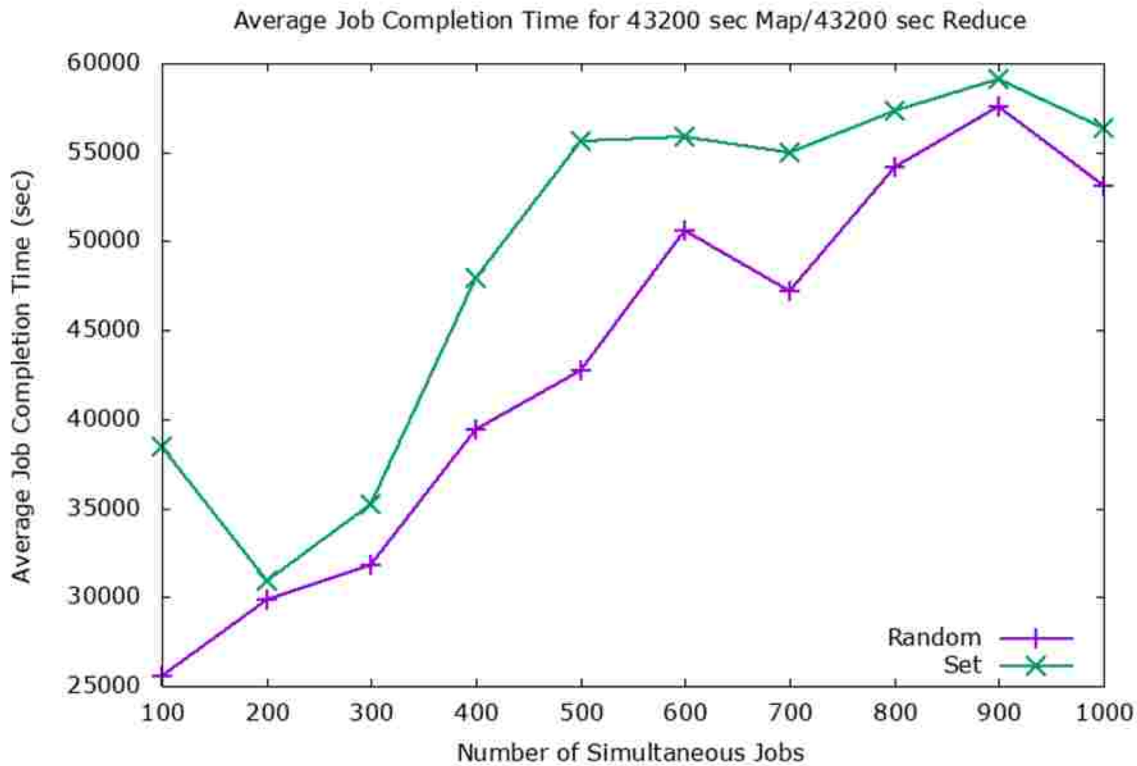


Figure 13: Large Job Size Comparison

Figure 11 demonstrates that for the 3600 sec job sizes, the randomizing of job sizes has a significant effect on job completion time. The average completion times for the jobs in the randomized job size simulations are considerably higher than for the set job size simulations. Figure 12 demonstrates that for the 21600 sec job sizes, the randomizing of job sizes has does not affect the job completion time. An interesting result is found for the 43200 sec job sizes. Figure 13 demonstrates that for the 43200 sec job sizes, the randomizing of job sizes has an effect on the job completion time. The average completion times for the jobs in the randomized job size simulations are lower than for the set job size simulations. This cannot be taken as ground truth since the previous section determined that large job sizes have a weak to no correlation on average job completion times.

All related figures are located in Appendix B: Random and Set Job Sizes. Figures 78-79 demonstrate that randomizing has a significant impact on job completion times for the job sizes between 3600 sec and 14400 sec. Figures 80-82 demonstrate that the job completion times were consistent between the randomized job size simulations and the set job size simulations for job sizes between 18000 sec and 32400 sec. Figures 82-83 demonstrate that job completion times were higher for the set job size simulations than for the randomized job size simulations.

6.3.4 WIRELESS VS WIRED

The vehicle datacenter requires a great deal of network traffic with the downloading of the operating system (1 GB) and raw data (2 GB), copying intermediate data (0.5 GB to 2 GB) to backup workers, and the uploading of final data (0.5 GB to 2 GB). This would lead to the conclusion that the limited and shared bandwidth of a wireless network would not be sufficient to handle the necessary bandwidth of the vehicle datacenter. A simulation is run to verify this. The simulation network is modified so that the vehicles connected to an access point will share 54 Mbps vice the 1 Gbps dedicated link. This is a half-duplex connection. This means that the 54 Mbps is shared among all vehicles connected to the same access point. Furthermore, only one communication can occur at a time so that collisions can occur when more than one communication is attempted. This causes retransmits to occur. This is contrast to that of the wired model that uses a 1 Gbps switched network that utilizes connections that are full-duplex. This means that each vehicle can communicate with the access point at the same time as any other vehicle on that access point.

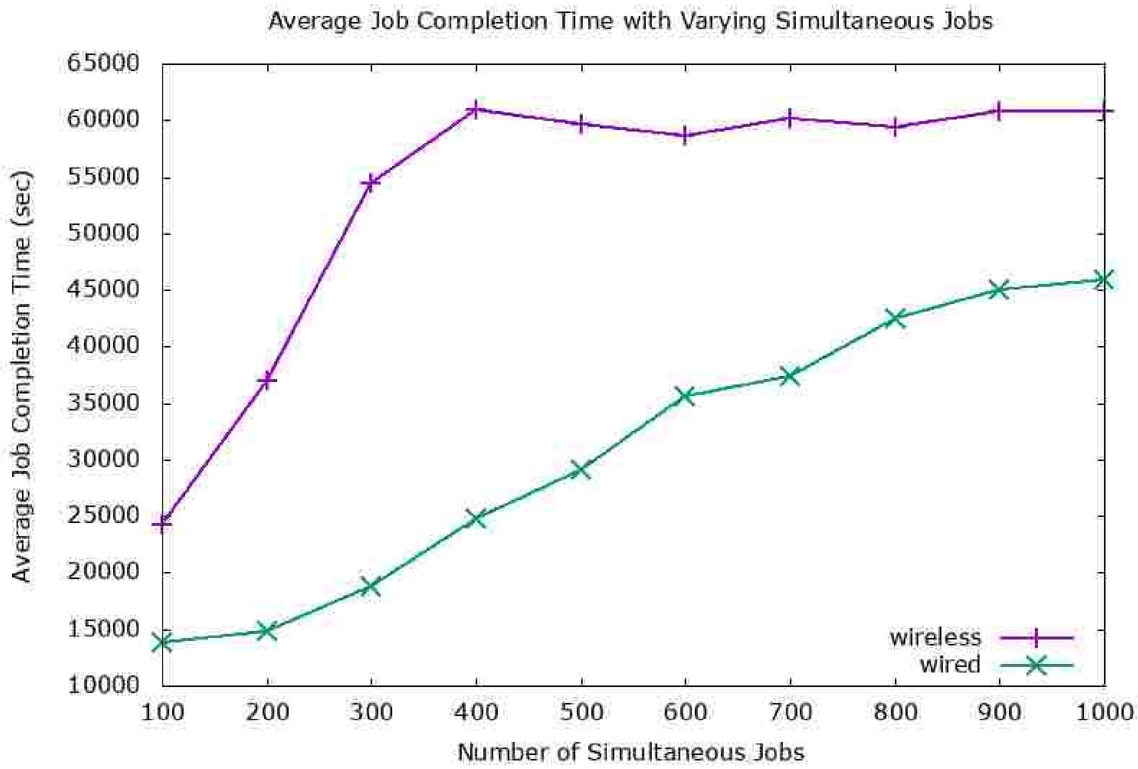


Figure 14: Average Job Completion Time with Varying Simultaneous Jobs

Figures 14 displays the average completion time of jobs for both the wireless and wired simulations. While the average completion times are higher for the wireless than for the wired, there is no indication that the wireless is not functional. Figure 15 displays the number of completed jobs for both the wireless and wired simulations. The number of completed jobs for the wireless is considerably lower than the wired. Again, there is no indication that the wireless is not functional. If one takes into consideration that the wired bandwidth is nearly 20 times that of the wireless for the vehicles, the results show that the wireless model performs better than one might expect.

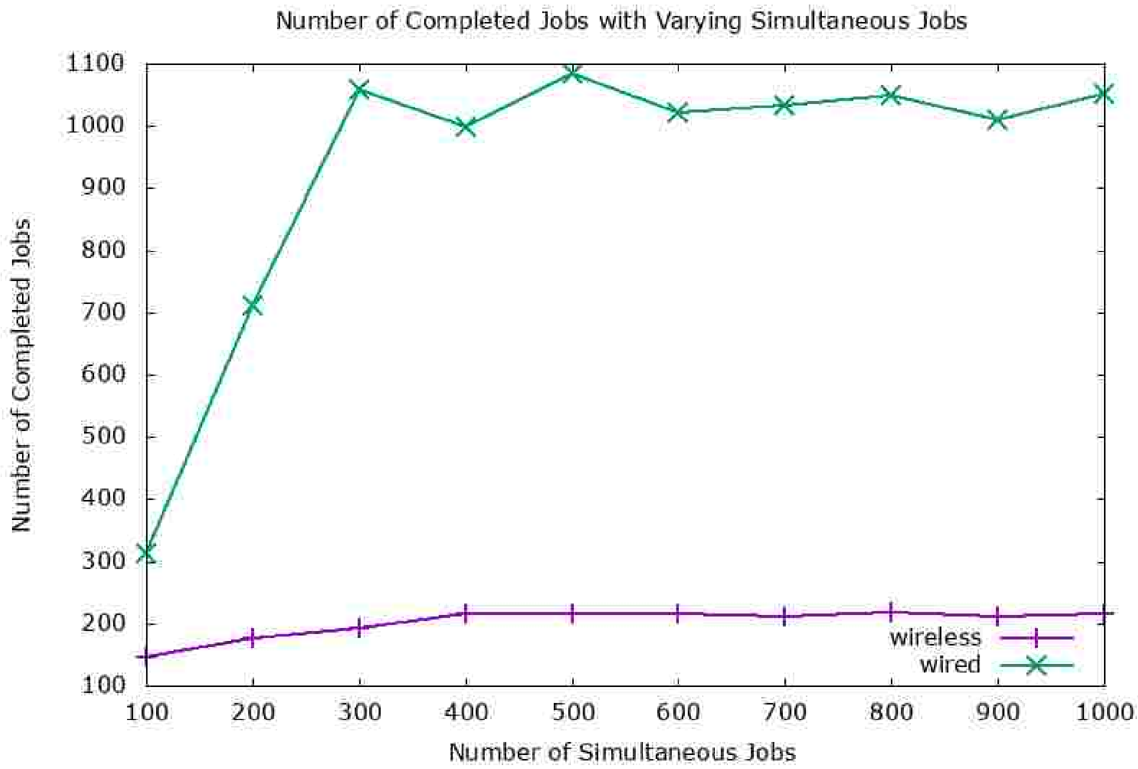


Figure 15: Number of Completed Jobs with Varying Simultaneous Jobs

6.3.5 VEHICLE DATACENTER VS TRADITIONAL DATACENTER

The final step is to compare the wired model with that of a traditional datacenter. These 2 simulations are identical except that the traditional model has an infinite residency time for its processors (vehicles). This means that once a worker starts a job it will not be interrupted.

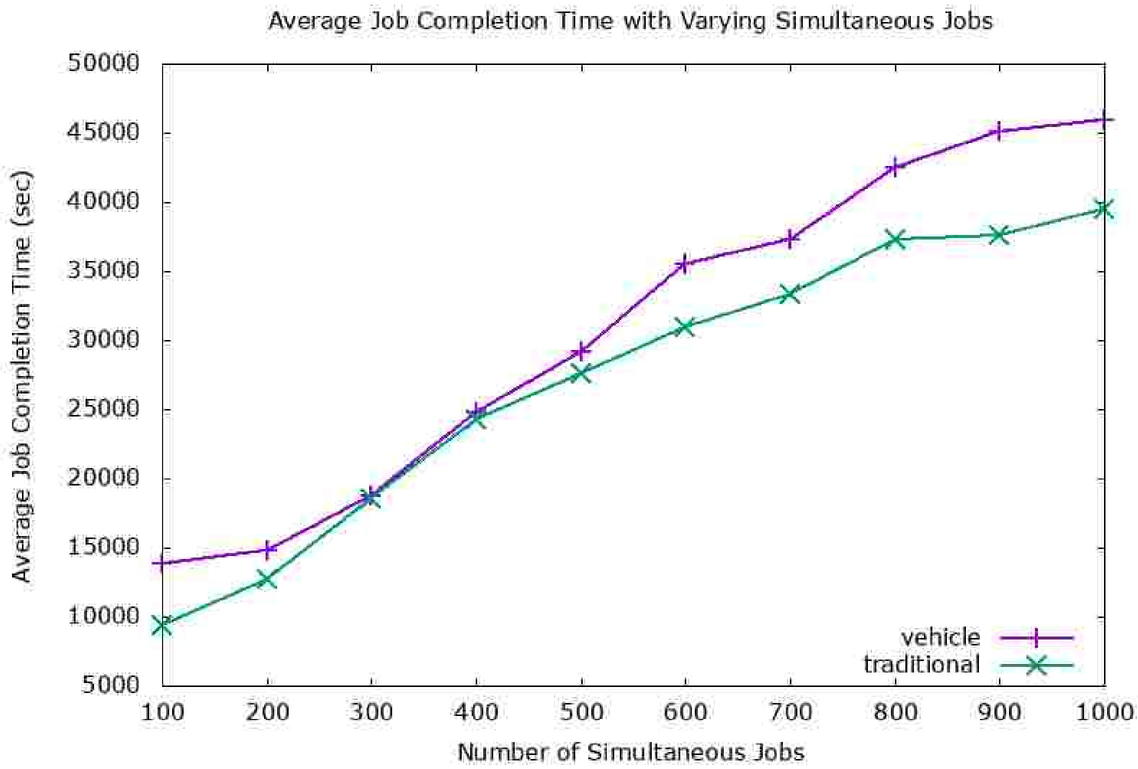


Figure 16: Average Job Completion Time with Varying Simultaneous Jobs

Figures 16 displays the average completion time of jobs for both the vehicle and traditional datacenter simulations. While the average completion times are higher for the vehicle datacenter than for the traditional, the gap is not as significant as one might expect. Furthermore, between 300 and 400 simultaneous jobs there appears to be no real difference between the two. Figure 17 displays the number of completed jobs for both the vehicle and traditional datacenter simulations. The results are somewhat unexpected. It appears that the vehicle outperforms the traditional model. Further analysis shows that simultaneous jobs greater than 200 results in a shortage of resources and both the traditional and vehicle datacenter are waiting on available resources. This is since every worker has 2 backup workers. For the current case of 5 workers assigned to each job, this consumes a total of 15 vehicles (5 workers with 10 backup workers) for each job.

For 200 simultaneous jobs, this is a total of 3000 needed vehicles with only 2560 available. With no conflict of resources, the traditional datacenter far exceeds the vehicle datacenter.

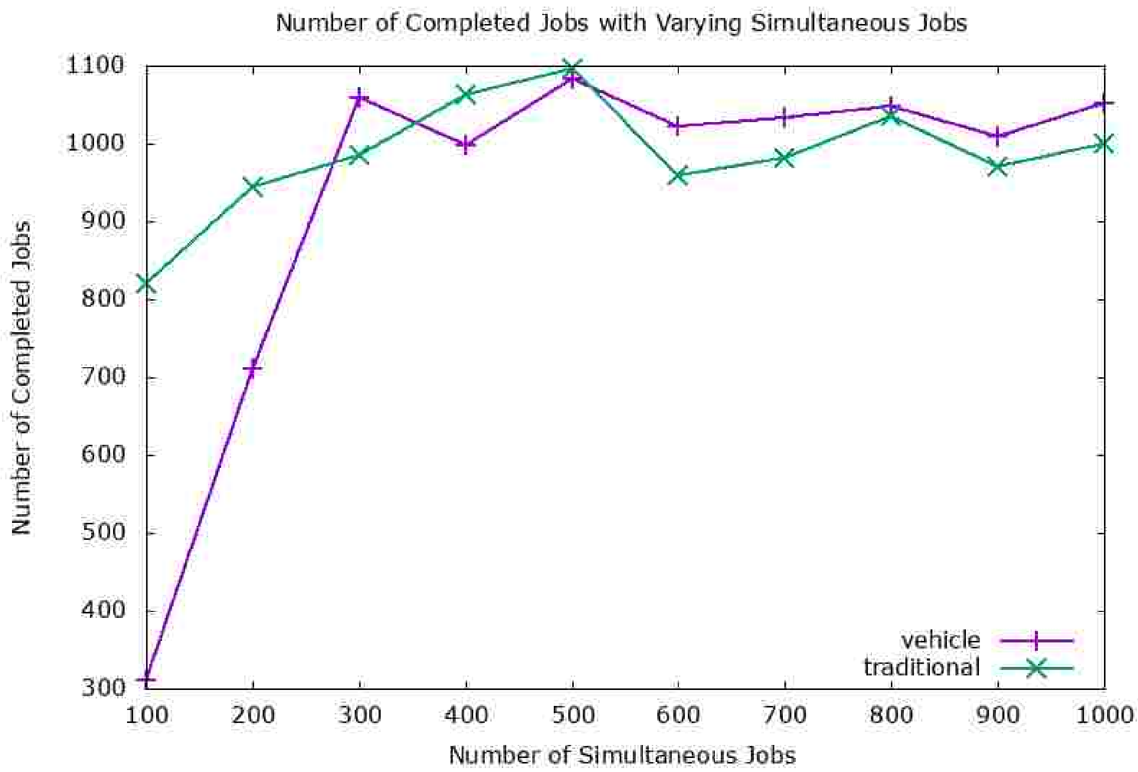


Figure 17: Number of Completed Jobs with Varying Simultaneous Jobs

CHAPTER 7

MAJOR CONTRIBUTIONS

This Thesis developed a simple vehicle datacenter solution based upon Smart Vehicles in a parking lot. While previous work had developed similar models based upon the idea of migration of jobs due to residency of the vehicles, this model assumed that residency times cannot be predicted and therefore no migration is utilized. To offset the migration of jobs, a divide-and-conquer approach was created. This used a MapReduce process to divide the job into numerous sub-jobs and process the subtask in parallel. Finally, a checkpoint was used between the Map and Reduce phase to avoid loss of intermediate data. This simple model was proven to be viable and serves to baseline a model for further study.

CHAPTER 8

CONCLUSIONS

8.1 CONCLUDING REMARKS

Some interesting results were obtained from this simulation. The first is that under certain conditions, a vehicle datacenter is viable. The next is that a wireless vehicle network performed much better than expected. Finally, the vehicle datacenter performed remarkably well in relation to that of a traditional datacenter.

This model took a simplistic approach to handling Big Data by a utilizing a simple divide and conquer approach. Large jobs are divided into smaller subtasks that can be processed in parallel. This division helps reduce the time requirement for any one node in the datacenter thereby reducing the need for long residency times. The model does not predict residency times. Even with this limitation, is has been shown that a vehicular datacenter could be effectively implemented under certain conditions. It must be noted that with 5 workers per job that the total vehicle allocation for each job is 15 vehicles. This results from every one of the workers having 2 vehicle backups. With this allocation, the 2560 vehicles in the vehicle datacenter become fully allocated around 170 simultaneous jobs. Unsurprisingly, analysis revealed that large job sizes with large number of simultaneous jobs was not viable. However, with the proper throttling of simultaneous jobs, the vehicle datacenter is viable.

The most surprising results indicated that wireless model may not have met the performance of wired model, but it was not a magnitude of order difference as might be expected. The wired model has the vehicles connecting at 1 Gbps while the wireless model has the vehicles connecting at 54 Mbps. Furthermore, the wired model uses a full-duplex connection while the wireless model uses a shared half-duplex connection. The wired connection is 20 times the bandwidth of the wireless connection. However, the average completion times and the number of completed jobs were not proportional to that of the difference in bandwidth. This indicates that while bandwidth is an important factor in the vehicle datacenter model, it is possibly not a major factor. Further advances in wireless technologies will make the wireless model a close performance competitor to that of the wired model.

Finally, the vehicle datacenter performed remarkably well compared to that of the traditional datacenter model. This was derived from the comparison between the vehicular model and a traditional model. As with the comparison between wireless and wired, the comparison between the traditional and vehicle datacenter did not demonstrate the order of magnitude difference that might be expected. In fact, they were rather close in performance. This result alone provides the necessary justification for further study of the vehicle datacenter. This solution is the baseline first step for further improvements towards a versatile and robust solution.

8.2 LOOKING INTO THE CRYSTAL BALL

A baseline has now been conducted and shown that a viable datacenter can be created from collection of vehicles in a parking lot. This opens the door to a wide variety of interesting research. At the very least the vehicle datacenter model can now be refined. Scheduling managers could be used to inject job sizes based upon utilization or prioritization. Migration techniques could be employed to reduce job restarts. With the baseline that has been produced, the cost of implementation of performance improvements can be weighted with the actual performance improvements to determine that viability of these improvements.

The vehicle datacenter could be expanded to dynamic datacenters. This vehicle datacenter has been constructed here in a parking lot that guarantees a specific capacity. However, there are many other occurrences of vehicles coming together. For example, shopping malls and athletic events. What if these vehicles could be organized into dynamic datacenters to handle needed services? In shopping malls, this could run applications supporting the customer's needs. These could deliver advertisements offered dynamically for the stores in the shopping mall. They could notify of the lengths of checkout lines at stores so that customers could adjust their shopping patterns. It must be noted that the need for these applications would be proportional to the number of customers. In other words, the dynamic datacenter would be a good fit for the dynamic need of the applications.

This dynamic need would be even more appropriate for athletic events. As parking lots fill with vehicles and "tail gate parties", finding an available parking space and route to that parking space becomes a daunting task. An application to alleviate this would be greatly needed. Furthermore, finding one's seat can be difficult enough without adding numerous other people trying to find their seat causing pedestrian congestion. What if an application existed to guide one efficiently to their seat? This would then reduce the number of seating attendants needed. Finally, the most important aspects of all athletic events are that of food and restrooms. No one wants to have to wait long periods of time waiting for either. An application displaying the lines at all concessions and restrooms could limit the time away from the event increasing the enjoyment of the attendees.

One of the most exciting technologies on the horizon is that of the smart city. Smart vehicles will play an important role in the creation of smart cities. Smart vehicles will fully utilize their abilities. Smart vehicles will be processing nodes, sensor nodes, data aggregator nodes, and consuming nodes. The datacenter has been shown to utilize the smart vehicle as a processing node. With just this, a smart city with total communication coverage could make every vehicle a processing node. With constant communications, residency is not an issue. With populations being easily in the hundreds of thousands of people with similar numbers of smart vehicles, the datacenter becomes enormous. Think of the computing power of 200,000 processing nodes.

The smart vehicle also has sensor capabilities. Add to this the data aggregation capability of smart vehicles and there now exists a powerful tool for the smart city. In the case of traffic patterns, the smart vehicle could aggregate data from other vehicles to provide recommendations to the smart city to alter traffic lights. For example, someone is sitting at a red light and seeing the next light in their path show green while no traffic approaches only to have that further light turn red when their light turns green is a great annoyance and a cause of congestion. Smart vehicles could help the smart city optimize the use of signals so that traffic is nearly always flowing through traffic signals.

Of course, the smart vehicle would be a consumer of data. Self-driving cars are becoming more and more a reality. Smart vehicles in smart cities would be self-driving. Route selection, traffic avoidance, and parking are all consuming data that a vehicle would require.

A very promising variation of the vehicle datacenter is that of hybrid storage. If there existed a central storage facility that allowed all vehicles to mount external storage, then virtualization becomes a viable option for all vehicles regardless of their transient nature. A vehicle mounts the external storage and executes a virtual machine image that is stored on that external storage. This reduces the "spin-up" time of the virtual machine since the entire operating system does not have to be downloaded. Furthermore, any data that needs to be processed is accessed and saved on the external storage. This prevents the necessity of downloading a large data set before processing the data. The most important aspect of this is that a memory file of the working virtual machine is kept on the external storage. This allows a vehicle to in theory pick up a terminated virtual machine from another vehicle quickly and efficiently.

This idea holds great potential when working in conjunction with small footprint operating systems. These small virtual machines can be created to support individual applications making them extremely small. This would allow a smart city processing manager to assign these virtual machines to vehicles with no concern of loss of data or functionality. If a vehicle is suddenly removed from the network, the virtual machine, memory file, and data are assigned to another vehicle. This vehicle would then launch the virtual machine, and memory file, and continue processing data with little time interruption nor loss of data.

This would also be a great benefit to smart vehicles that are acting as data aggregators. Vehicles will collect a large amount of data. The vehicle will need to decide what data is worth keeping and what data is worth discarding. This sometimes requires applications to process both local and external data to find the relevance of the local data. These applications could be small virtual machines that the vehicle "spins-up" to perform the analysis of the data. Finally, any processed data is then stored on the external storage thereby saving space on the vehicle and assuring that the processed data is persistent and available to others.

REFERENCES

[1] eMarketer, "Tablet users to surpass 1 billion worldwide in 2015," January 2015. [Online]. Available: http://www.emarketer.com/Article/Tablet-Users-Surpass-1-Billion-Worldwide-2015/1011806.

[2] eMarketer, "Worldwide smartphone usage grow 25% in 2014," June 2014. [Online]. Available: http://www.emarketer.com/Article/Worldwide-Smartphone-Usage-Grow-25-2014/1010920.

[3] Directive Outsource IT Chase, C., "The internet of things as the next big thing," June 2013. [Online]. Available: http://www.directive.com/ blog/item/the-internet-of-things-as-the-next-big-thing.html.

[4] WIRED Amyx, S., "Why the internet of things will disrupt everything," July 2014. [Online]. Available: http://innovationinsights.wired.com/insights/2014/07/internet-things-will-disrupt-everything.

[5] Advanced Network Systems, "Global mobile data traffic forecast update, 2014-2019," March 2015. [Online].

[6] IBM Research Zurich Haig A Peter, "Data at the edge, ibm global technology outlook," 2015. [Online]. Available: http://www-935.ibm.com/services/multimedia/Vortrag_IBM_Peter-Krick.pdf.

[7] Storedot, [Online]. Available: http://www.store-dot.com.

[8] uBeam, [Online]. Available: http://ubeam.com.

[9] Qualcomm, "Lte direct proximity services," [Online]. Available: https://www.qualcomm.com/invention/technologies/lte/direct.

[10] WiFi Alliance, "Discover wi-fi - wi-fi direct," [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wi-fi-direct.

[11] WiFi Alliance, "Discover wi-fi - wi-gig certified," [Online]. Available: http://www.wi-fi.org/discover-wi-fi/wigig-certified.

[12] opensource.com, "What is docker?," [Online]. Available: http://opensource.com/resources/what-docker.

[13] J. Morgan, "A Simple Explanation Of 'The Internet Of Things'," Forbes, 13 May 2014. [Online]. Available: https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#572153f71d09. [Accessed 18 December 2017].

[14] L. S. a. R. H. Erica Fink, "Your Hackable House," CNN, 2017. [Online]. Available: http://money.cnn.com/interactive/technology/hackable-house/. [Accessed 18 December 2017].

[15] A. Greenberg, "Hackers Remotely Kill a Jeep on the Highway—With Me in It," Wired, 15 July 2015. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/. [Accessed 18 December 2017].

[16] A. F. Mohammed, V. T. Humbe and S. S. Chowhan, "A review of Big Data environment and its related technologies," in *International Conference on*

Information Communication and Embedded Systems (ICICES), Chennai, India, 2016.

[17] M. Hilbert and P. Lpez, "The world's technological capacity to store, communicate, and compute information," *Science,* vol. 332, no. 6025, pp. 60-65, 2011.

[18] L. Columbus, "Roundup Of Internet Of Things Forecasts And Market Estimates, 2016," Forbes, 27 November 2016. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2016/11/27/roundup-of-internet-of-things-forecasts-and-market-estimates-2016/#50251a66292d. [Accessed 16 January 2018].

[19] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall and W. Vogels, "Dynamo: Amazons highly available key-value store," in *23-th ACM Symposium on Operating Systems Principles, (SOSP07),* Stevenson, Washington, 2007.

[20] L. A. Barroso, J. Clidaras and U. Holzle, The datacenter as a computer: An introduction to the design of warehouse-scale machines, 2nd edition ed., San Rafael, California: Morgan & Claypool, 2013.

[21] D. C. Marinescu, Complex Systems and Clouds: A Self-Organization and Self-Management Perspective, Elsevier: Morgan Kaufman, 2016.

[22] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM,* vol. 51, no. 1, p. 107–113, January 2008.

[23] J. L. Hennessy and D. A. Patterson, Computer Architecture a Quantitative Approach, Elsevier: Morgan Kaufman, 2012.

[24] C. J. Shafer, S. Rixner and A. L. Cox, "The Hadoop distributed file system," in *IEEE Symposium on Mass Storage Systems and Technologies, (MSST10),* Lake Tahoe, Nevada, 2010.

[25] B. K. Shvachko, H. Kuang, S. Radia and R. Chansler, "The Hadoop distributed file system: Balancing portability and performance," in *IEEE International Symposium on Performance Analysis of Systems and Software, (ISPASS10),* White Plains, NY, 2010.

[26] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Inc., 2009.

[27] X. Liu, "Understanding Big Data Processing and Analytics," 19 September 2013. [Online]. Available: https://www.developer.com/db/understanding-big-data-processing-and-analytics.html.

[28] "What is cloud computing?," Microsoft Inc., 2018. [Online]. Available: https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/.

[29] Z. Kerravala, "How a data center works, today and tomorrow," Network World, 25 September 2017. [Online]. Available: https://www.networkworld.com/article/3223692/data-center/what-is-a-data-center-architecture-components-standards-infrastructure-cloud.html.

[30] "Amazon web services," Amazon Inc, 2010. [Online]. Available: http://aws.amazon.com.

[31] "Google app engine," Google, Inc, 2010. [Online]. Available: http://code.google.com/appengine/, 2010.

[32] "Windows azure," Microsoft Corporation, 2010. [Online]. Available: http://www.microsoft.com/windowazure/.

[33] K. Hwang, G. Fox and J. Dongarra, "Cloud Architecture and Datacenter Design," 2 May 2010. [Online]. Available: https://edisciplinas.usp.br/pluginfile.php/98907/mod_resource/content/1/Chapter7-Cloud-Architecture-May2-2010.pdf.

[34] M. Eltoweissy, S. Olariu and M. Younis, "Towards Autonomous Vehicular Clouds," in *Proceedings of AdHocNets'2010*, Victoria, BC, Canada, August 2010.

[35] S. Olariu, I. Khalil and M. Abuelela, "Taking VANET to the Clouds," *International Journal of Pervasive Computing and Communications*, vol. 7, no. 1, p. 7–21, February 2011.

[36] M. Whaiduzzaram, M. Sookhak, A. Gani and R. Buyya, "A Survey of Vehicular Cloud Computing," *Journal of Network and Computer*, vol. 40, p. 325–344, 2014.

[37] S. Arif, J. W. Olariu, G. Yan, W. Yang and I. Khalil, "Datacenter at the airport: Reasoning about time-dependent parking lot occupancy.," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2067-2080, 2012.

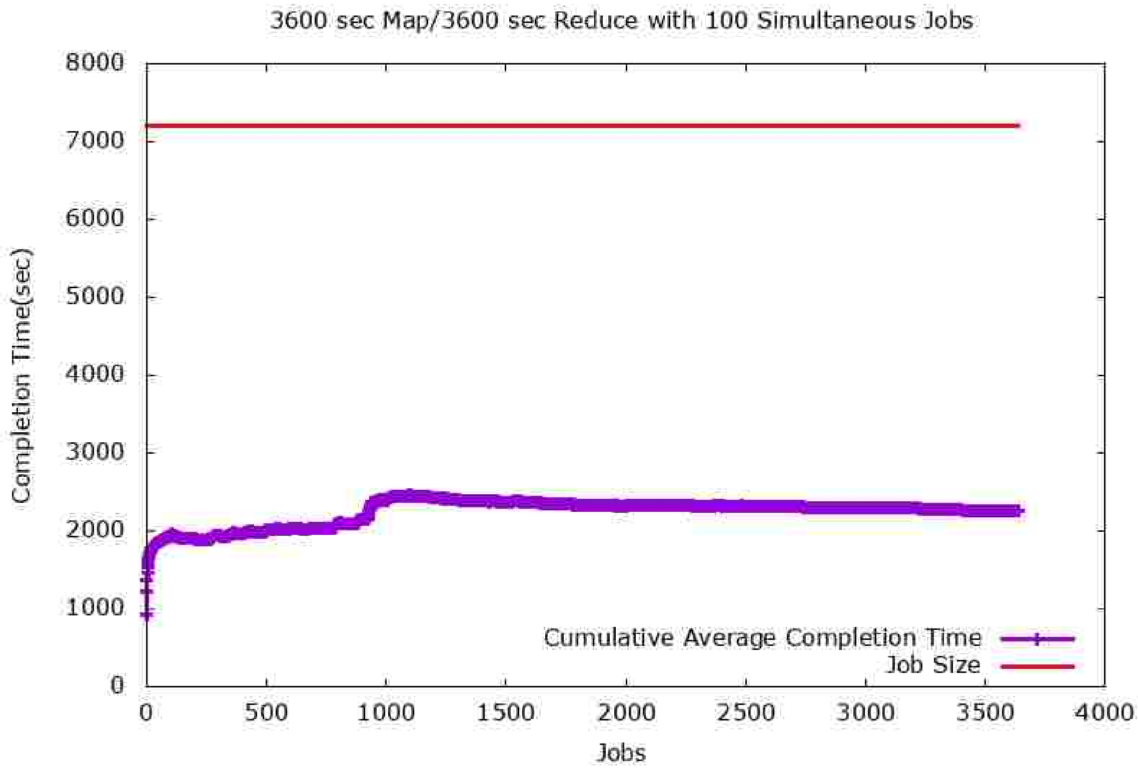
[38] N. Vignesh, S. Shankar, S. Sathyamoorthy and V. M. Rajam, "Value added services on stationary vehicular cloud.," *In Distributed Computing and Internet Technology*, vol. 8337, pp. 92-97, 2014.

[39] R. Hussain, F. Abbas, J. Son and H. Oh, "TIaaS: Secure cloud-assisted traffic information dissemination in vehicular ad hoc networks," *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium*, p. 178–179, May 2013.

[40] W. He, G. Yan and L. D. Xu, "Developing vehicular data cloud services in the IoT environment.," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587-1595, May 2014.

[41] R. Florin, S. Abolghasemi, A. G. Zadeh and S. Olariu, "Big Data in the Parking Lot," in *Big Data: Management, Architecture, and Processing*, New York, Taylor and Francis, 2017.

APPENDIX A: JOB COMPLETION TIMES



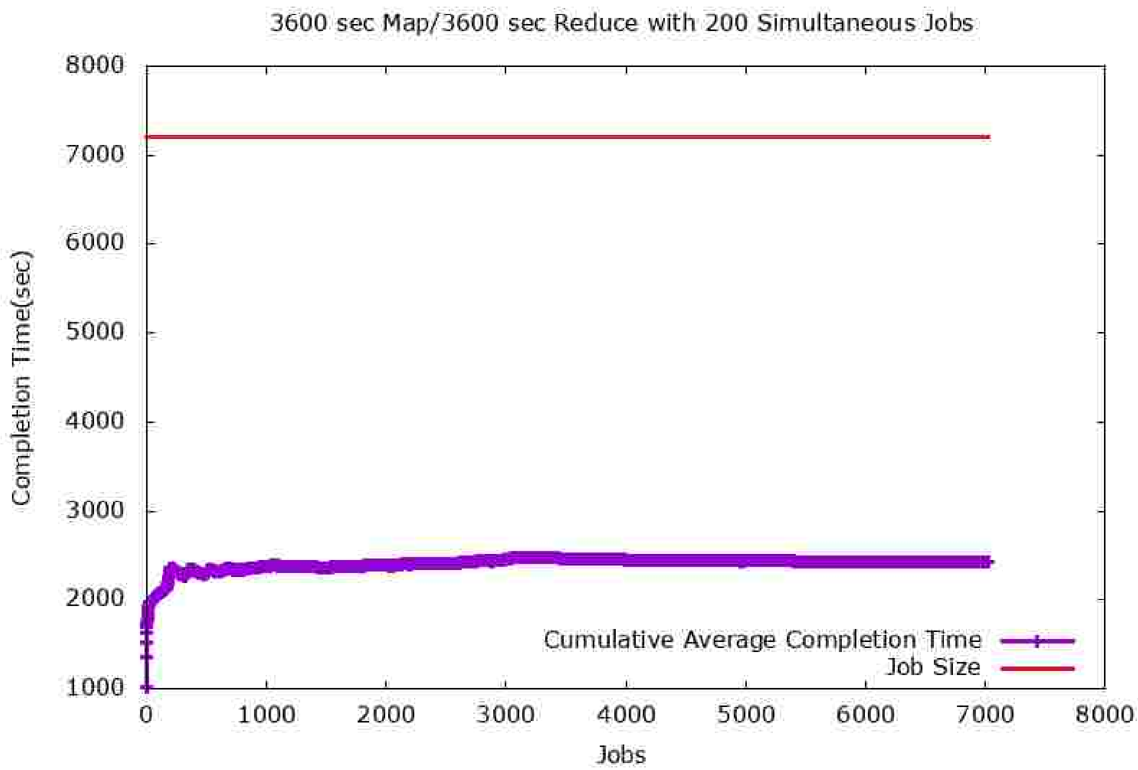
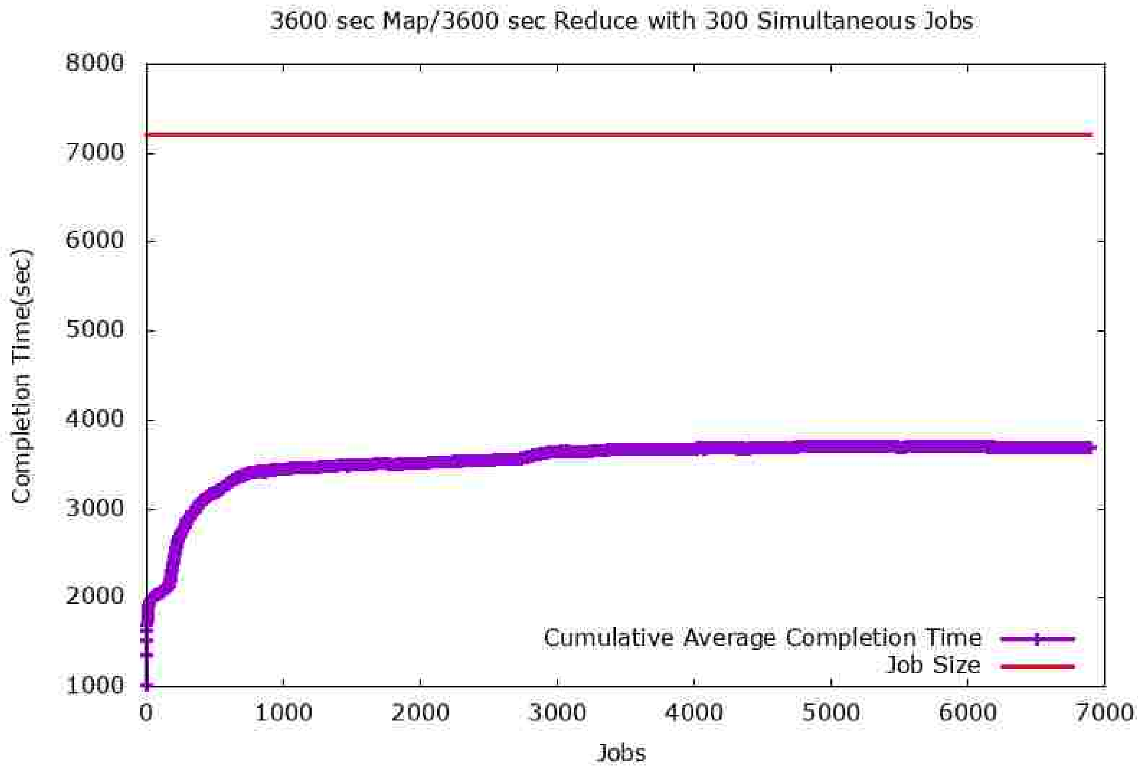


Figure 18: Job Completion Times



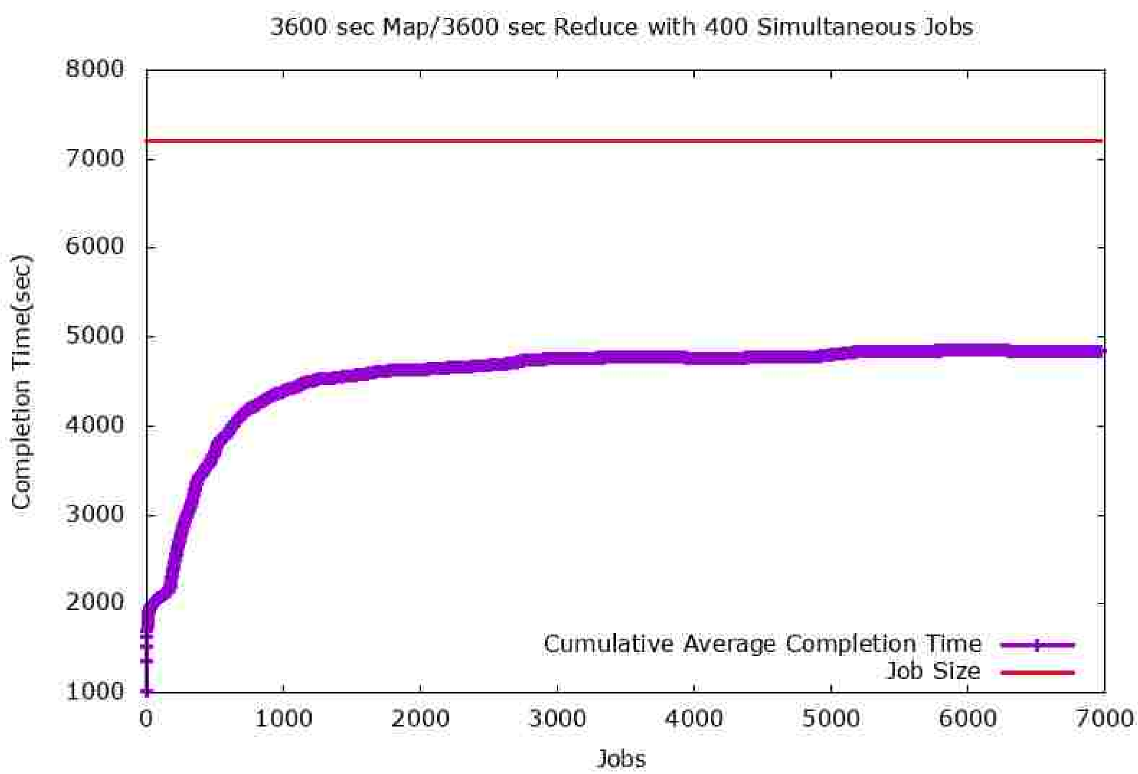
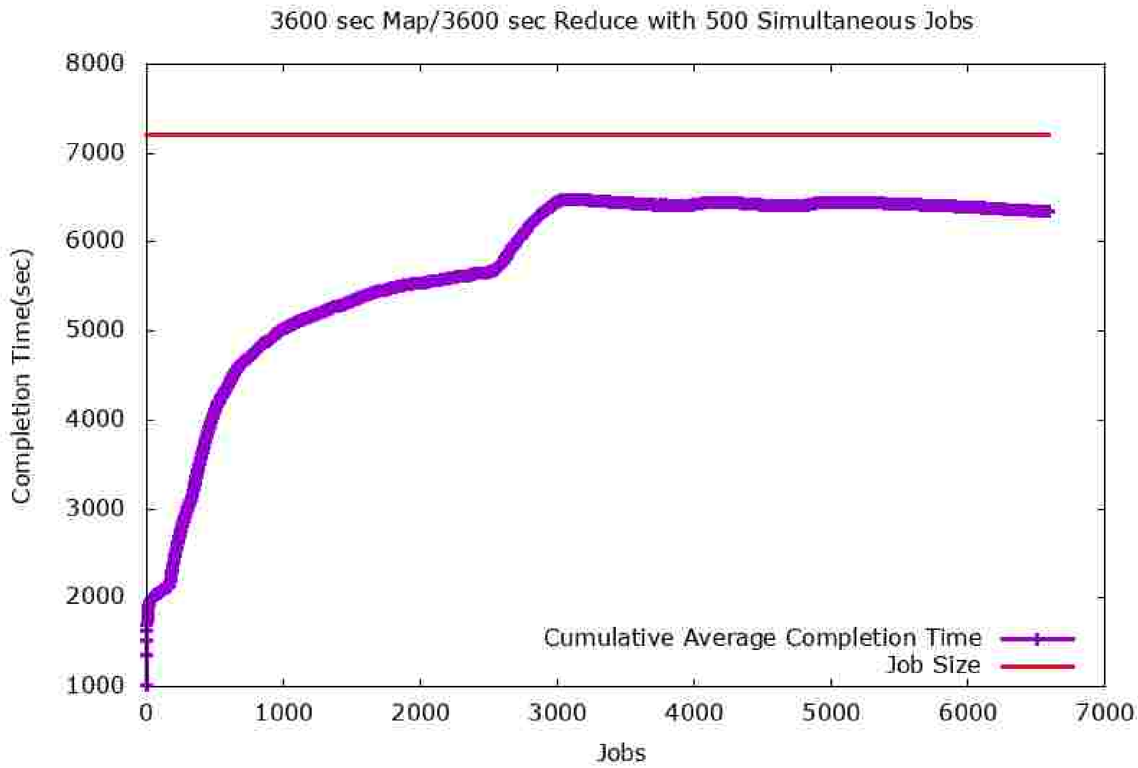


Figure 19: Job Completion Times



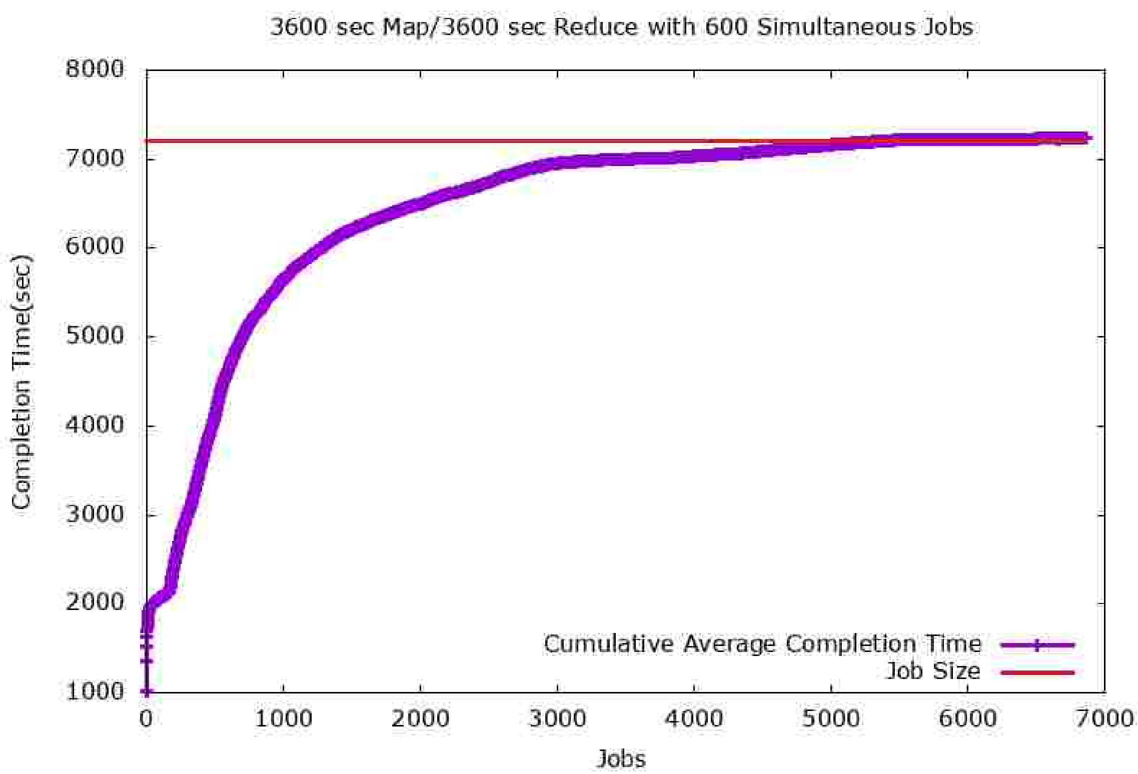
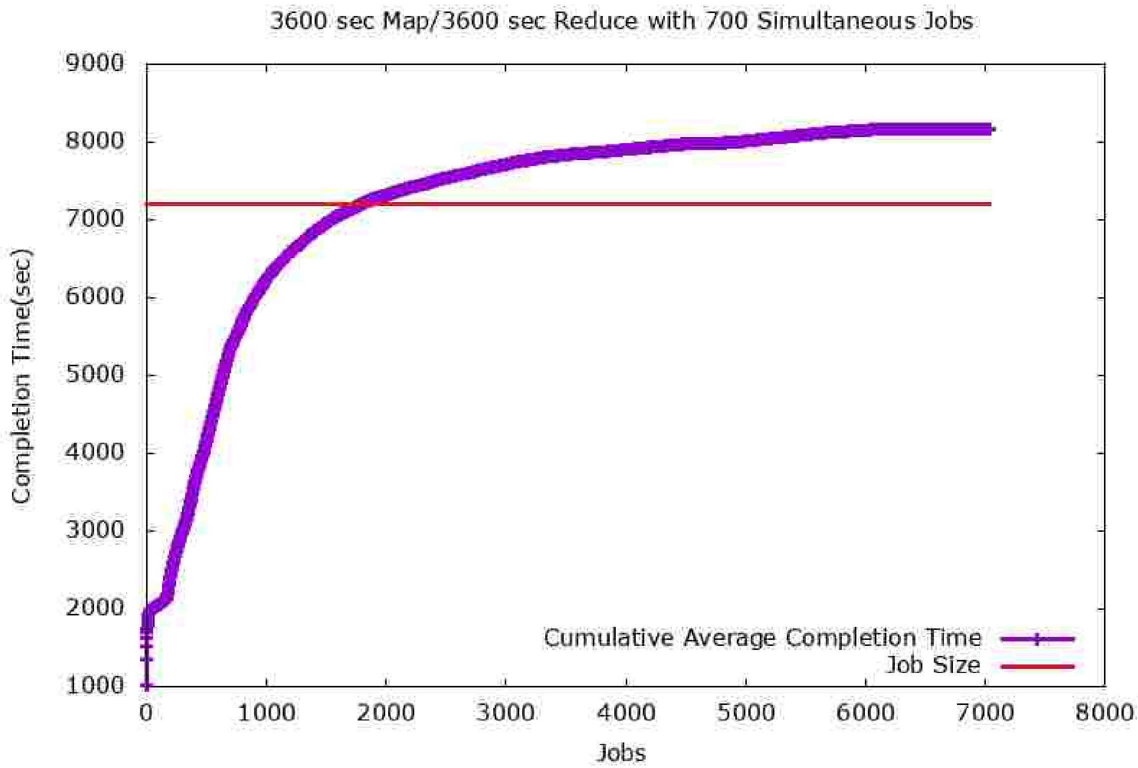


Figure 20: Job Completion Times



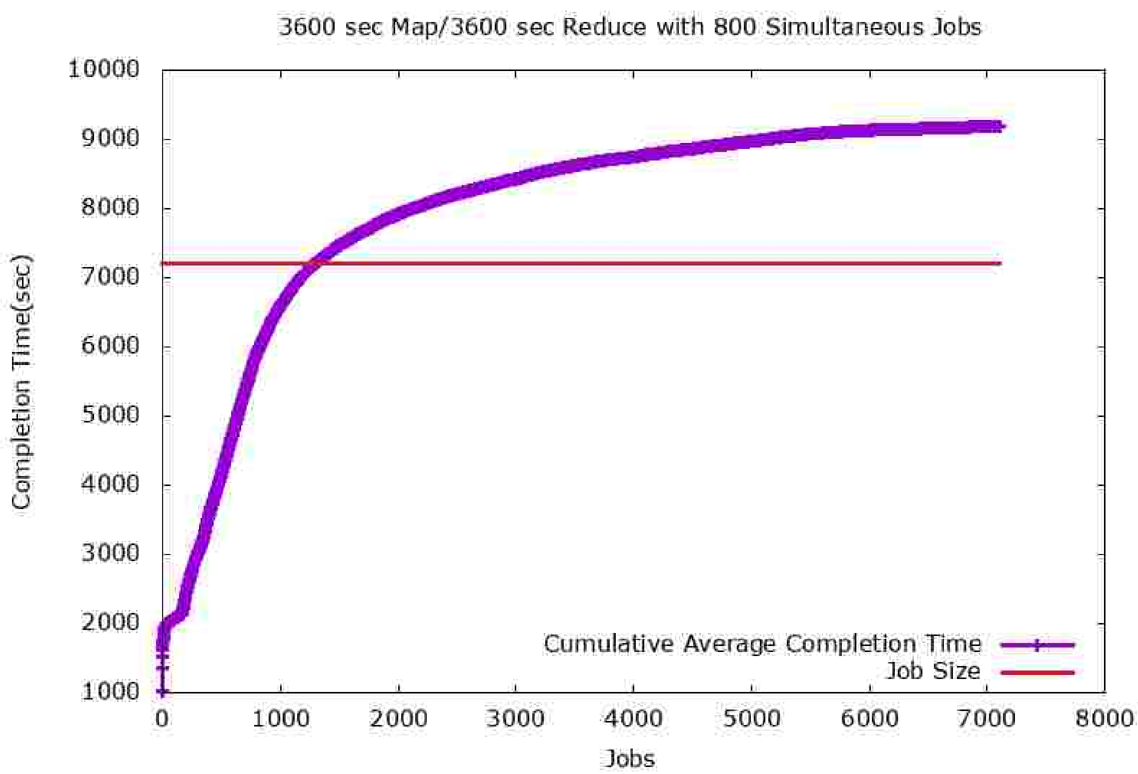
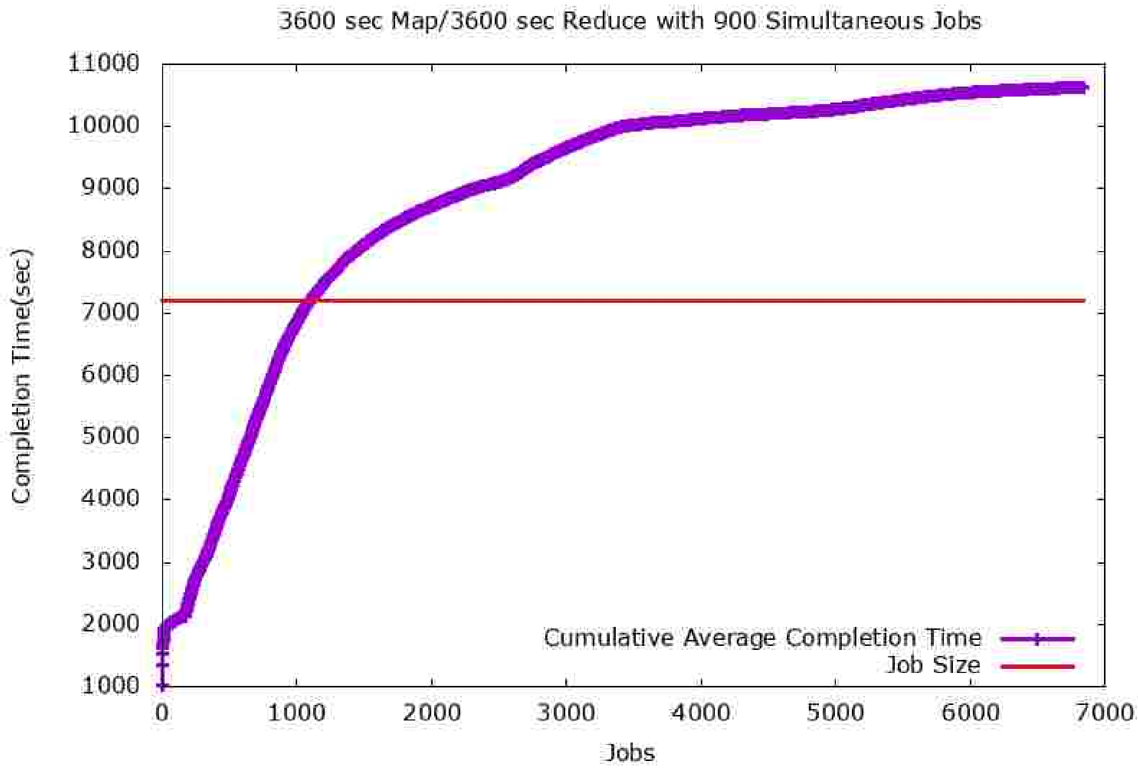


Figure 21: Job Completion Times



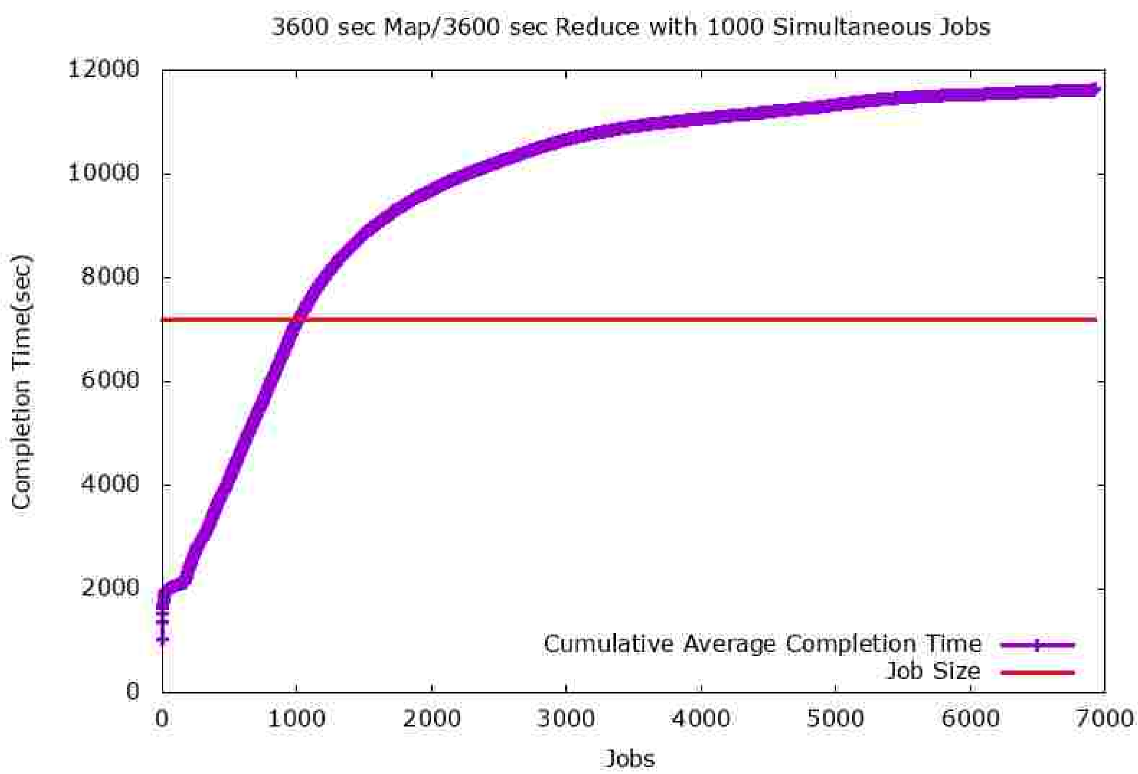
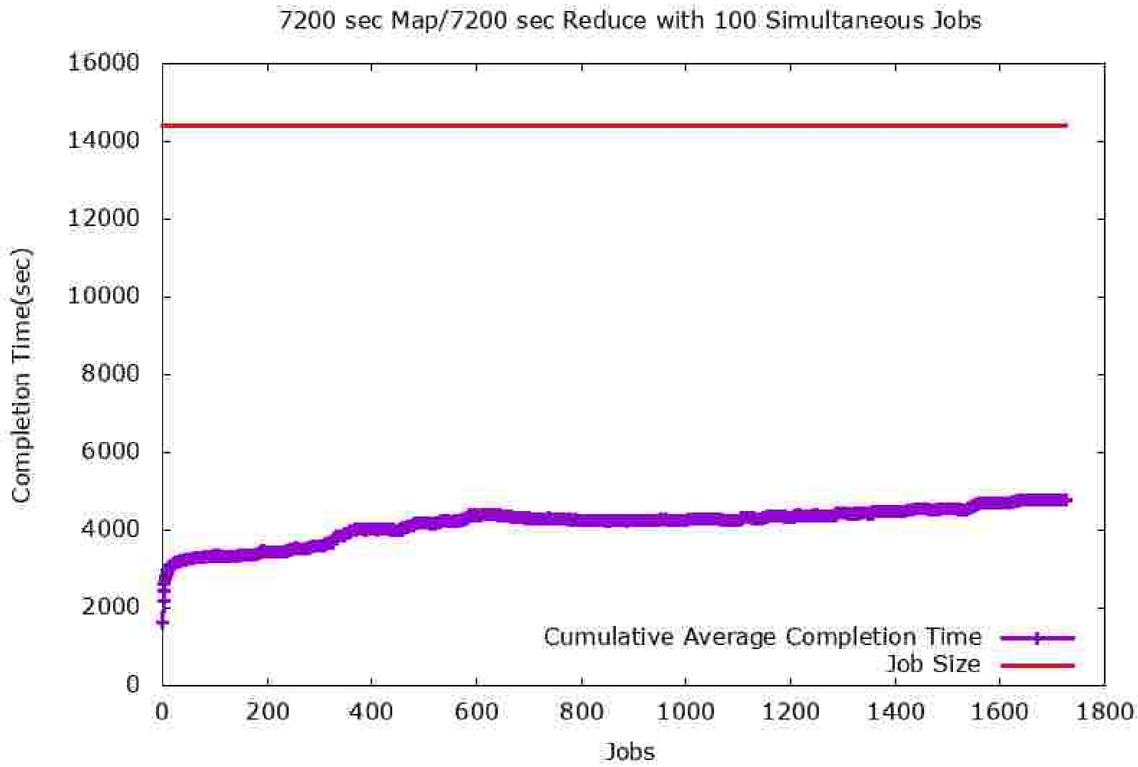


Figure 22: Job Completion Times



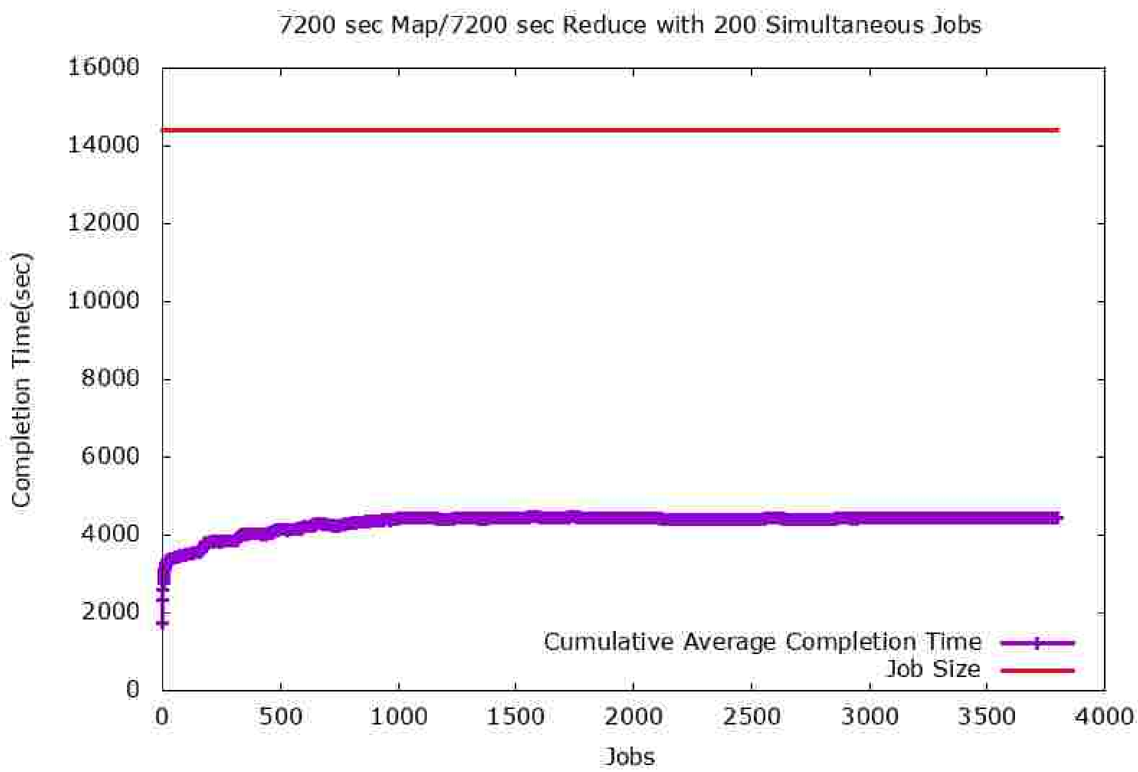


Figure 23: Job Completion Times

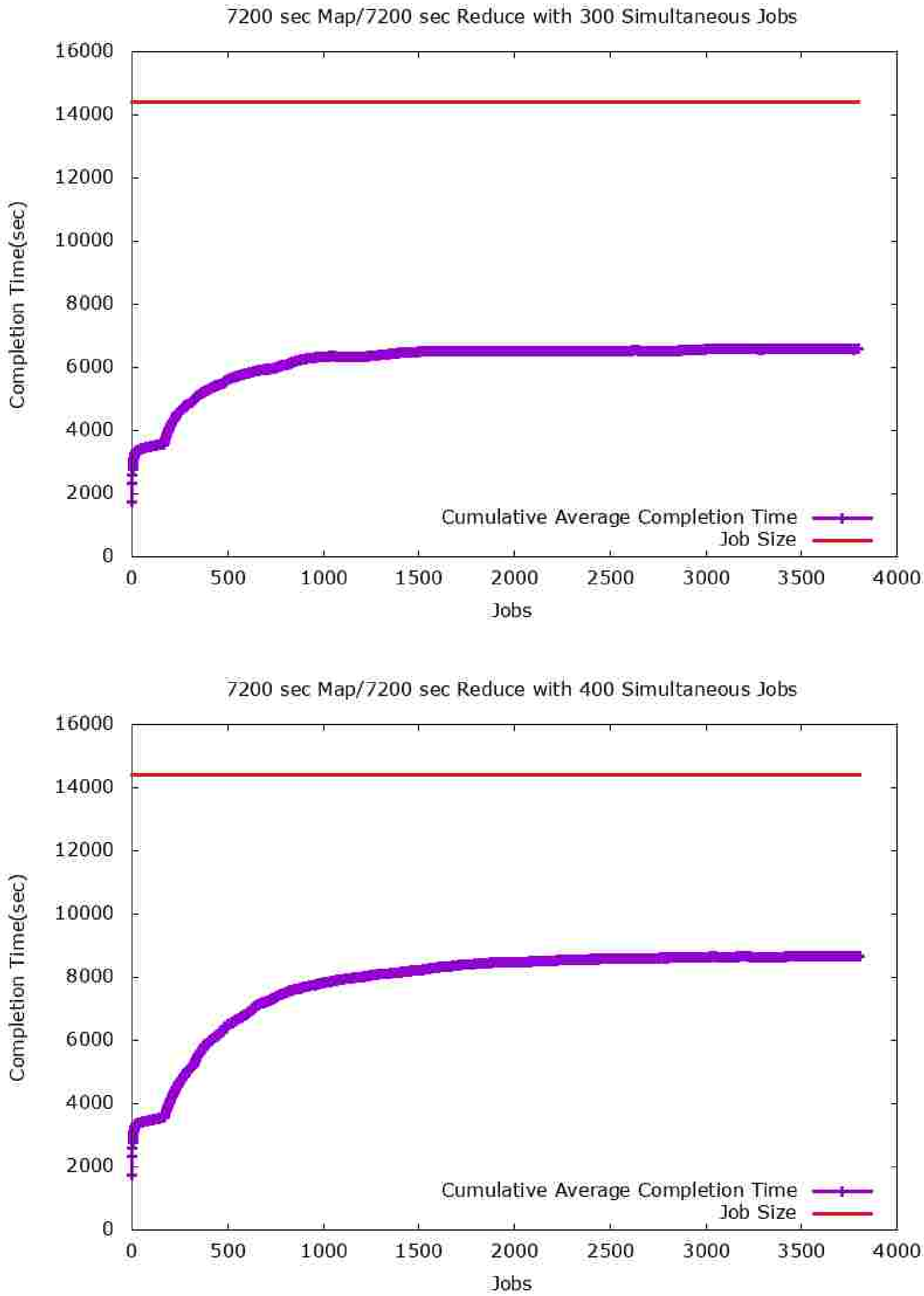


Figure 24: Job Completion Times

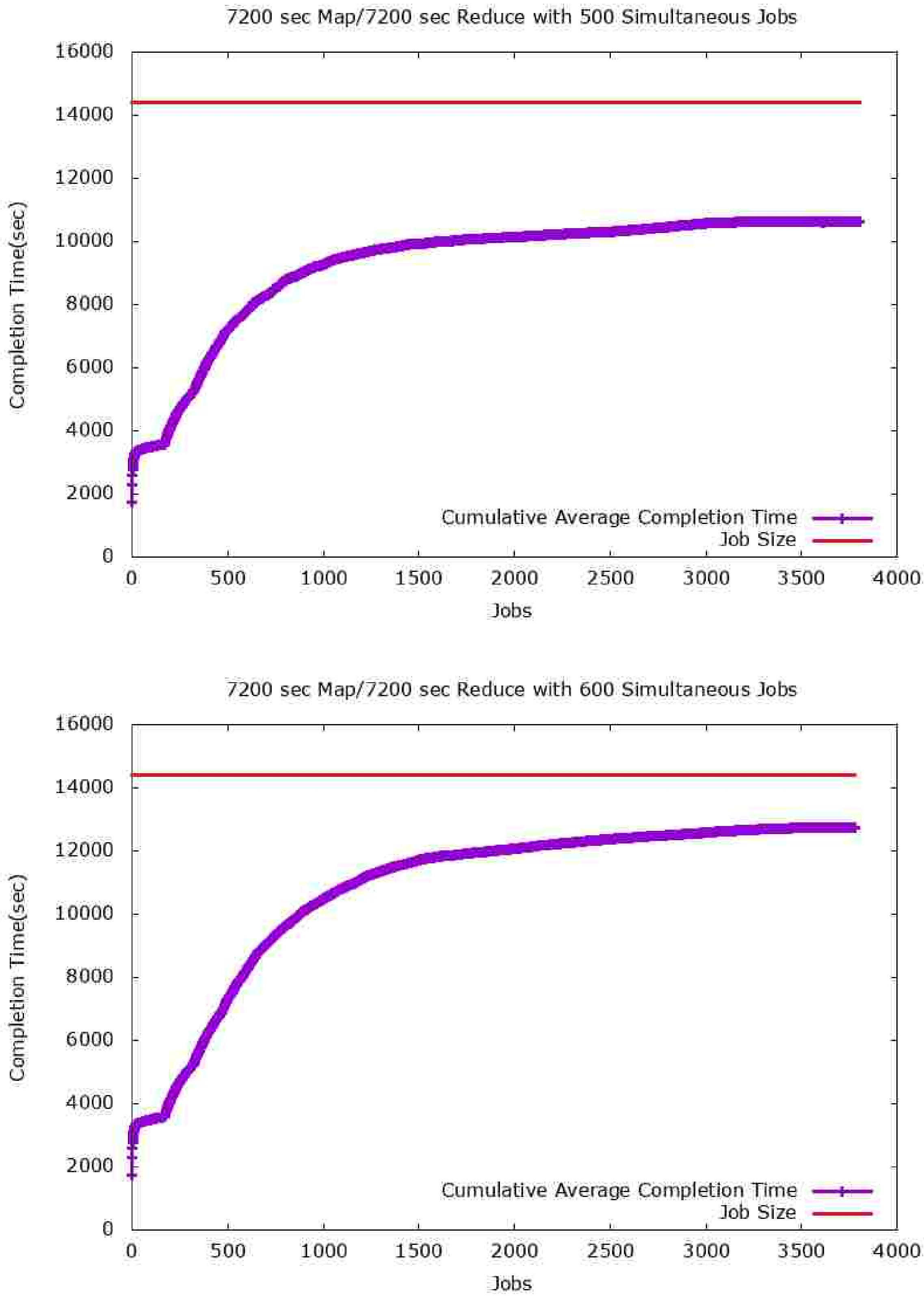


Figure 25: Job Completion Times

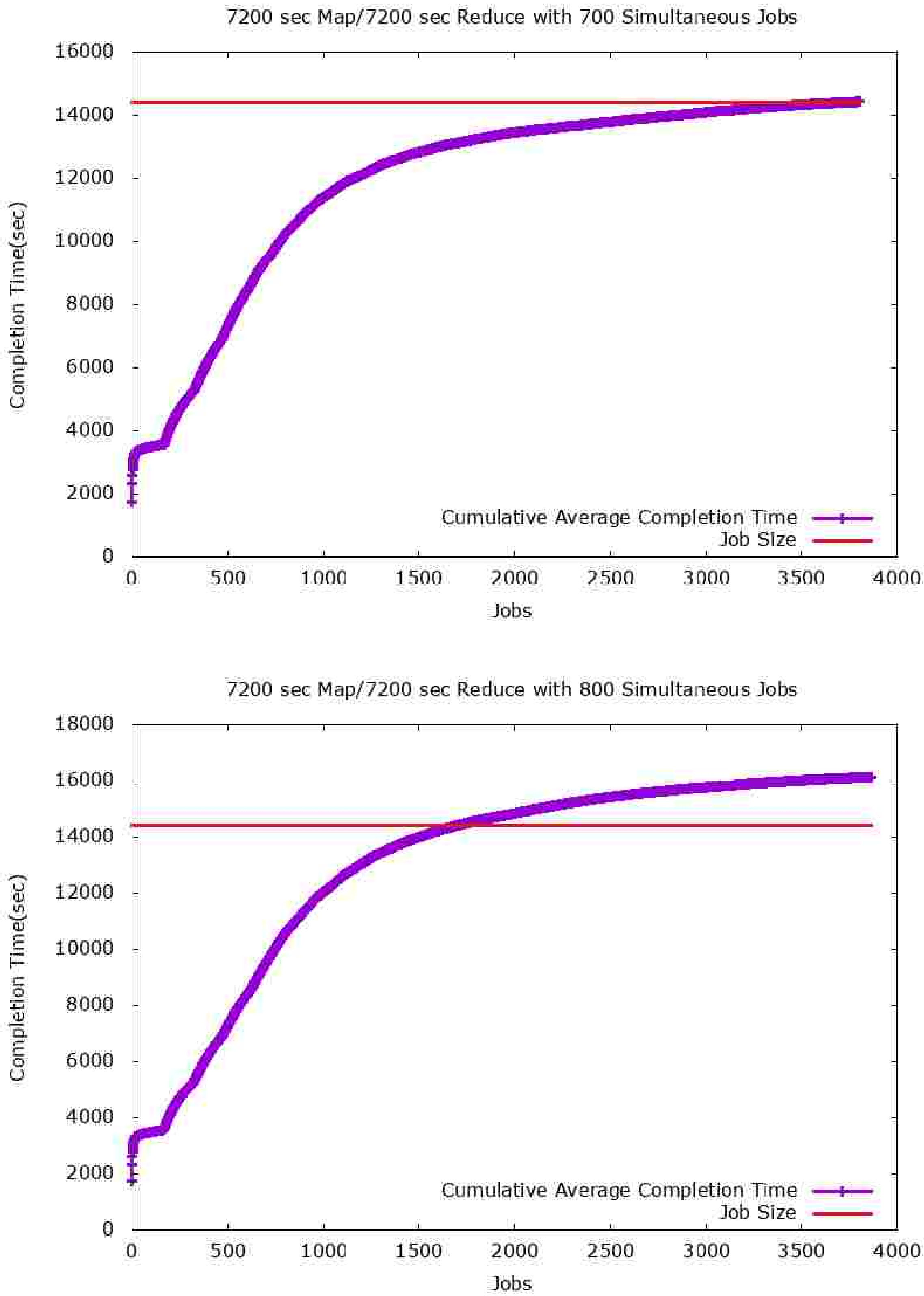
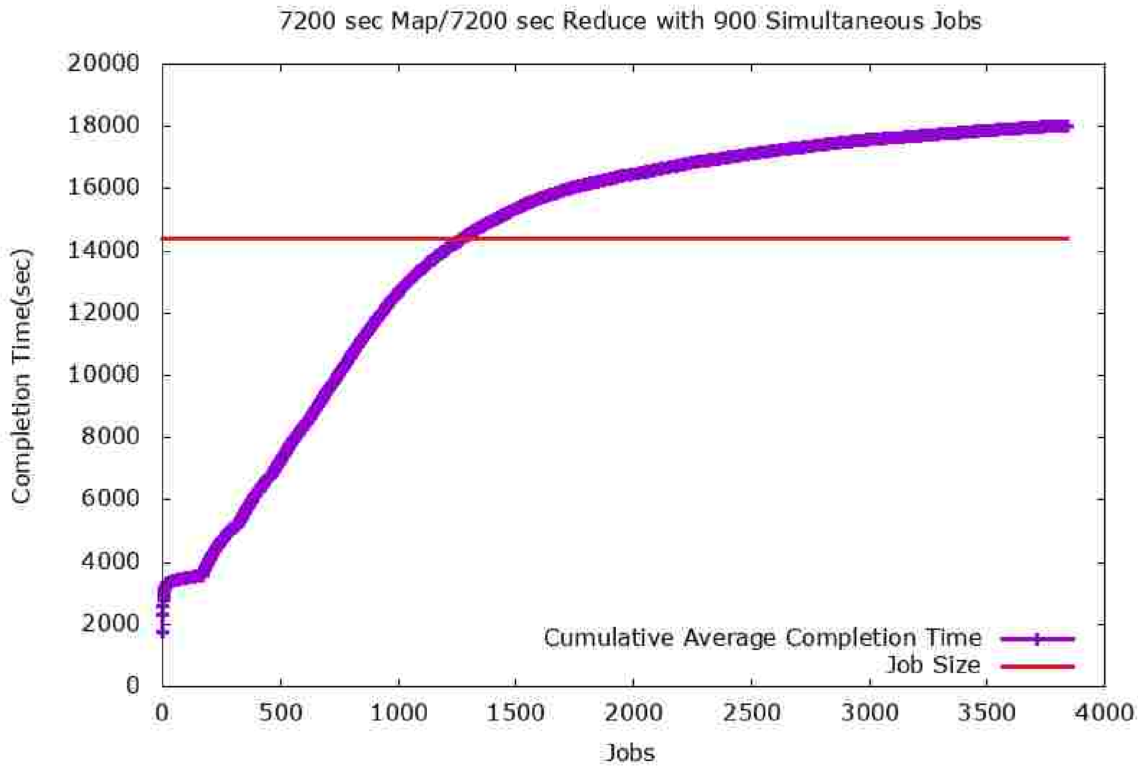


Figure 26: Job Completion Times



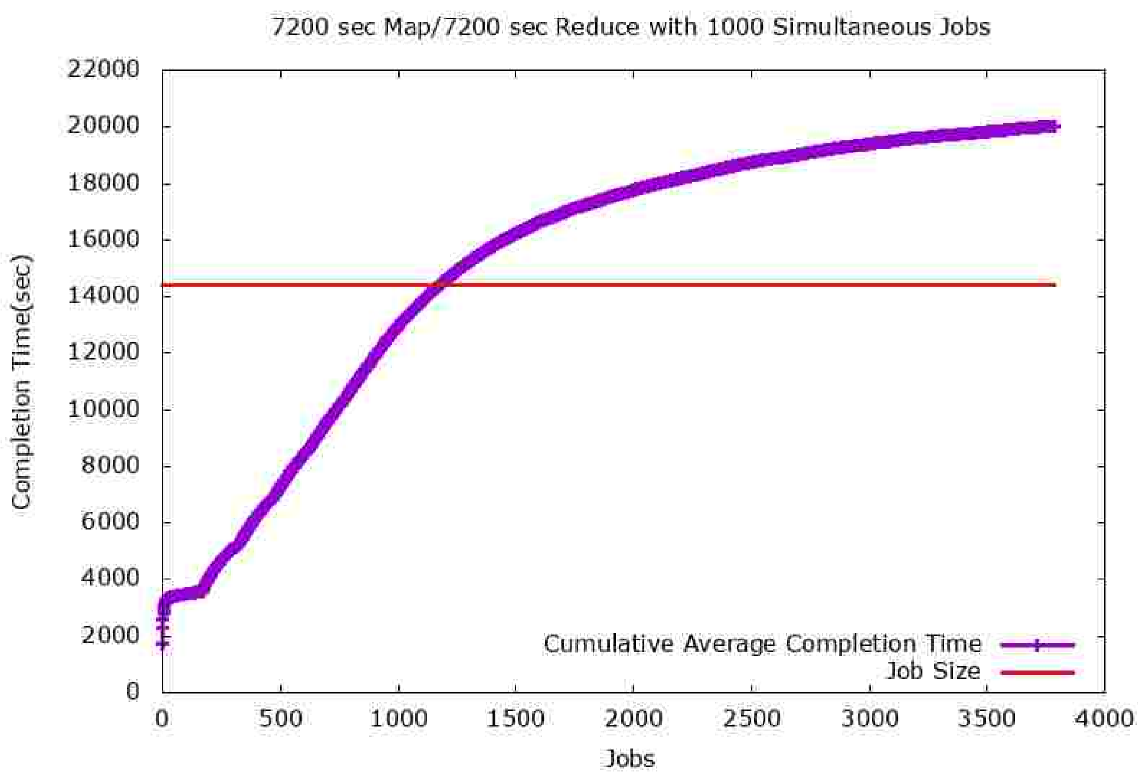
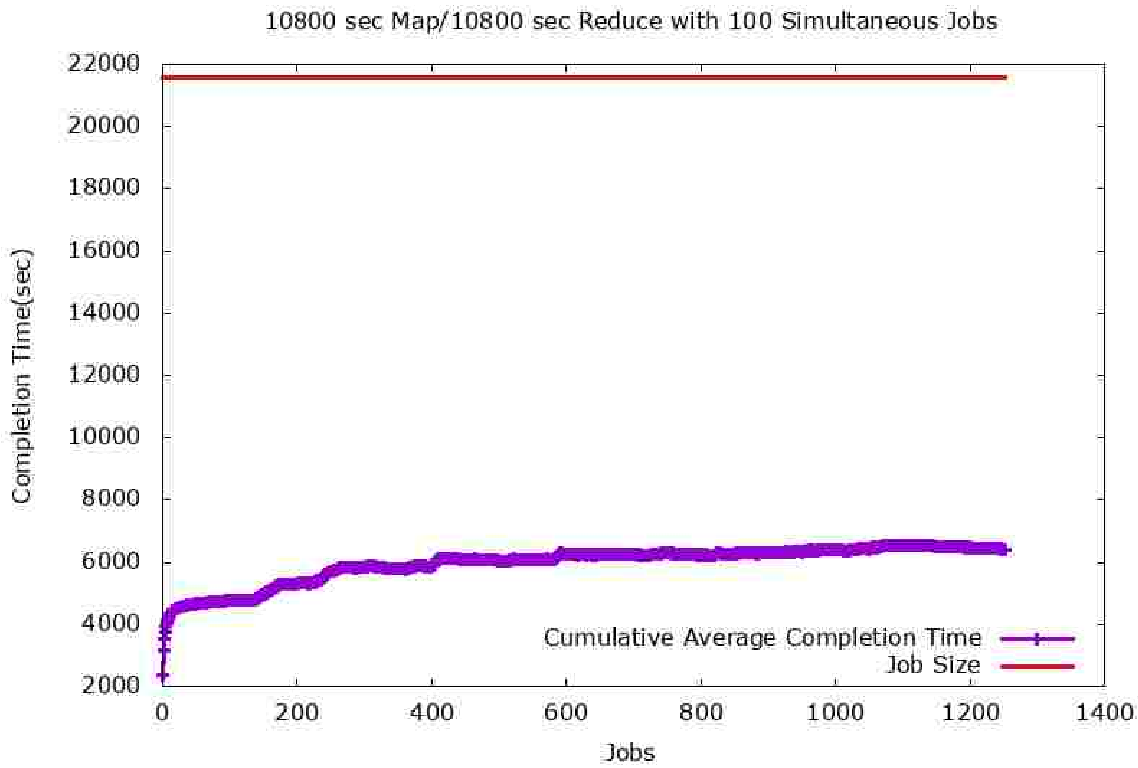


Figure 27: Job Completion Times



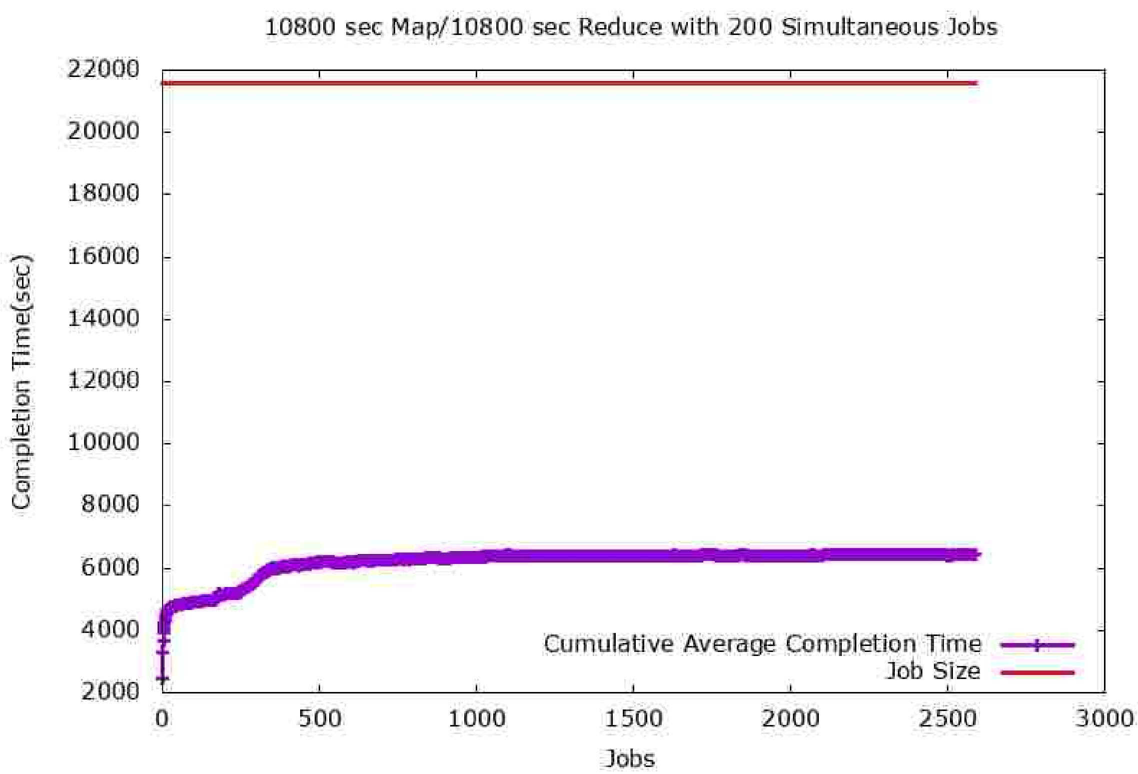
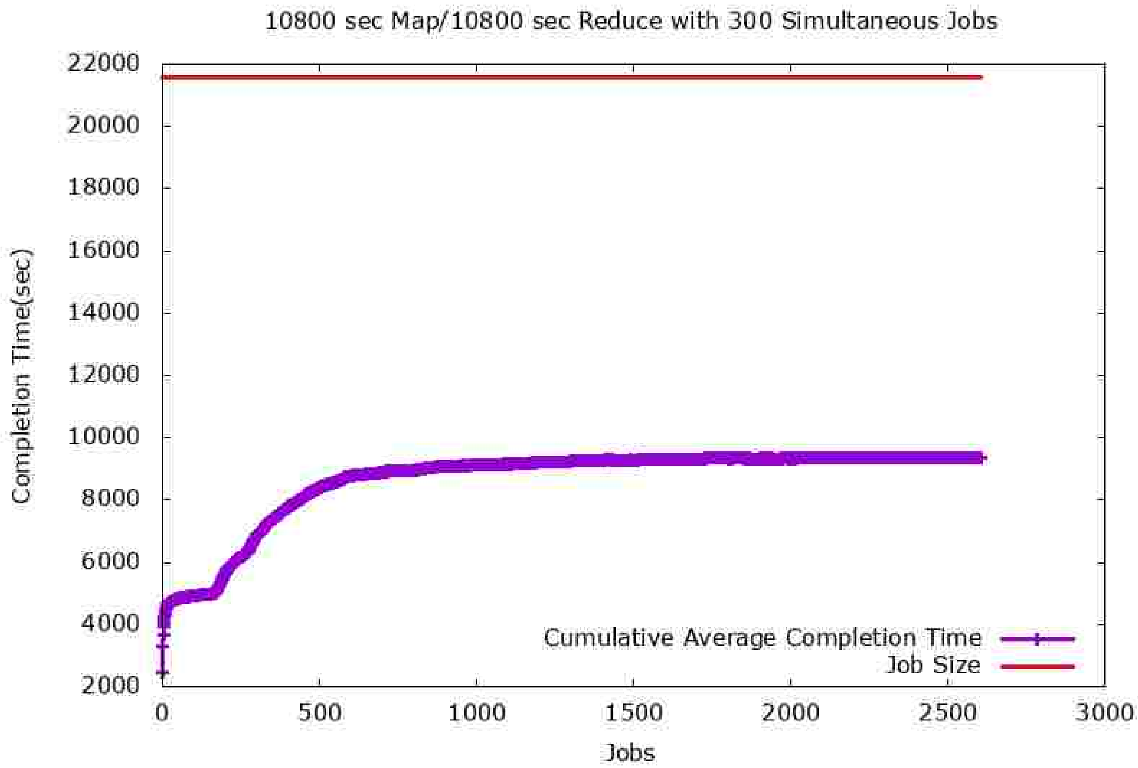


Figure 28: Job Completion Times



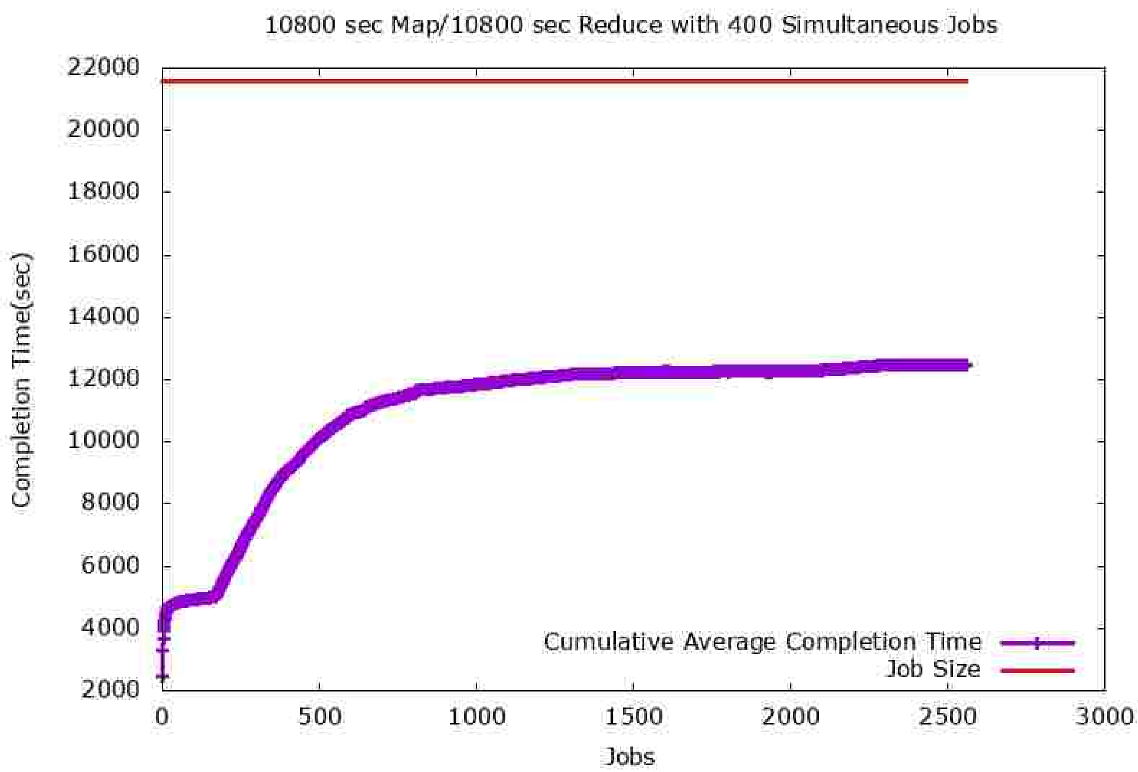


Figure 29: Job Completion Times

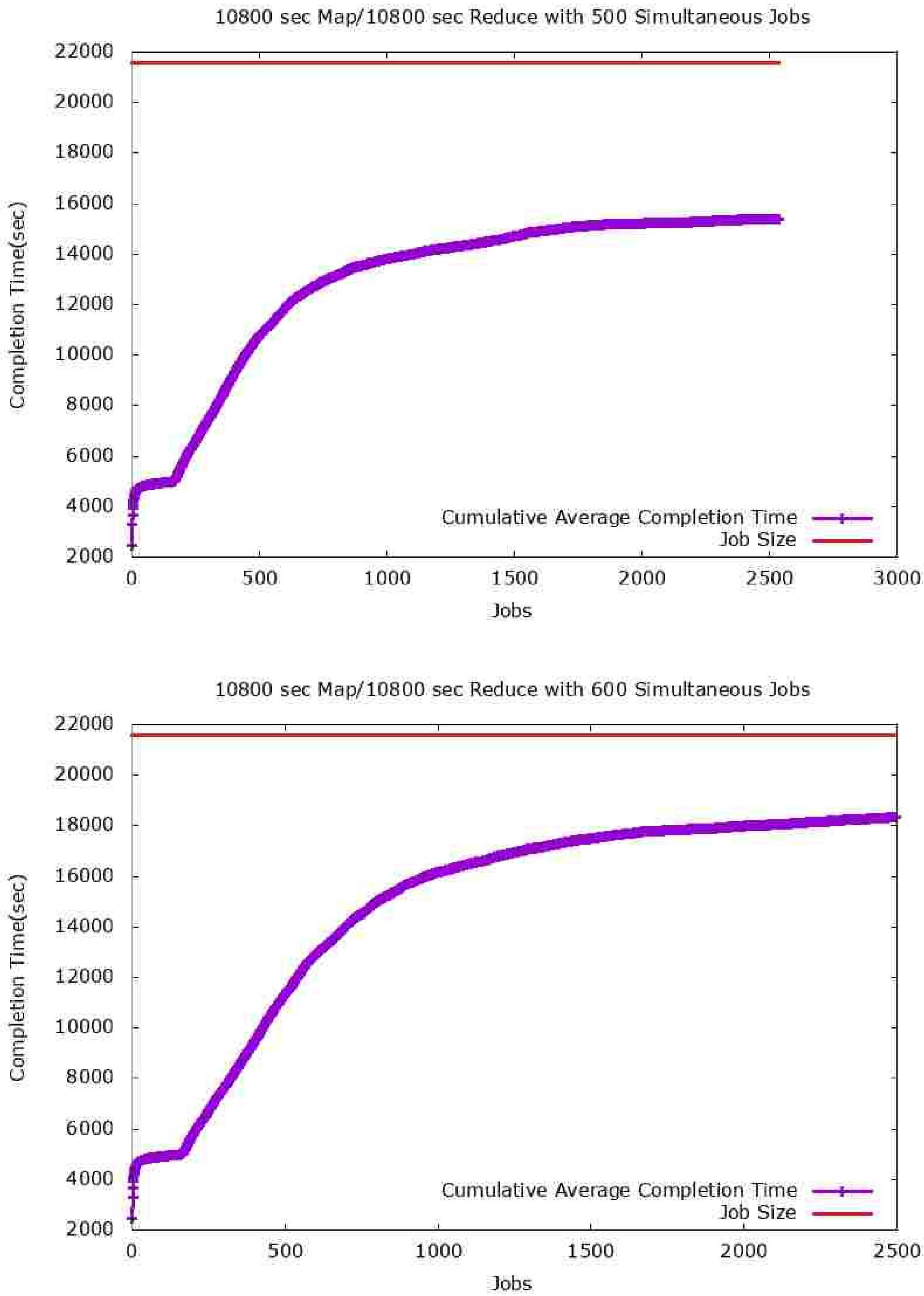
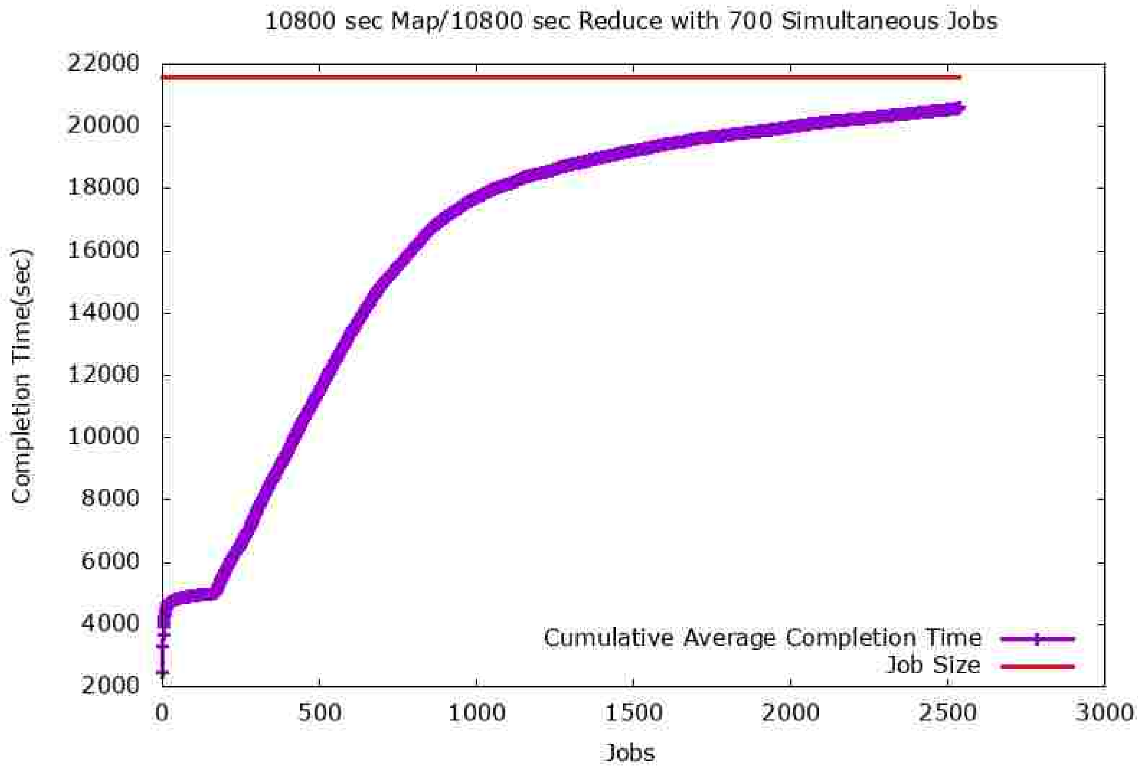


Figure 30: Job Completion Times



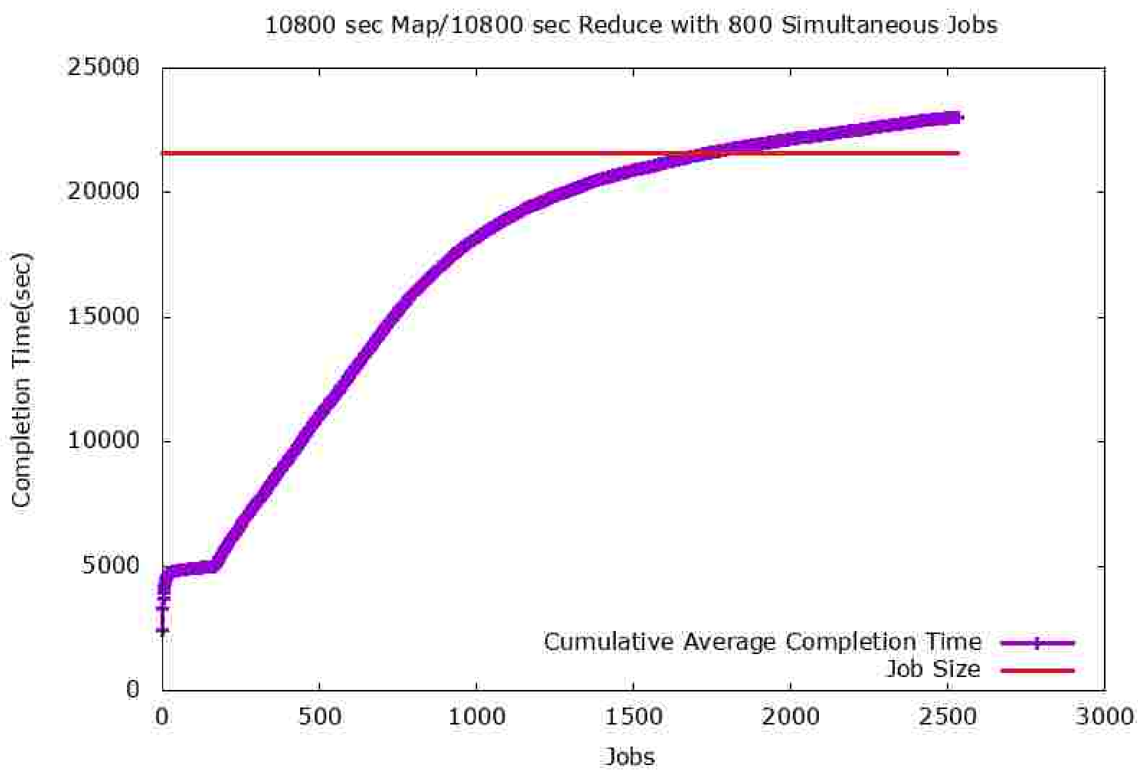


Figure 31: Job Completion Times

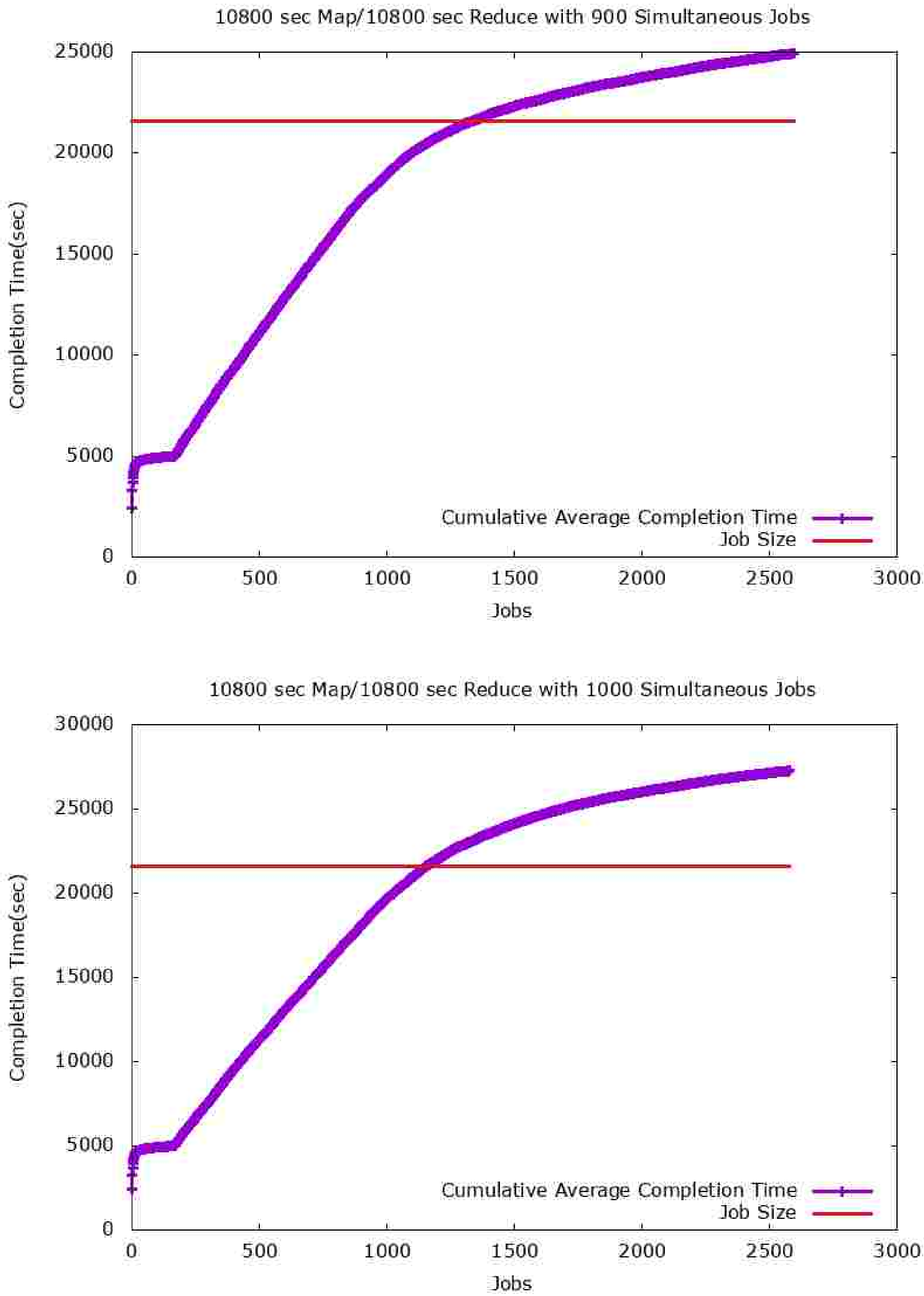
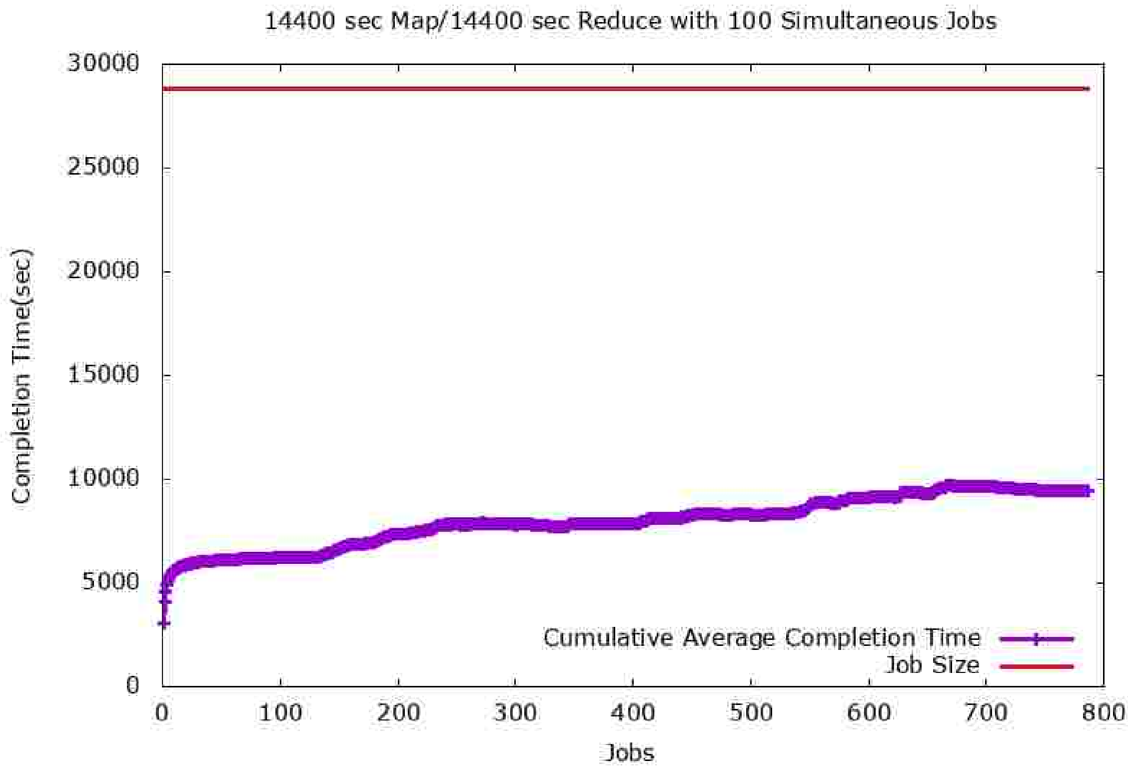


Figure 32: Job Completion Times



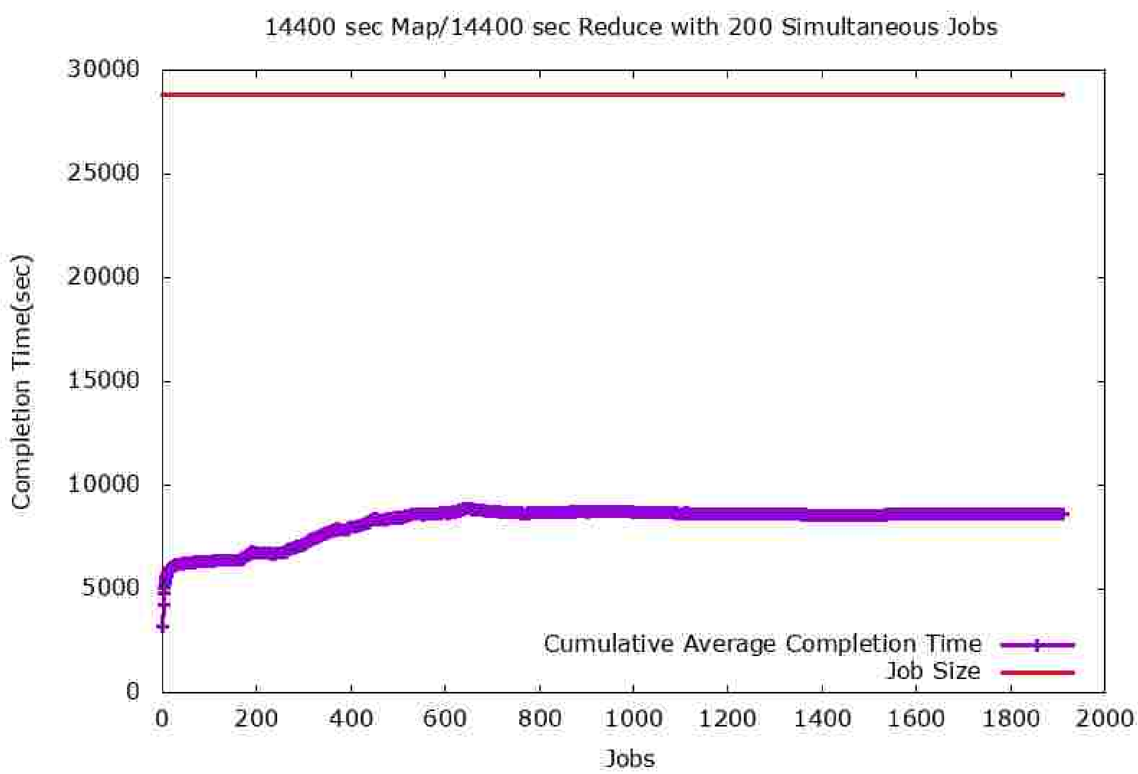


Figure 33: Job Completion Times

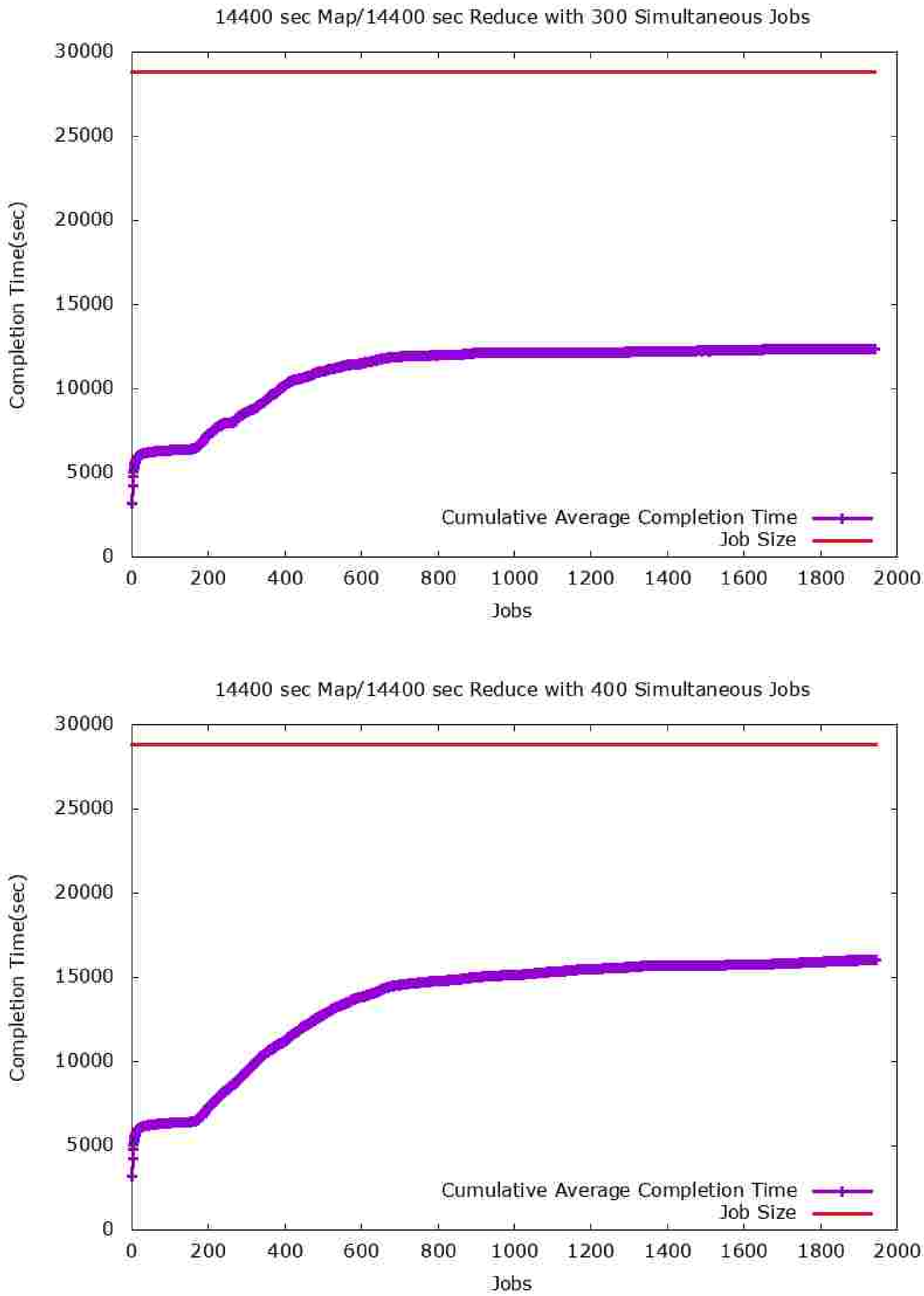
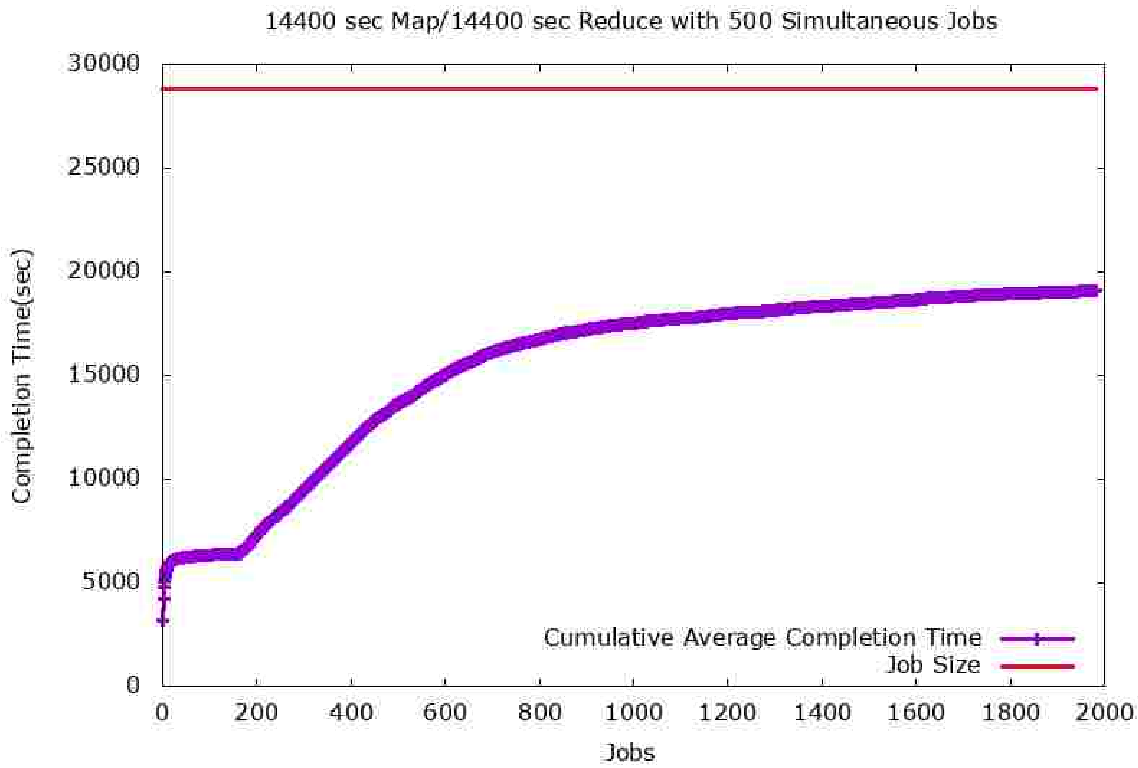


Figure 34: Job Completion Times



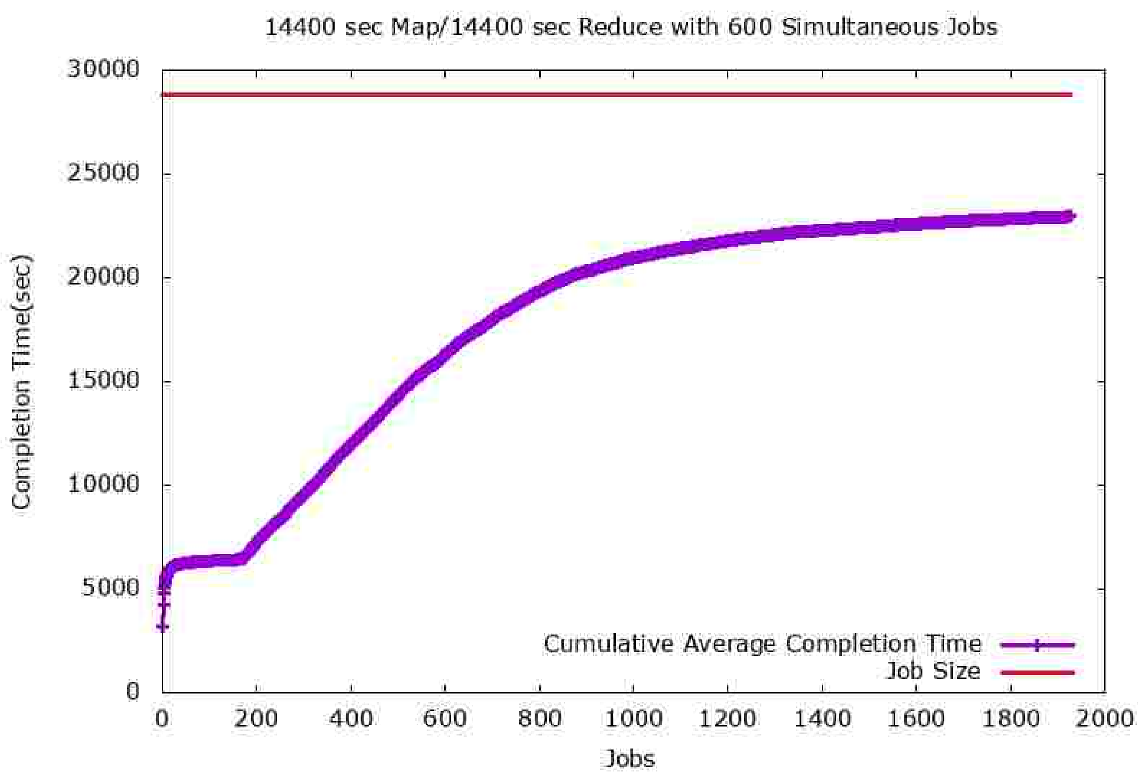
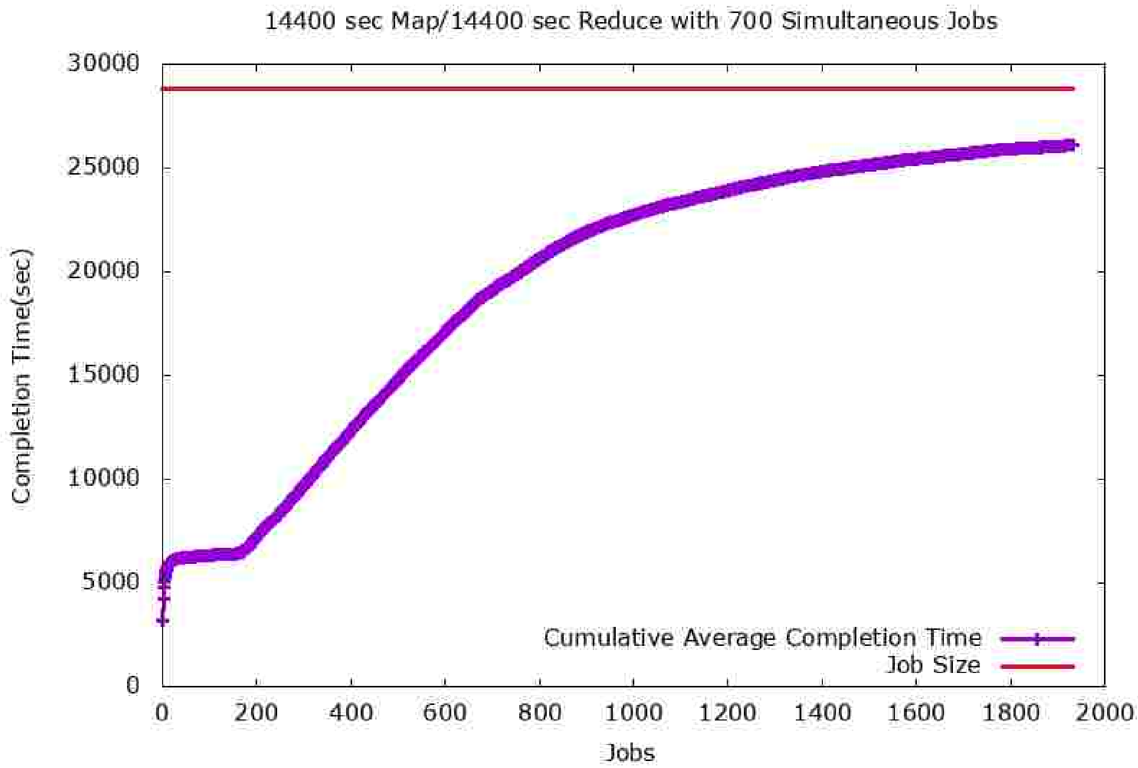


Figure 35: Job Completion Times



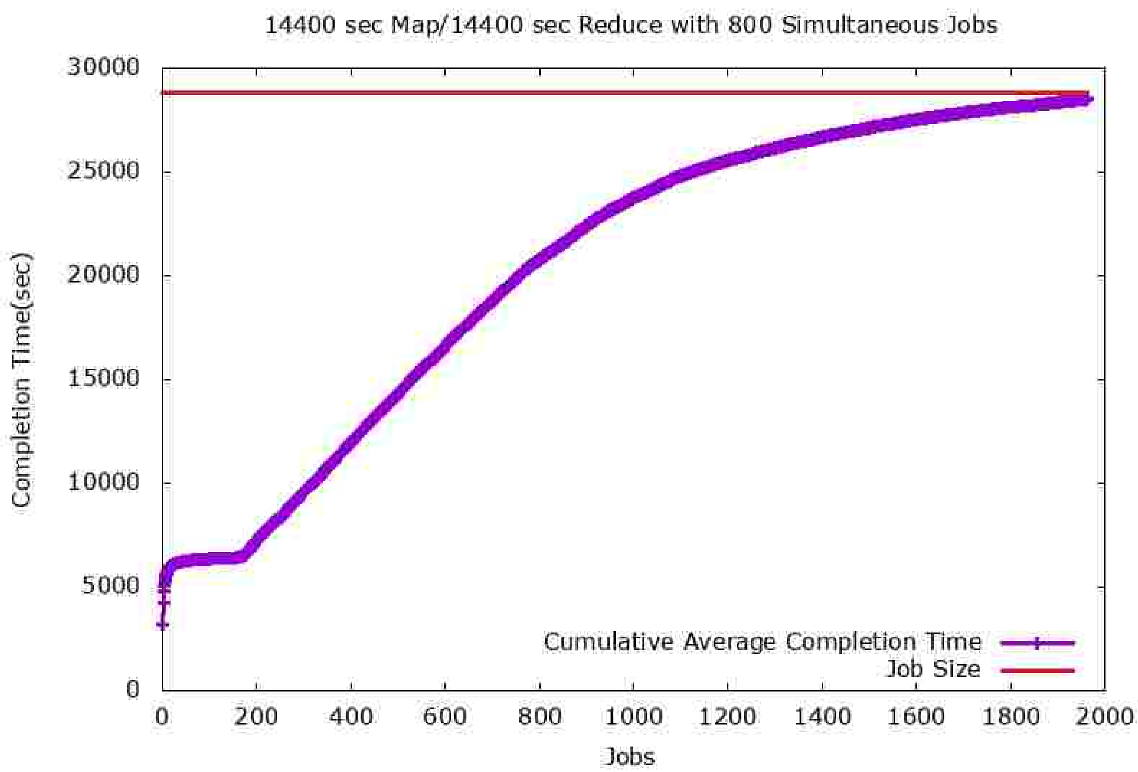
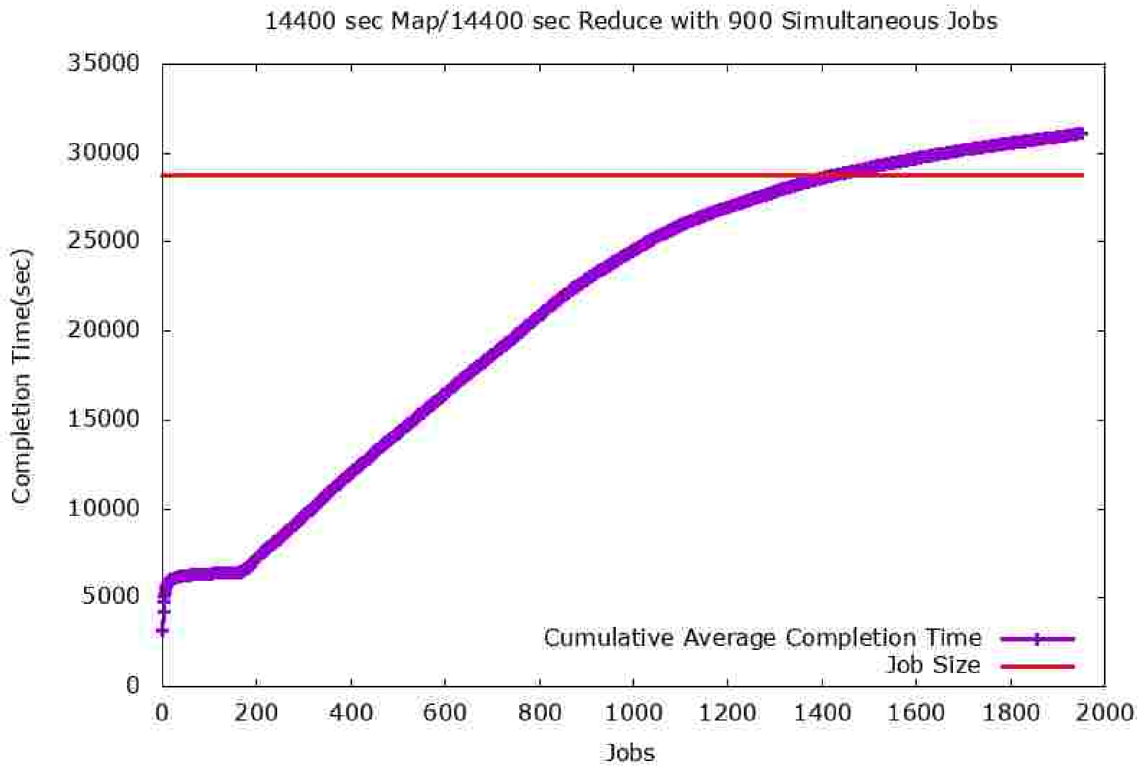


Figure 36: Job Completion Times



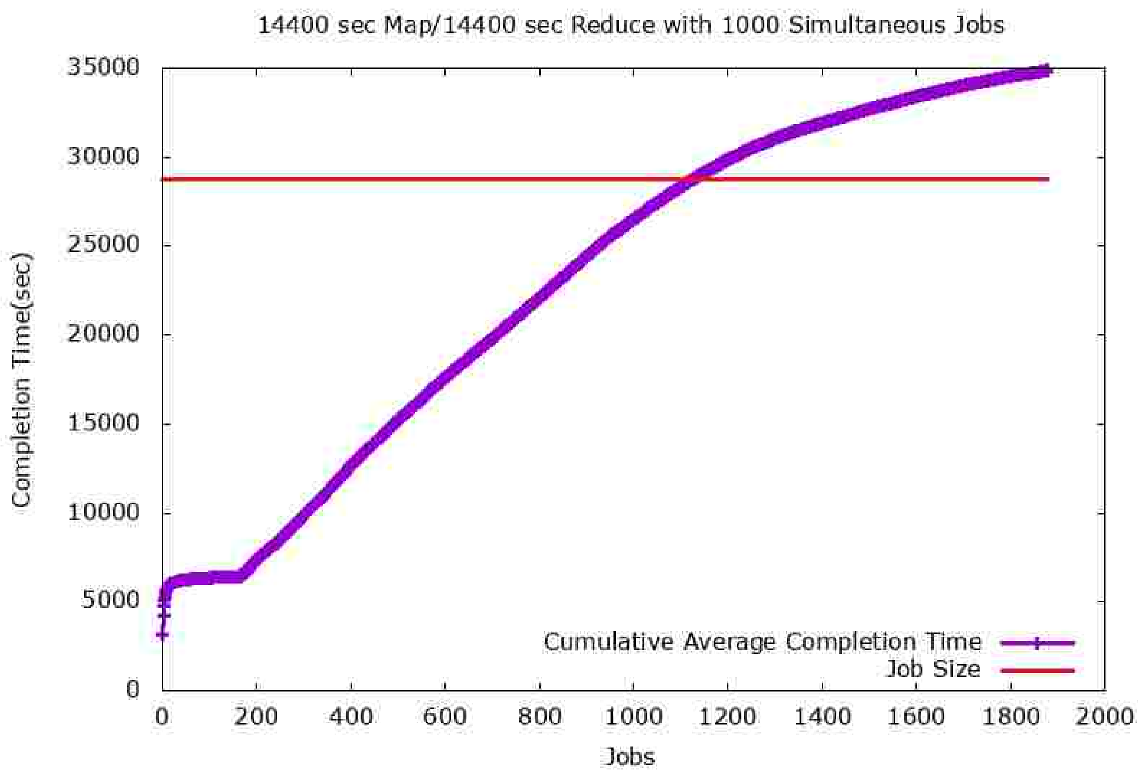
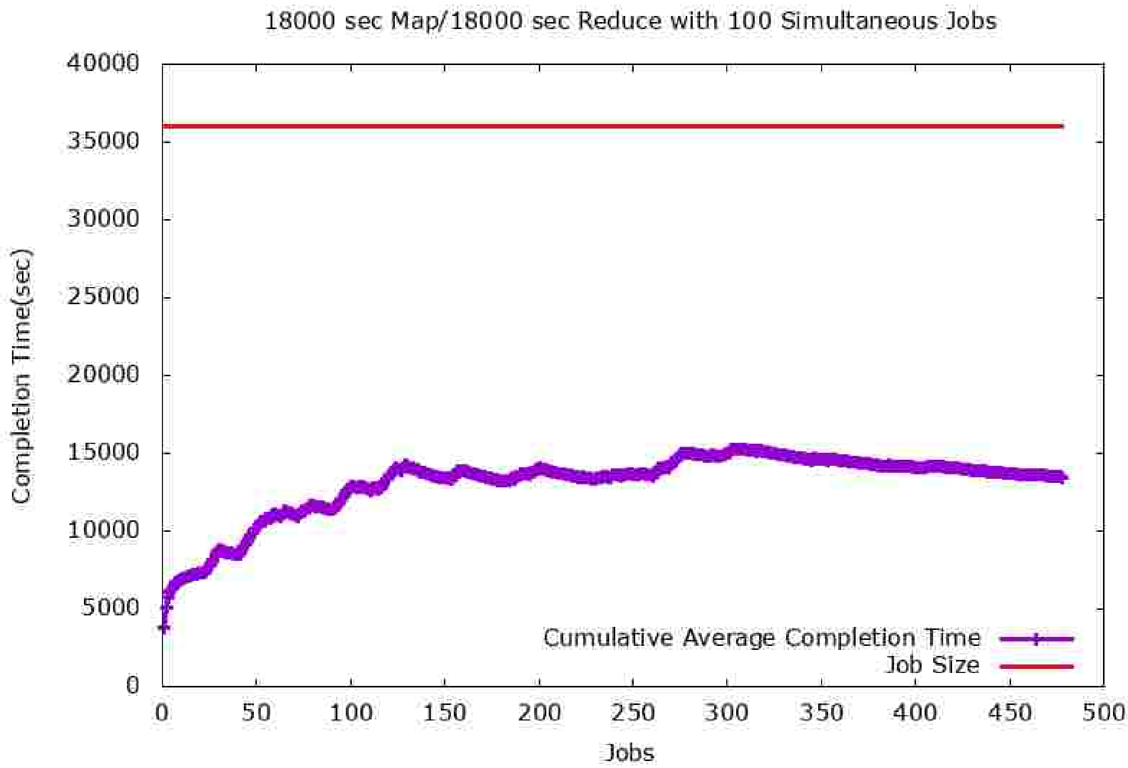


Figure 37: Job Completion Times



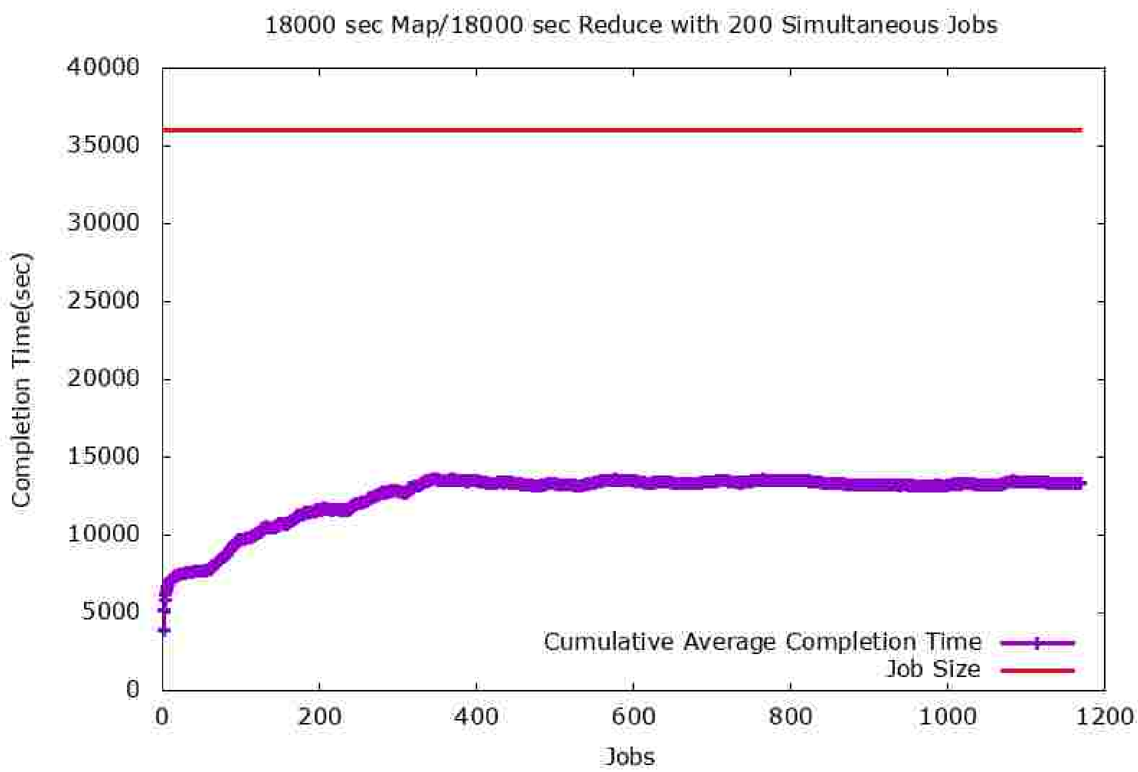
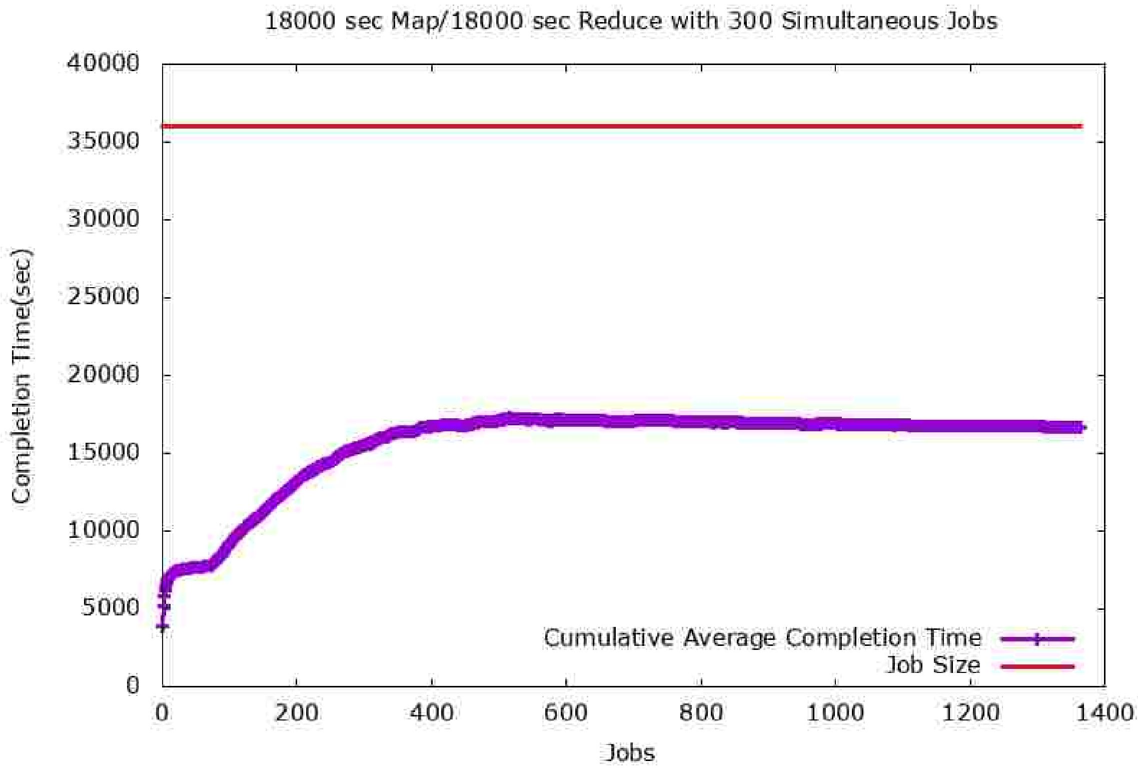


Figure 38: Job Completion Times



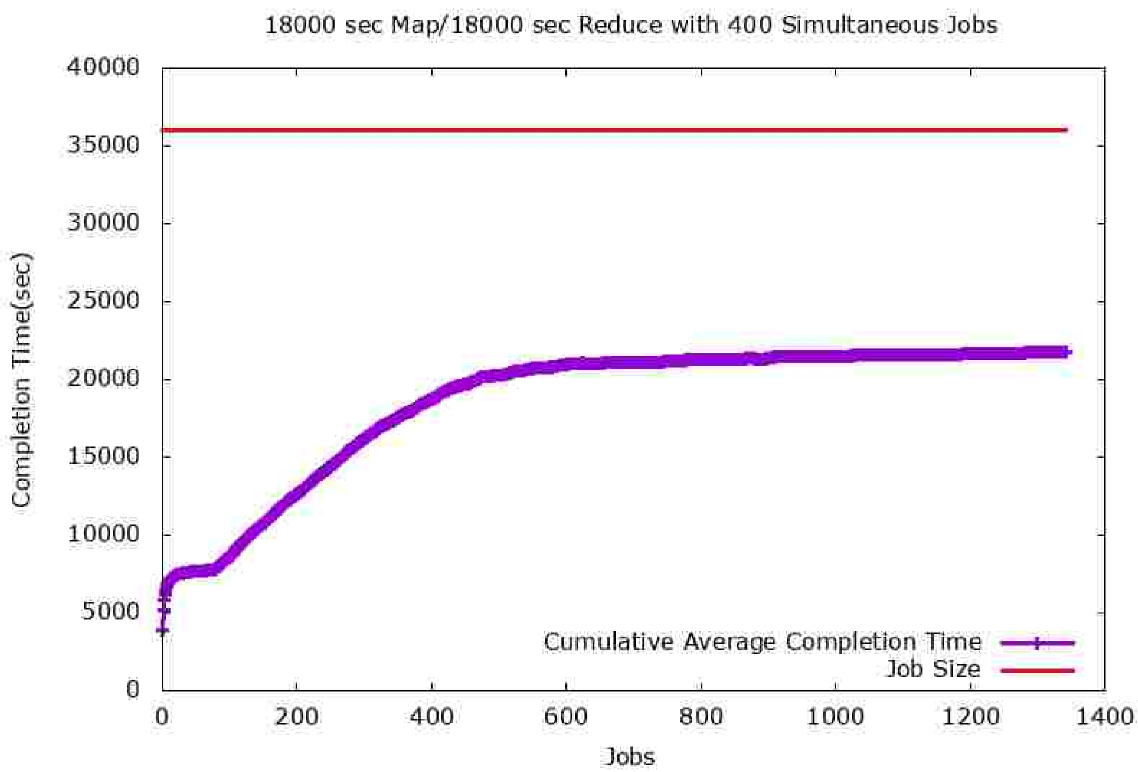
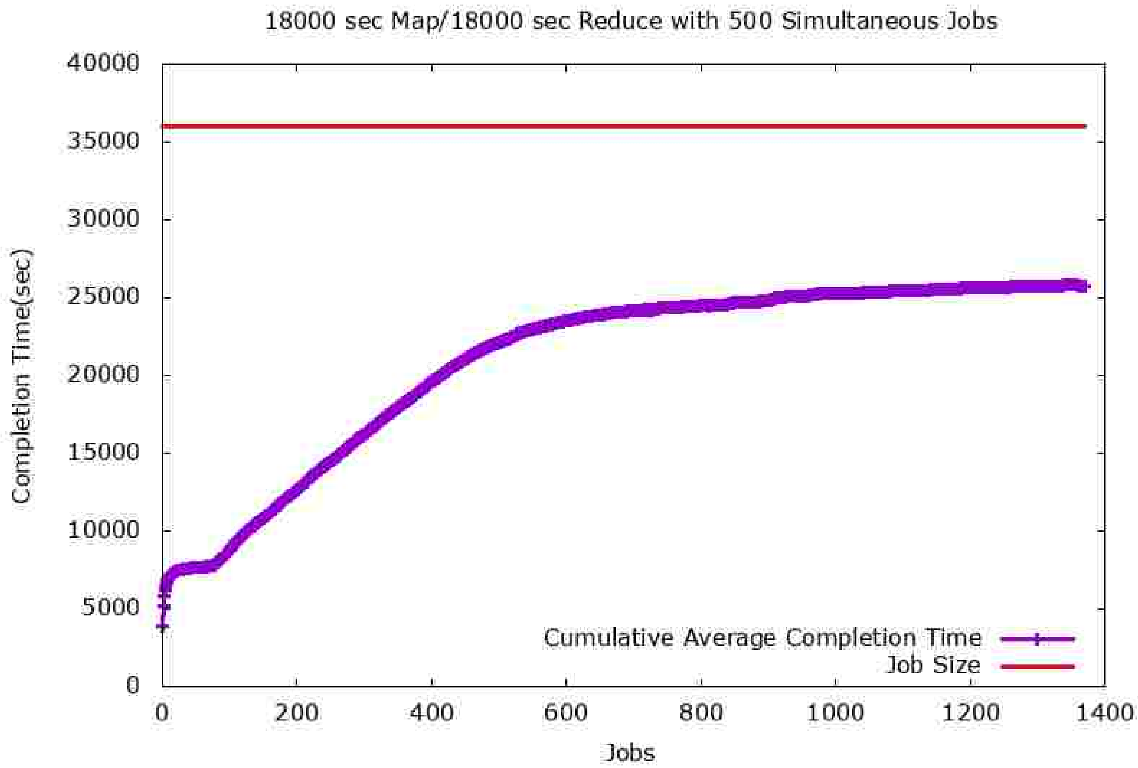


Figure 39: Job Completion Times



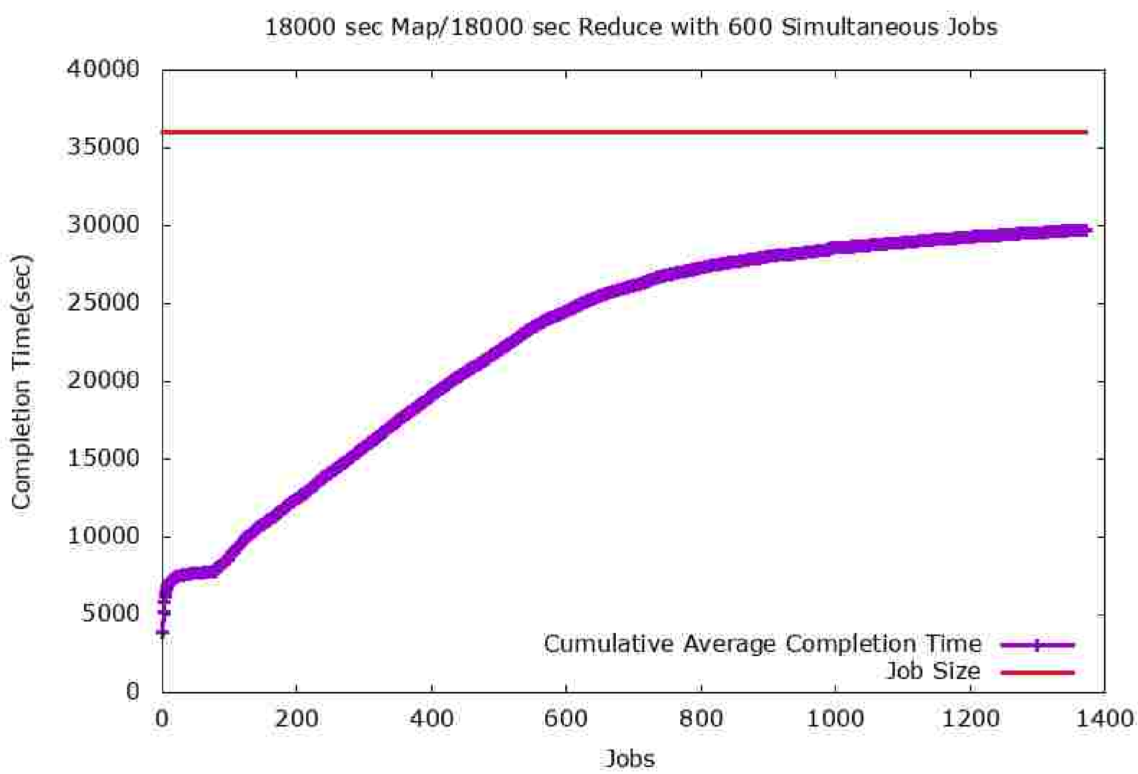


Figure 40: Job Completion Times

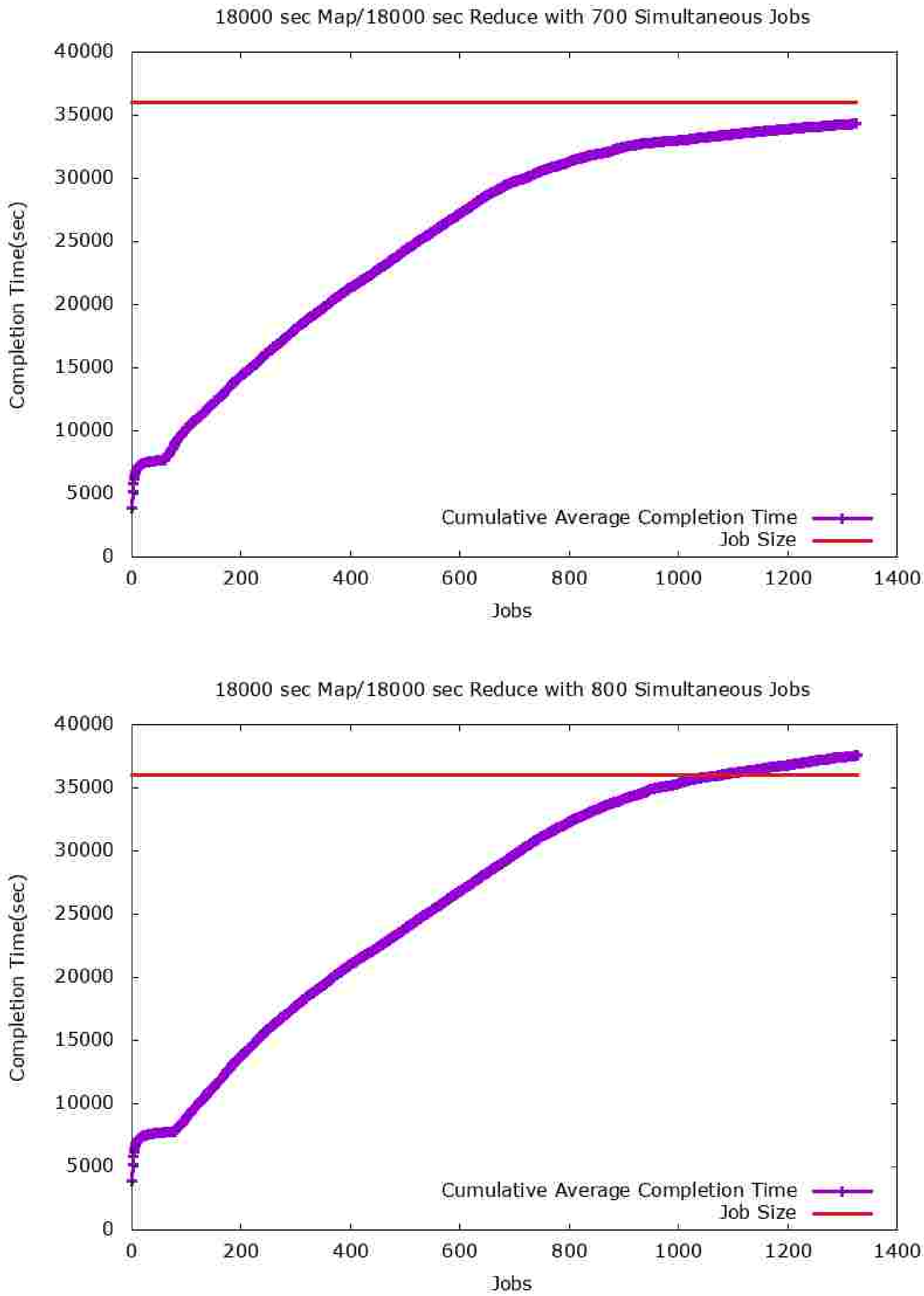
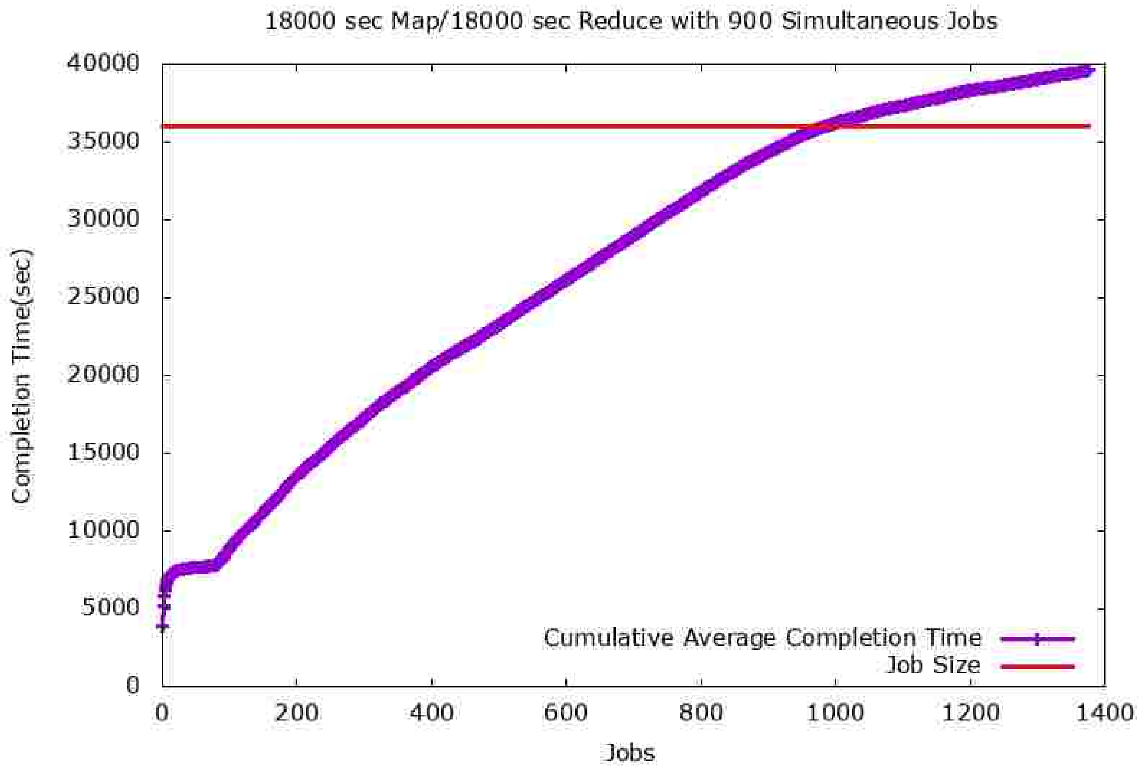


Figure 41: Job Completion Times



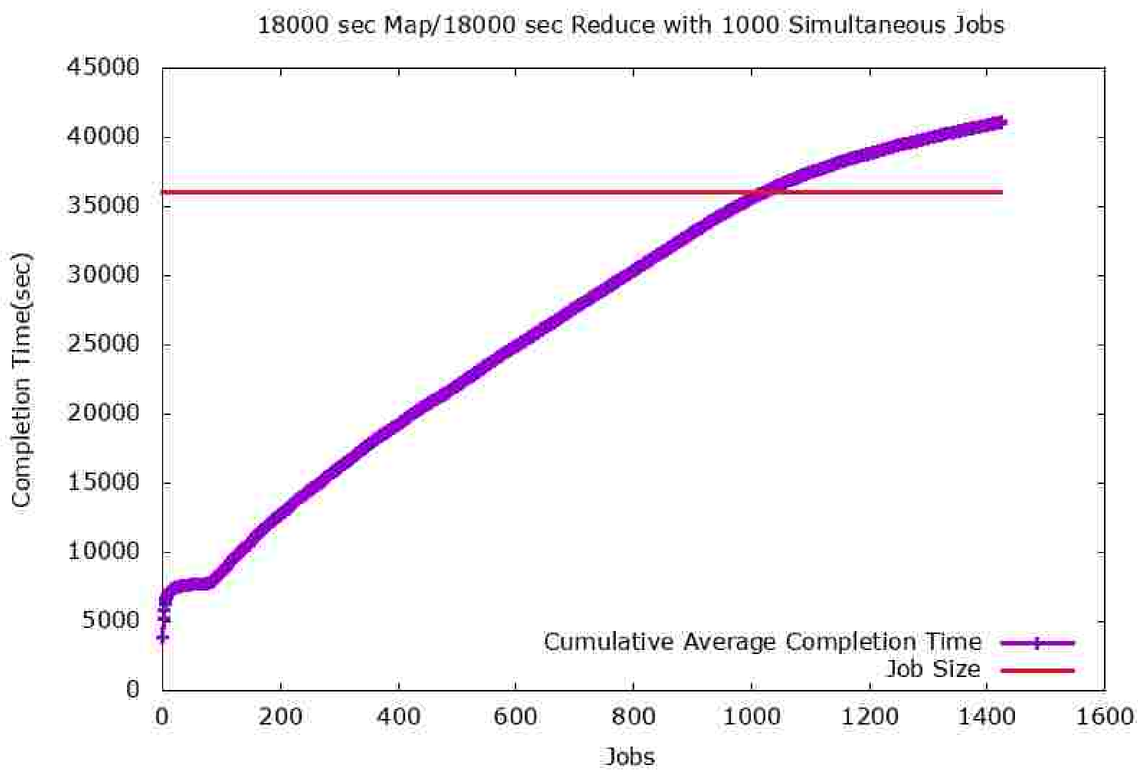
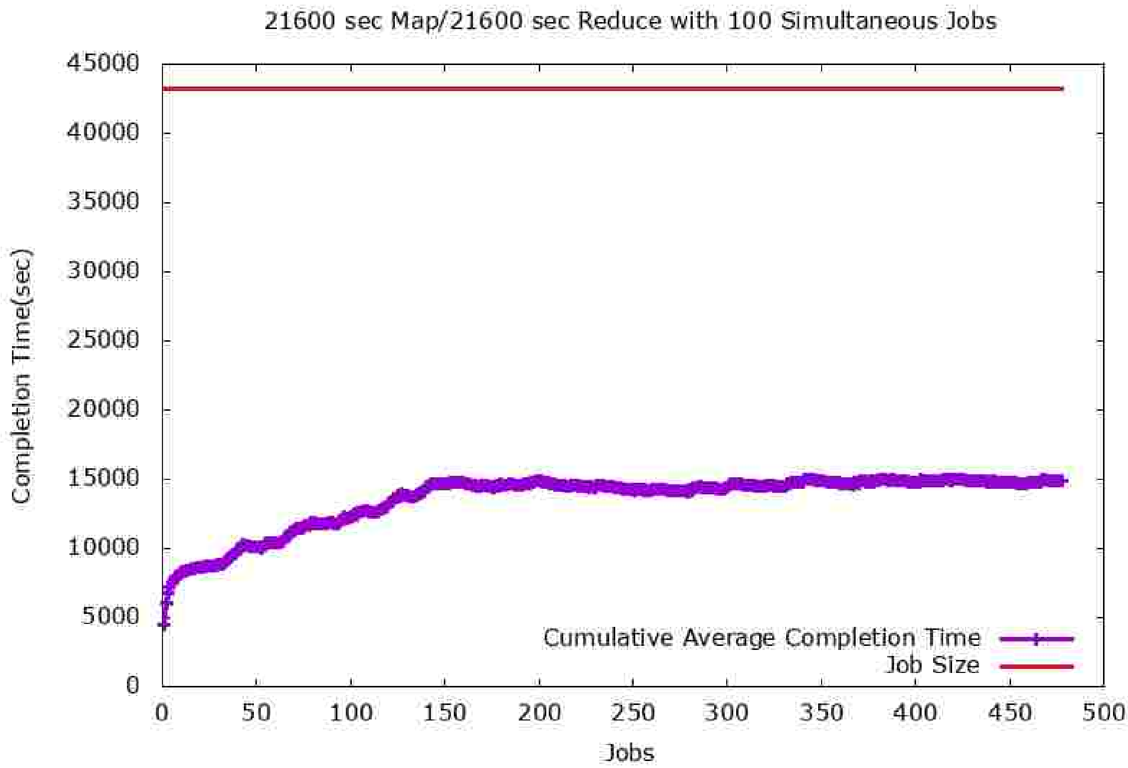


Figure 42: Job Completion Times



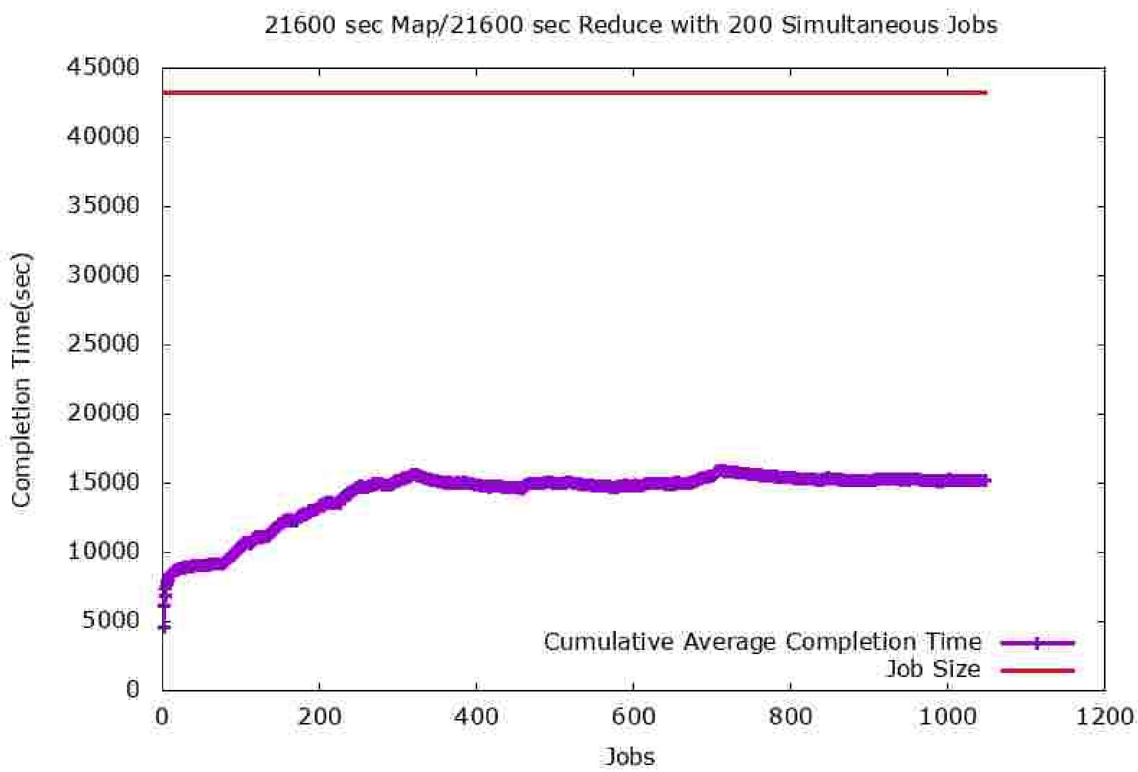
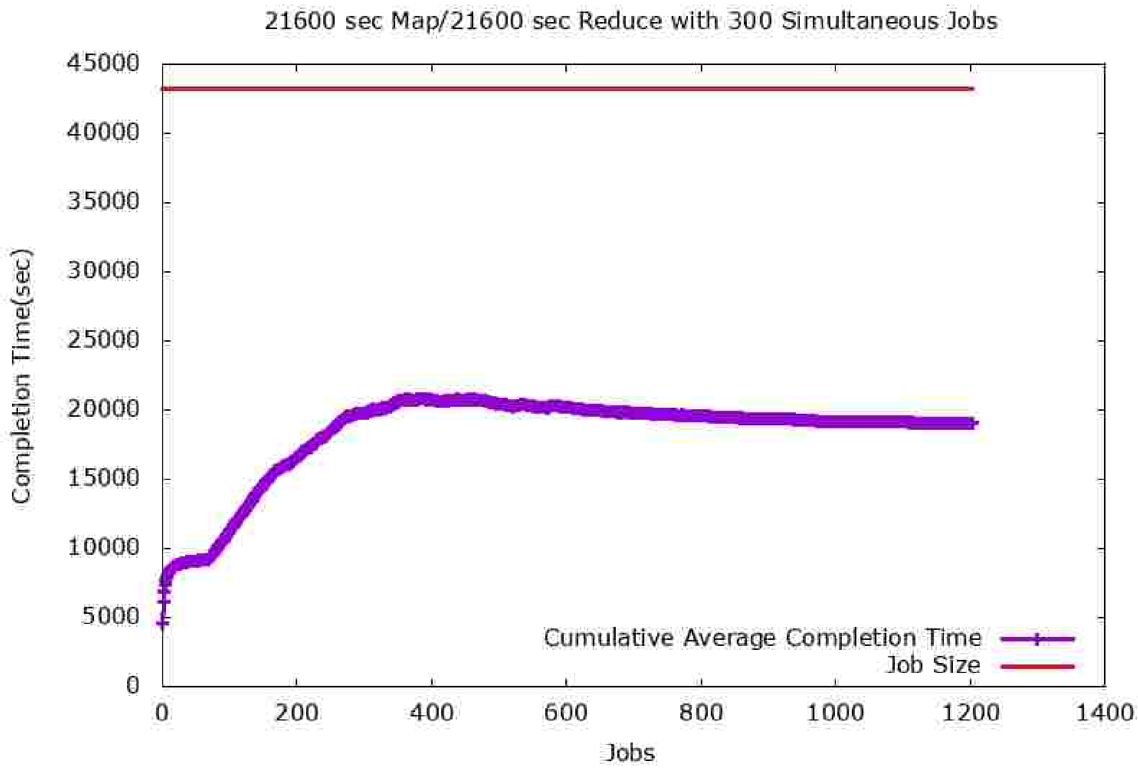


Figure 43: Job Completion Times



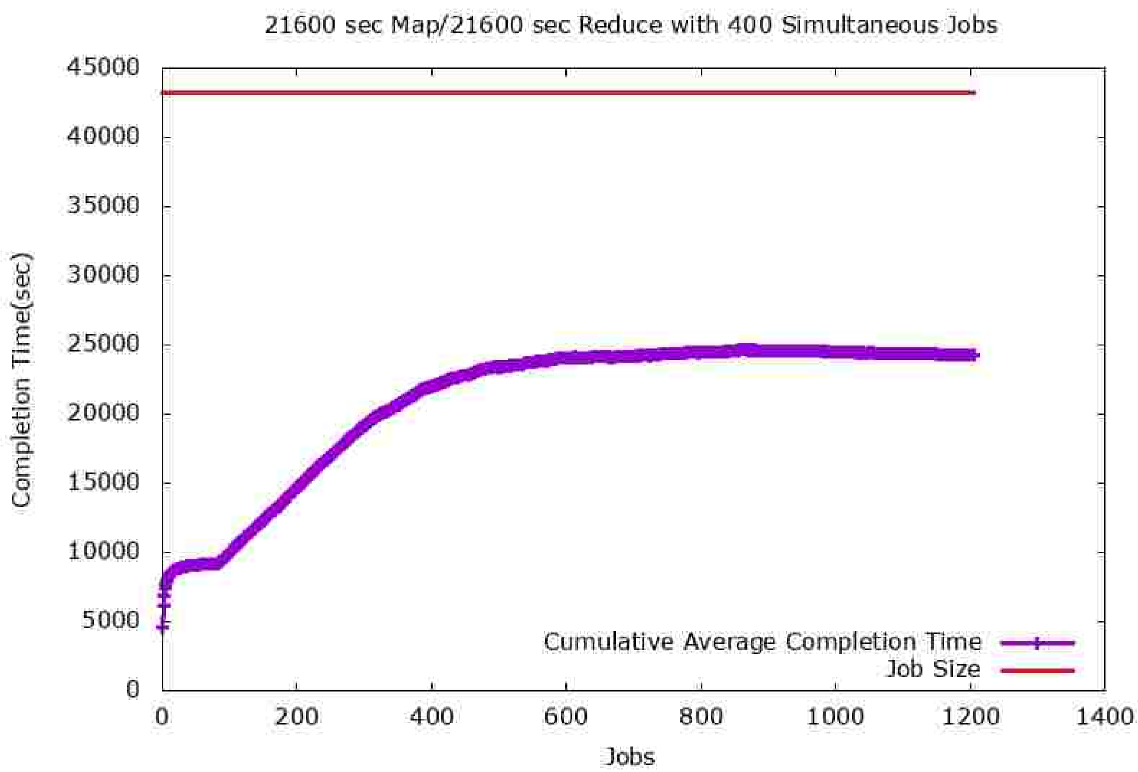
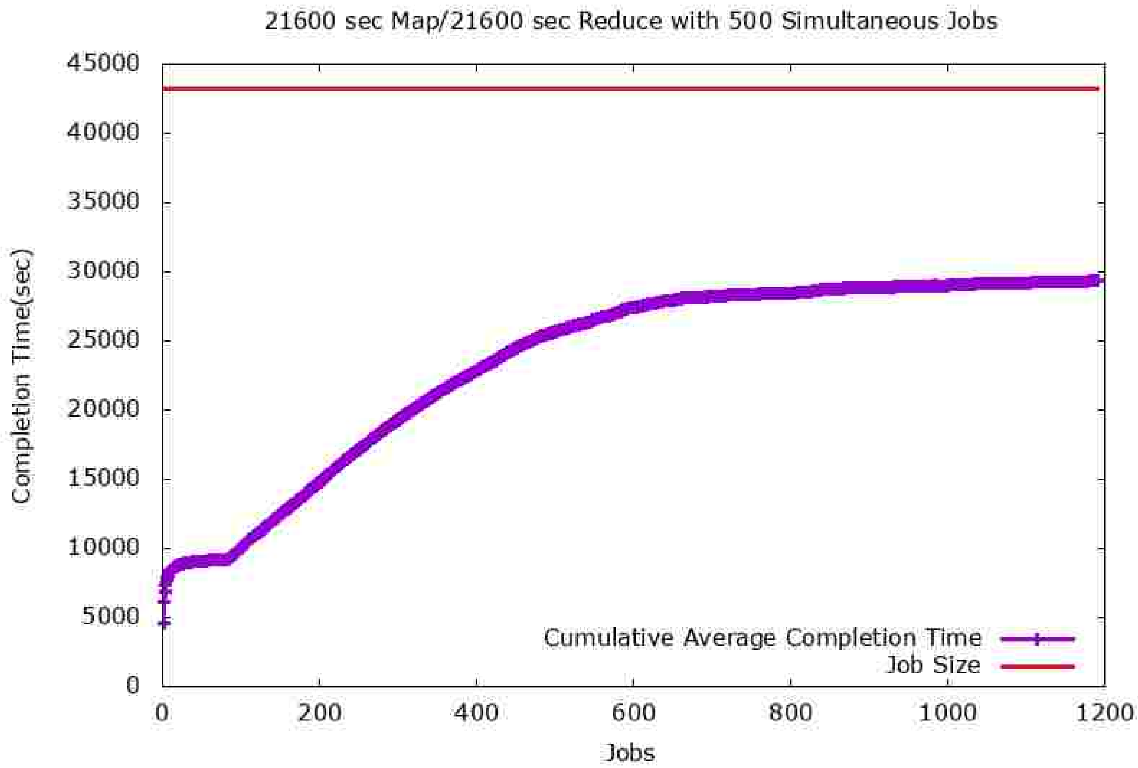


Figure 44: Job Completion Times



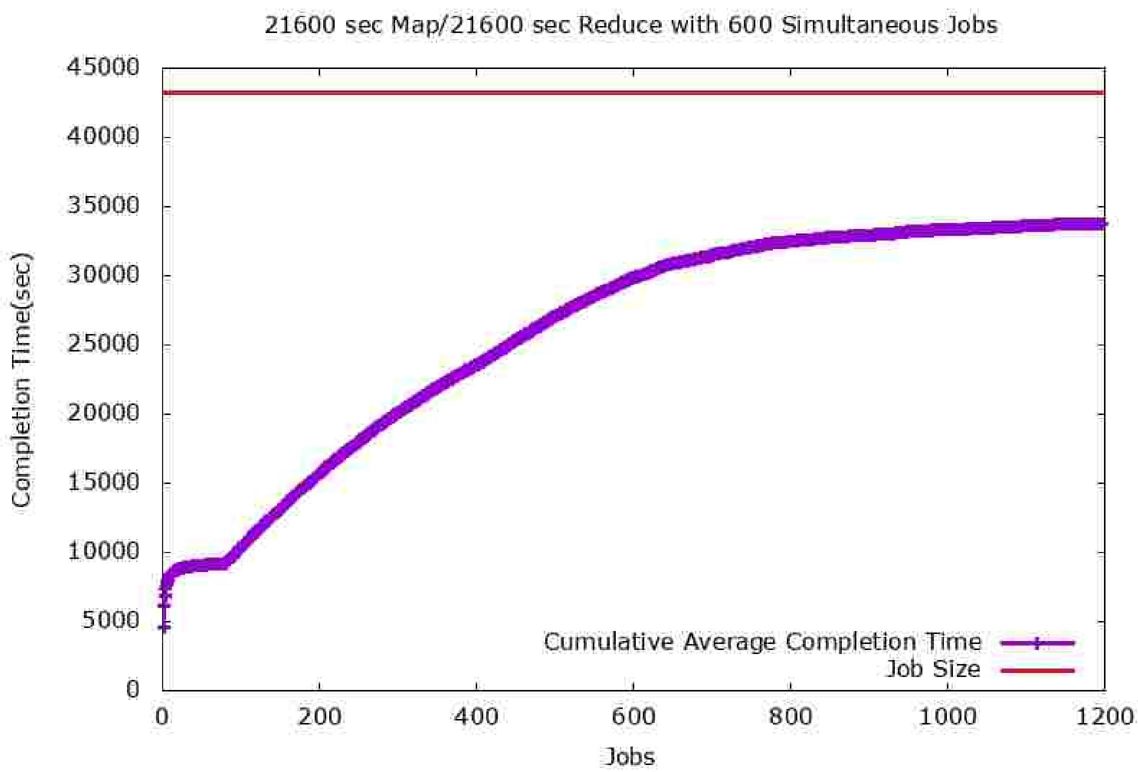


Figure 45: Job Completion Times

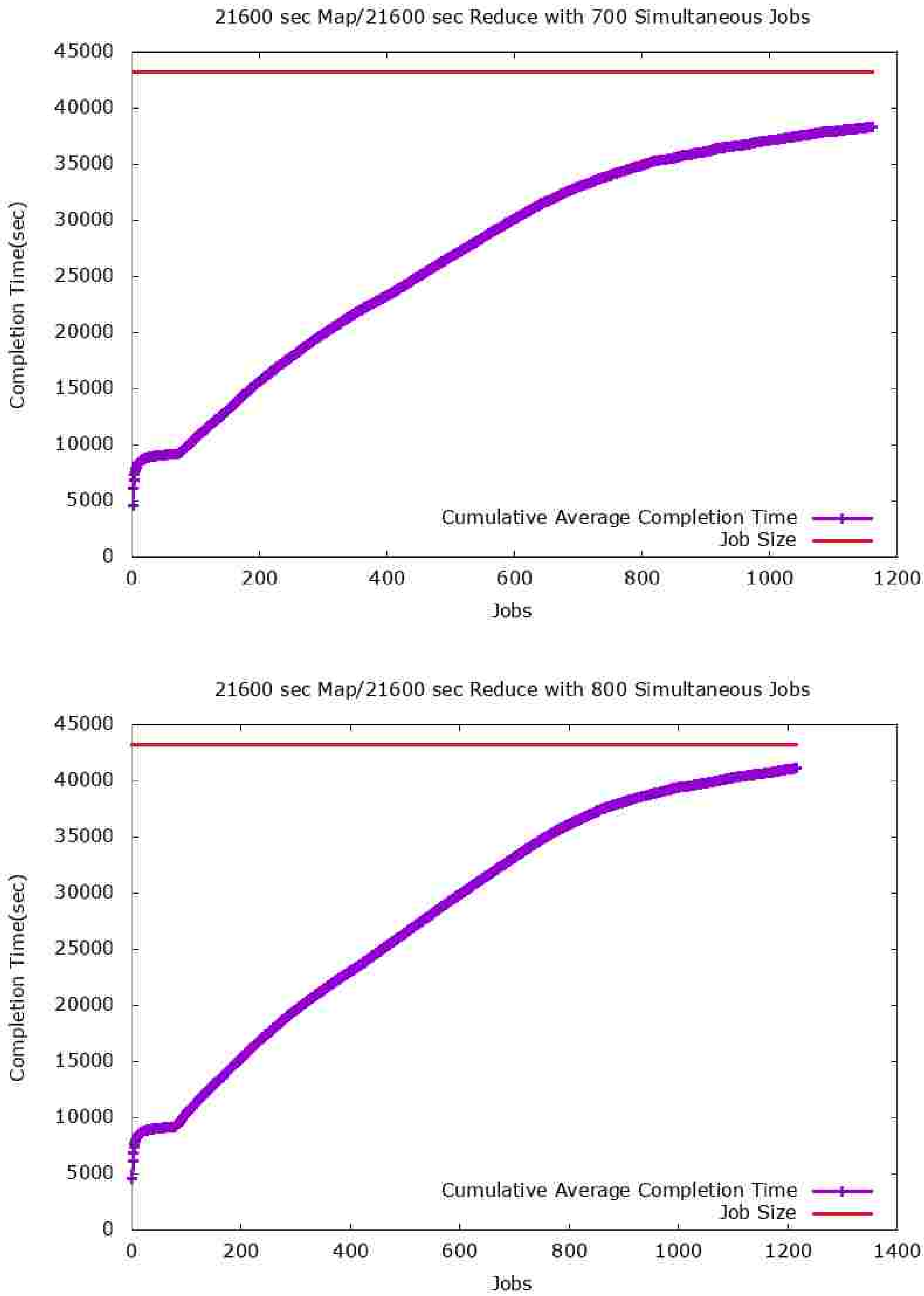
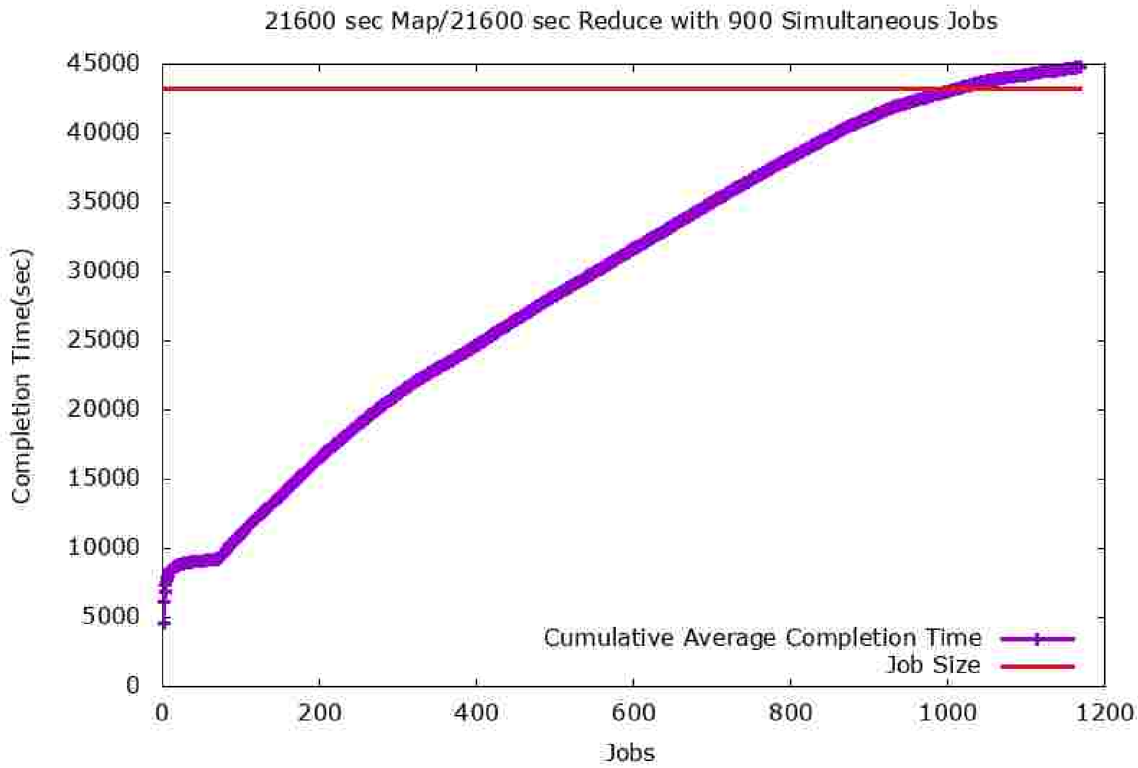


Figure 46: Job Completion Times



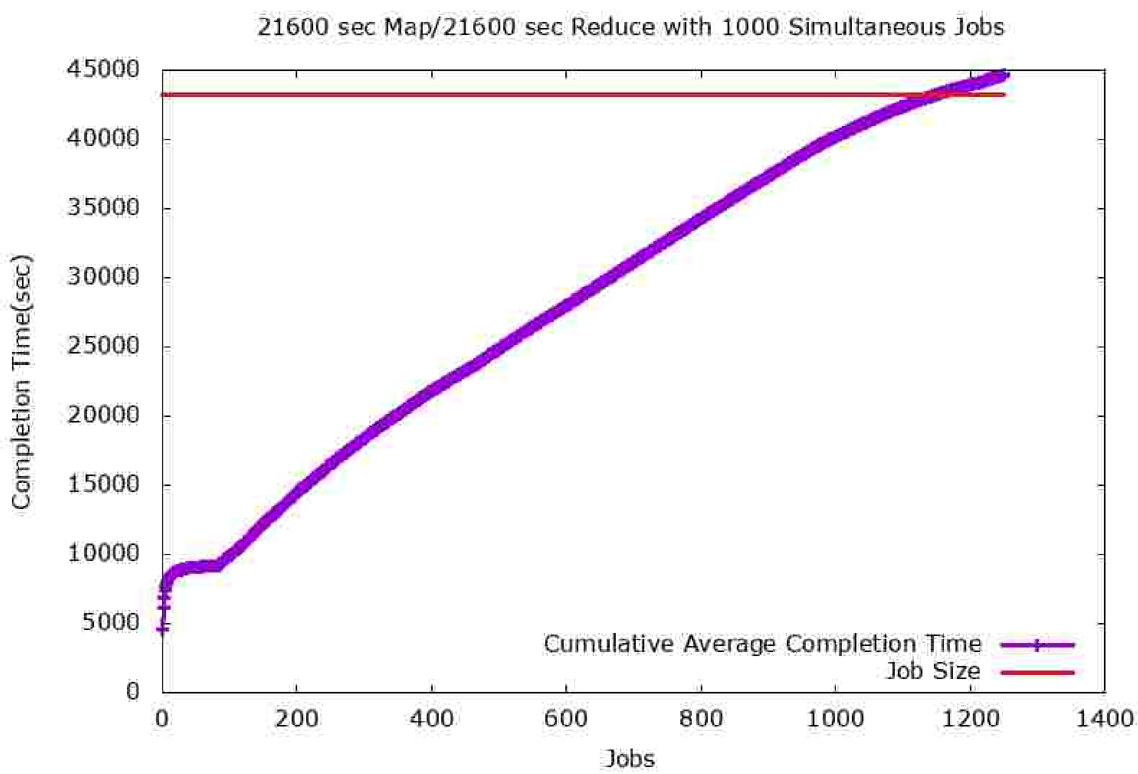
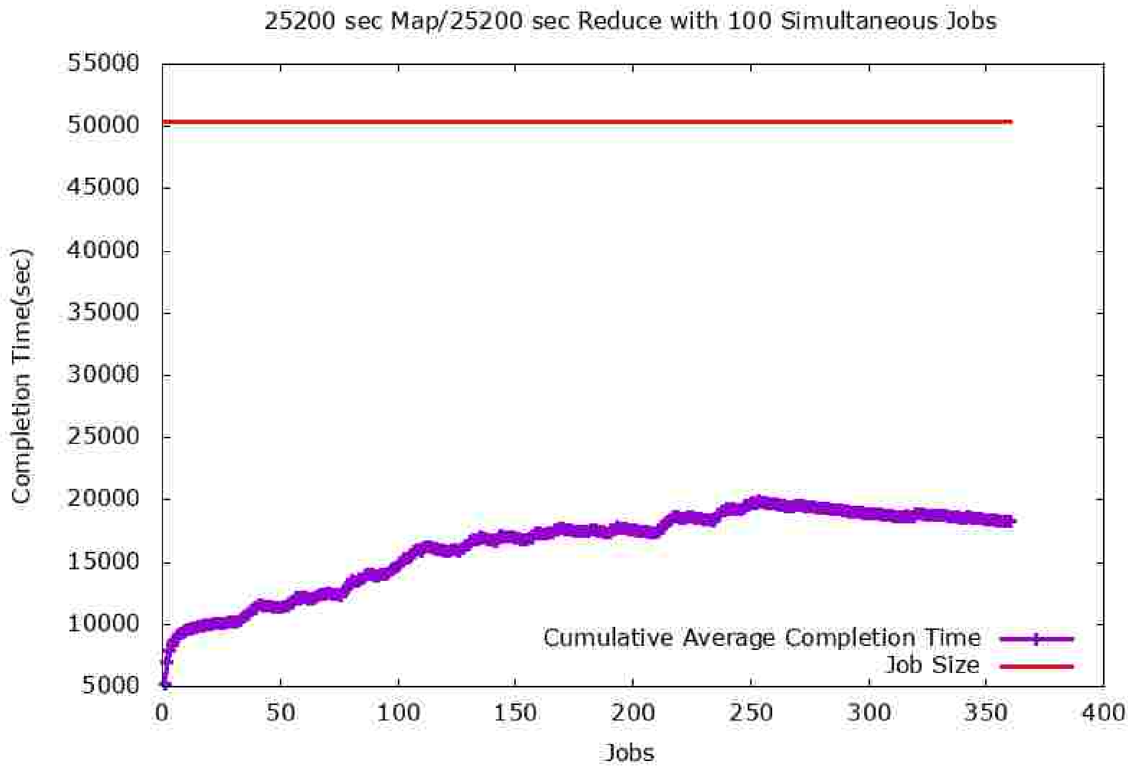


Figure 47: Job Completion Times



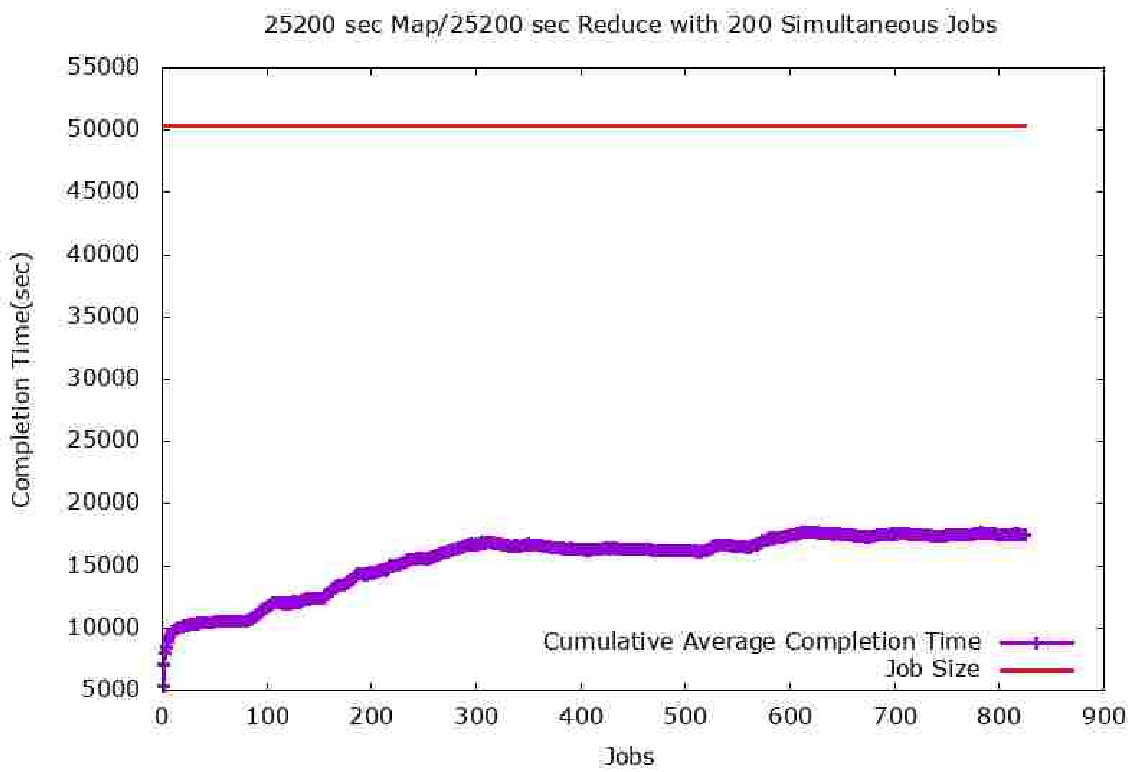
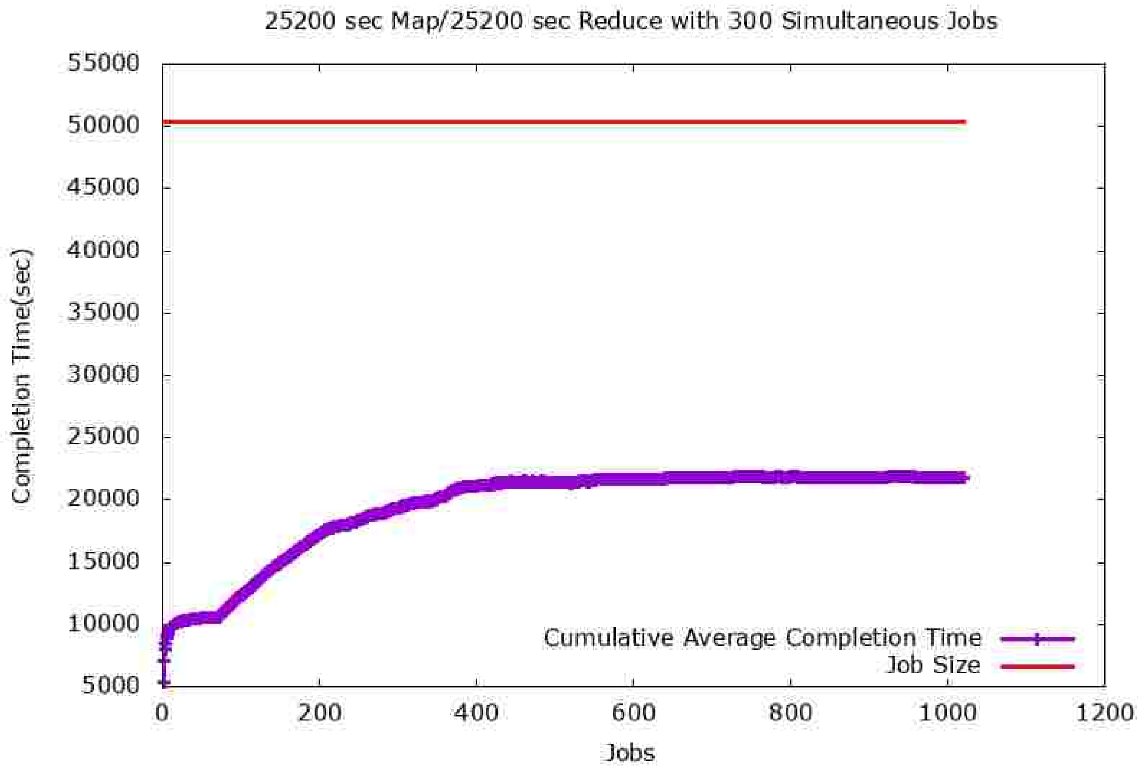


Figure 48: Job Completion Times



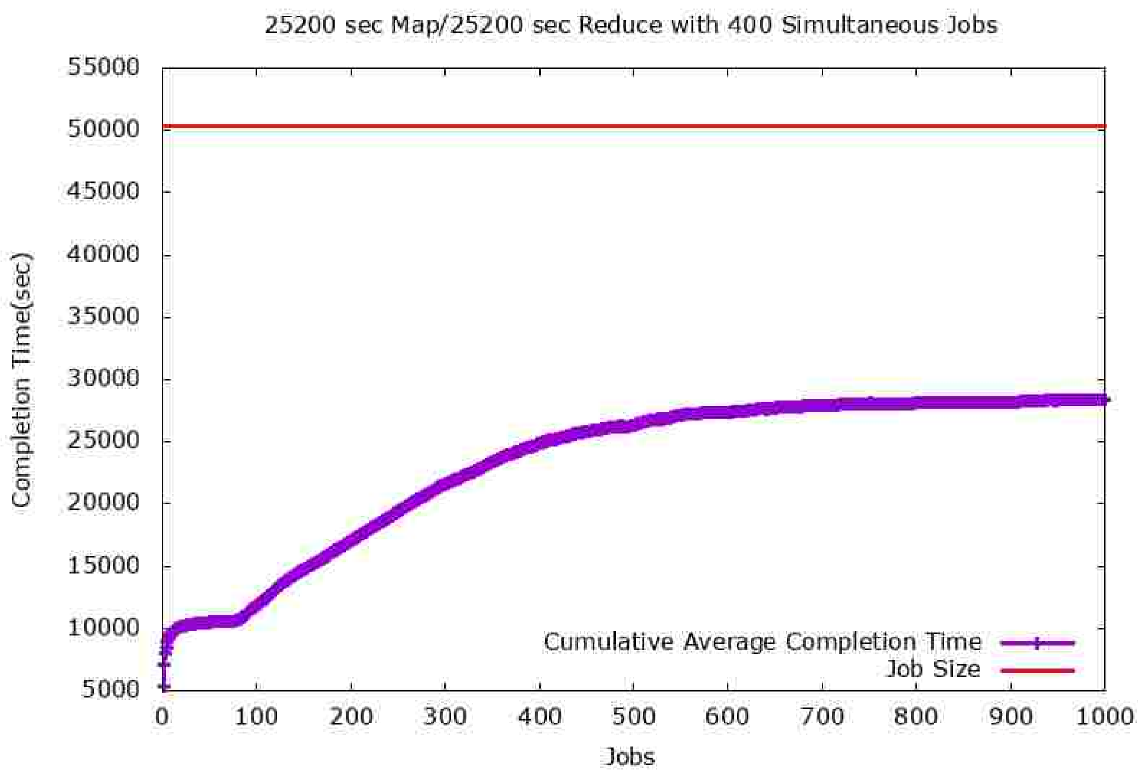
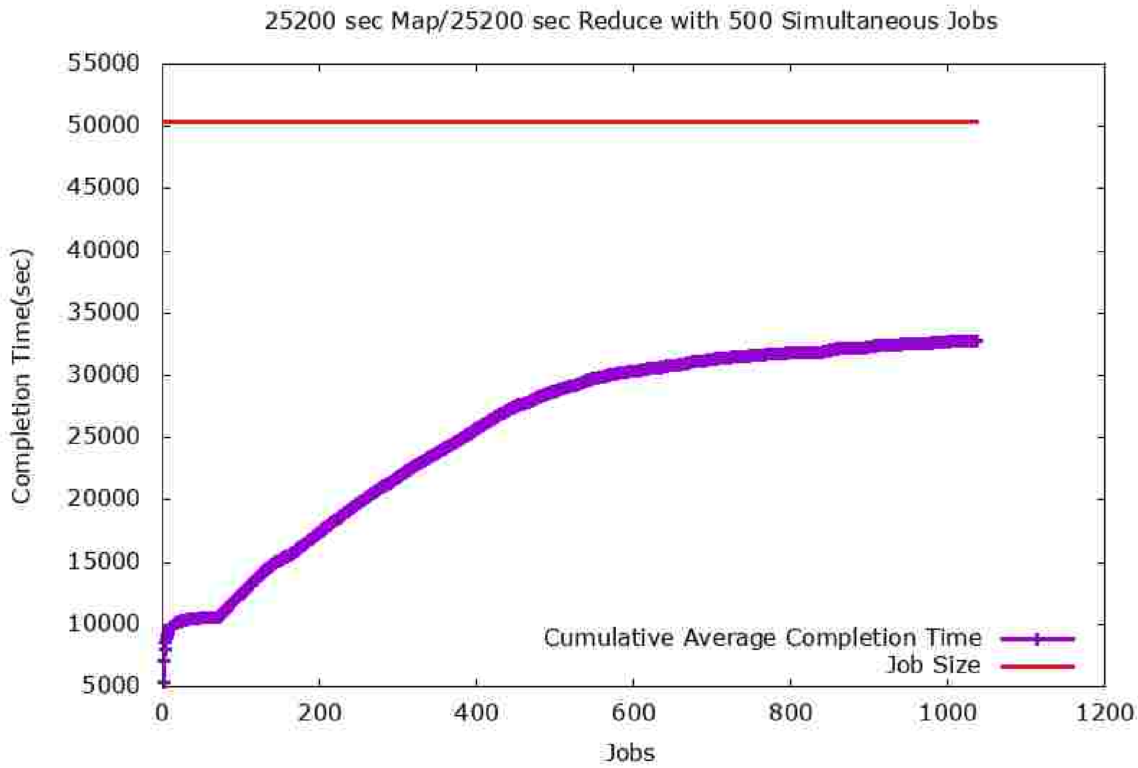


Figure 49: Job Completion Times



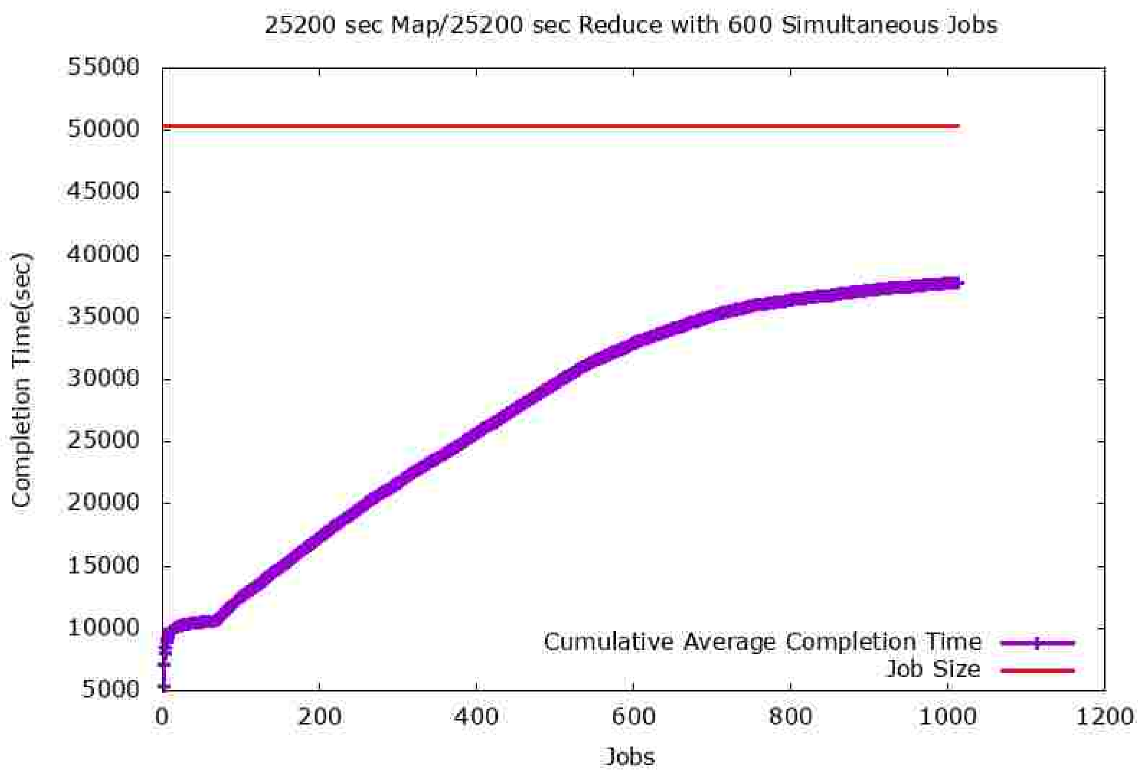
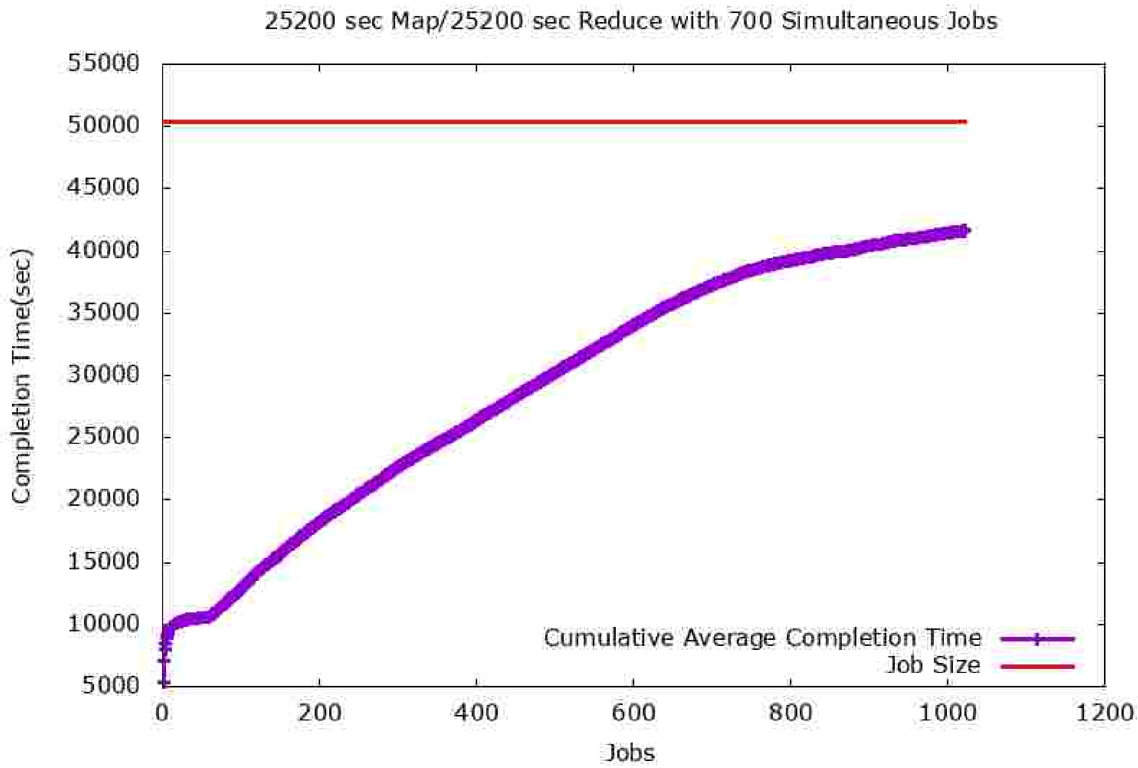


Figure 50: Job Completion Times



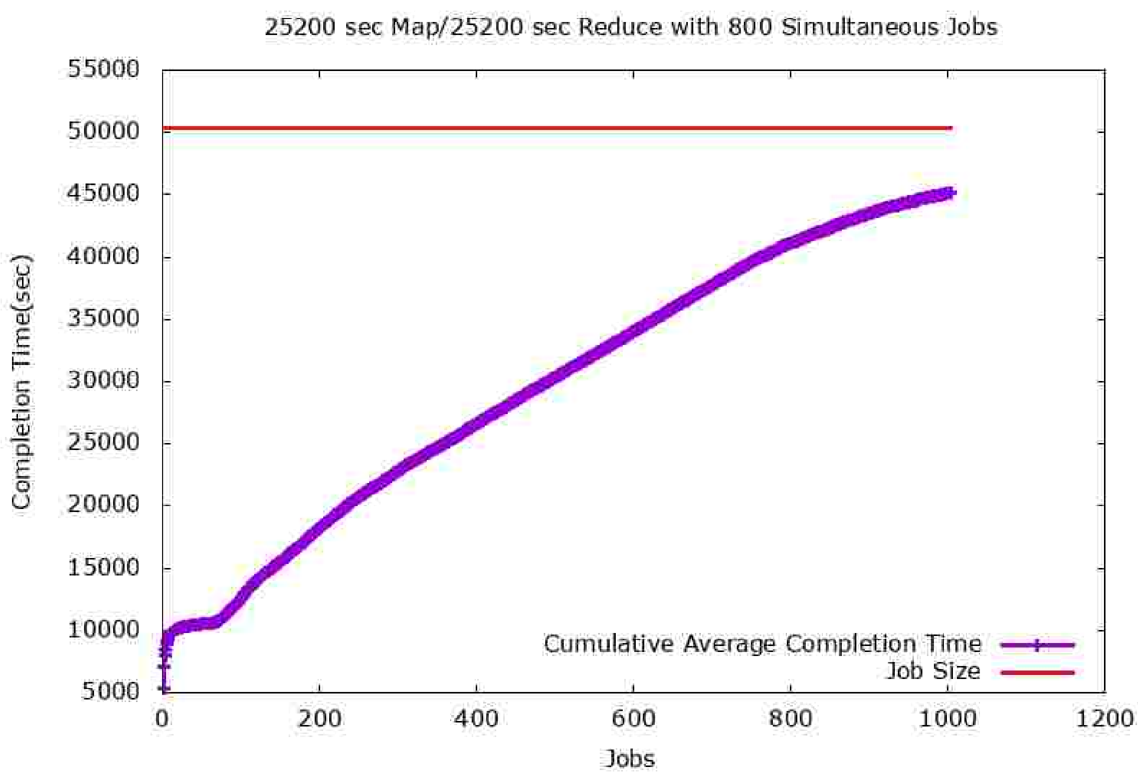
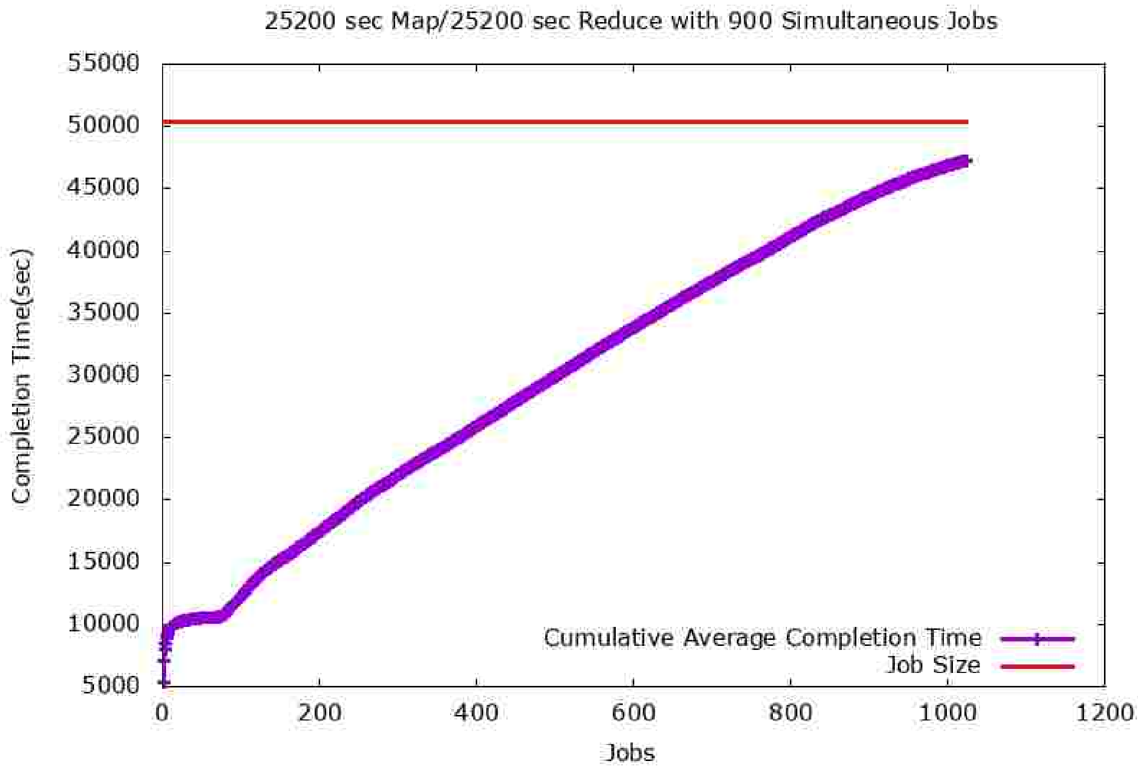


Figure 51: Job Completion Times



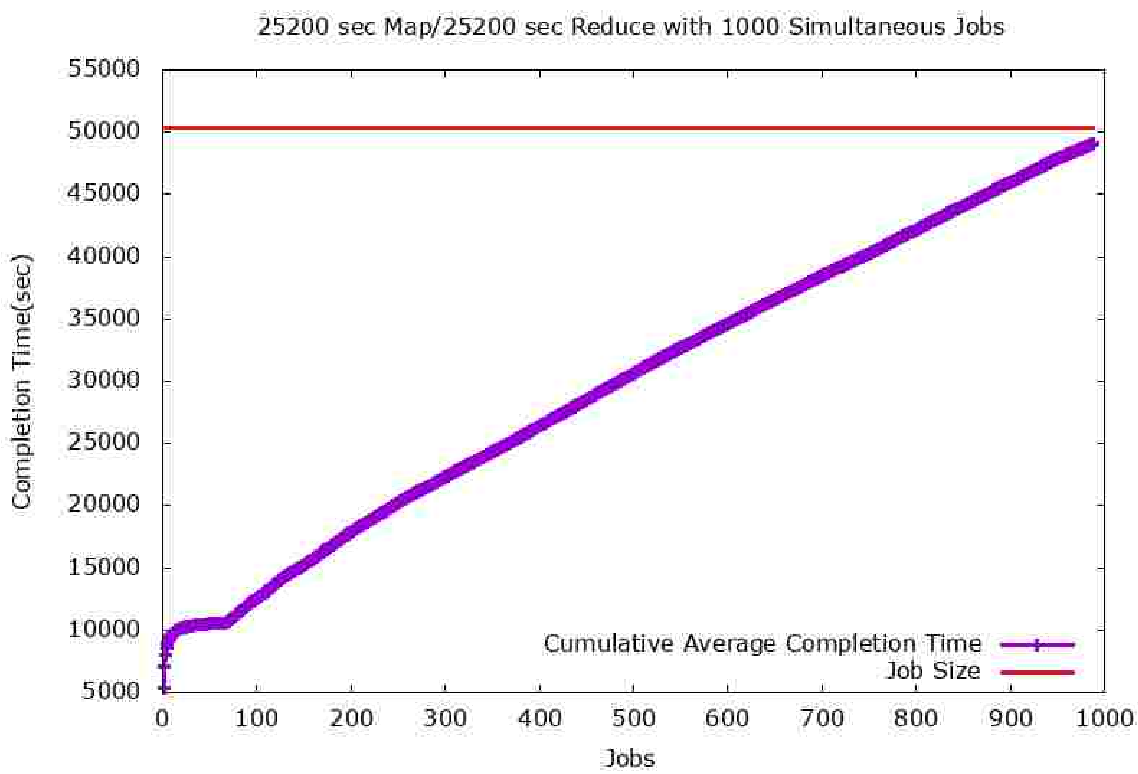
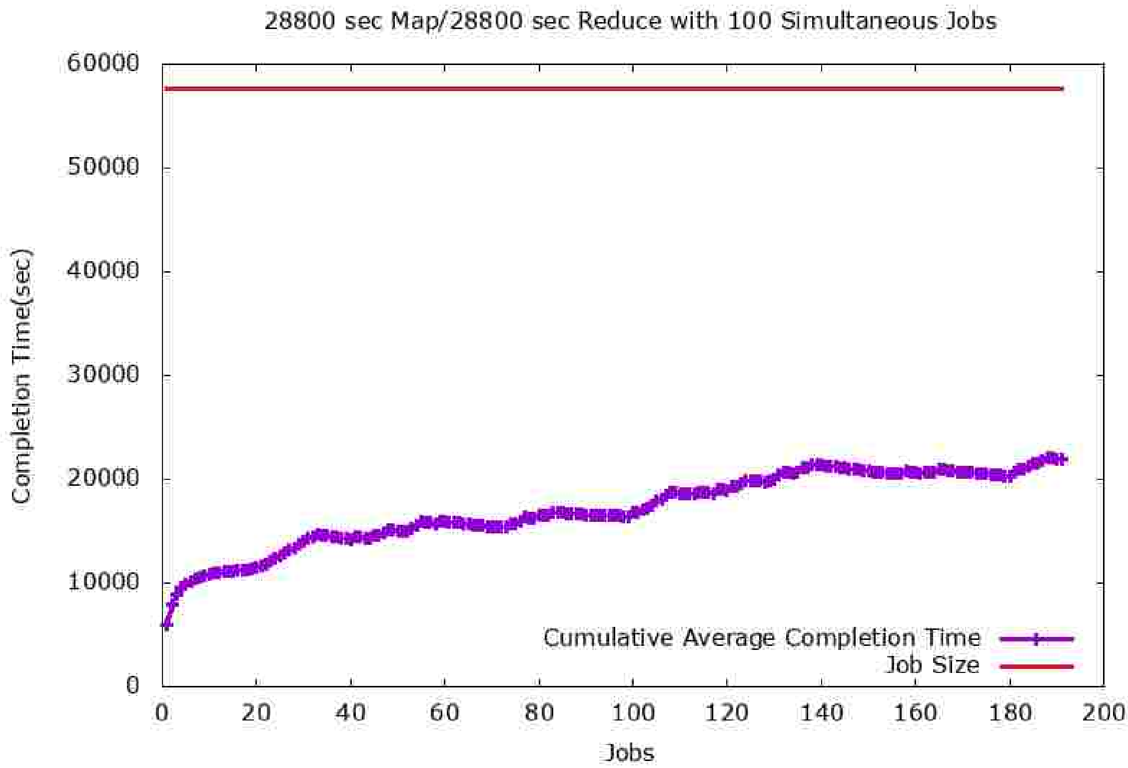


Figure 52: Job Completion Times



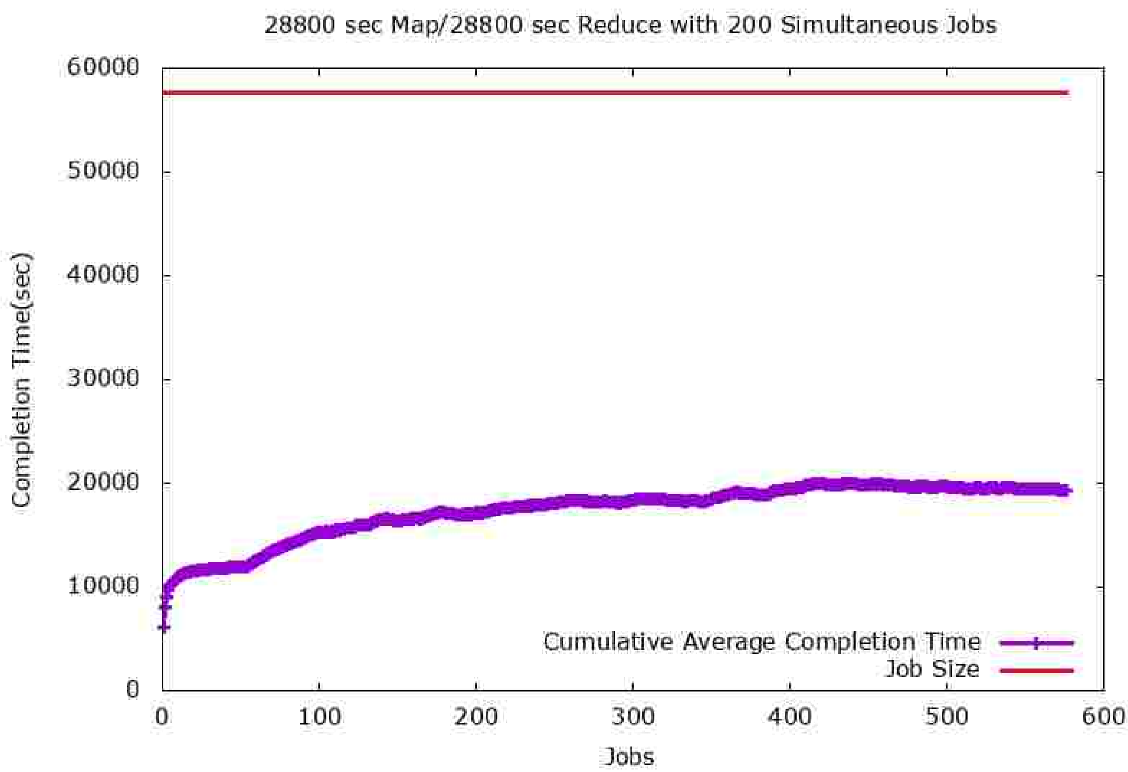
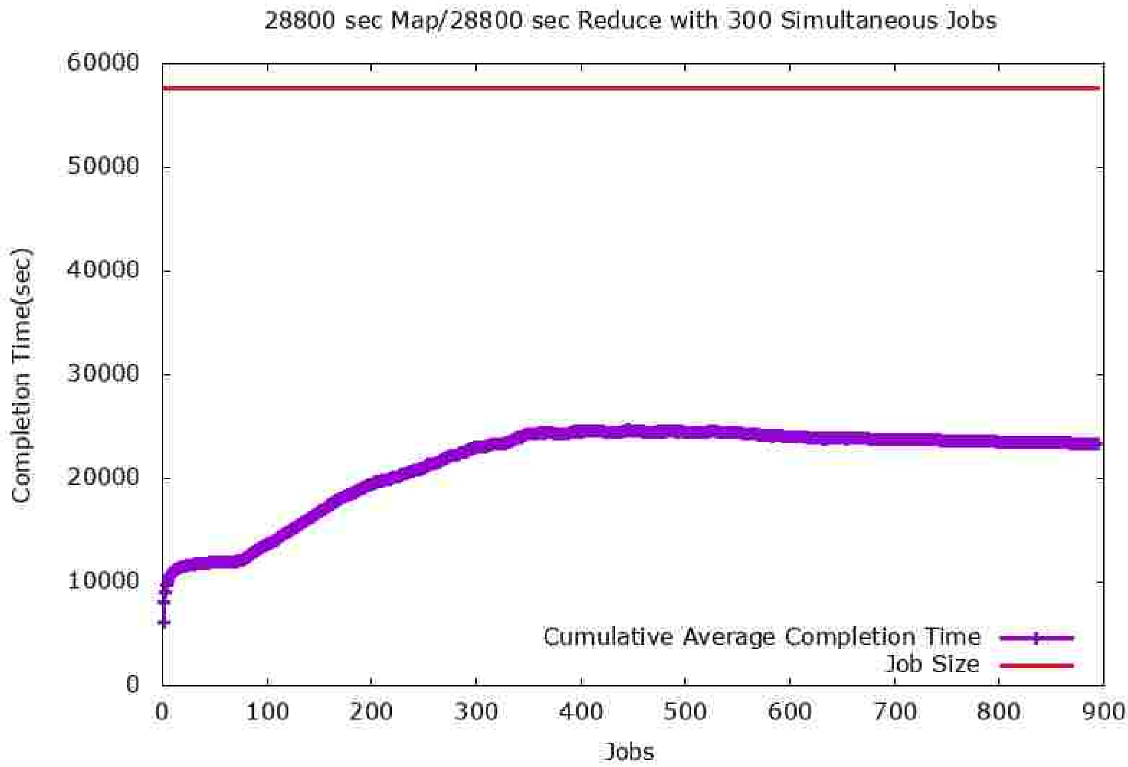


Figure 53: Job Completion Times



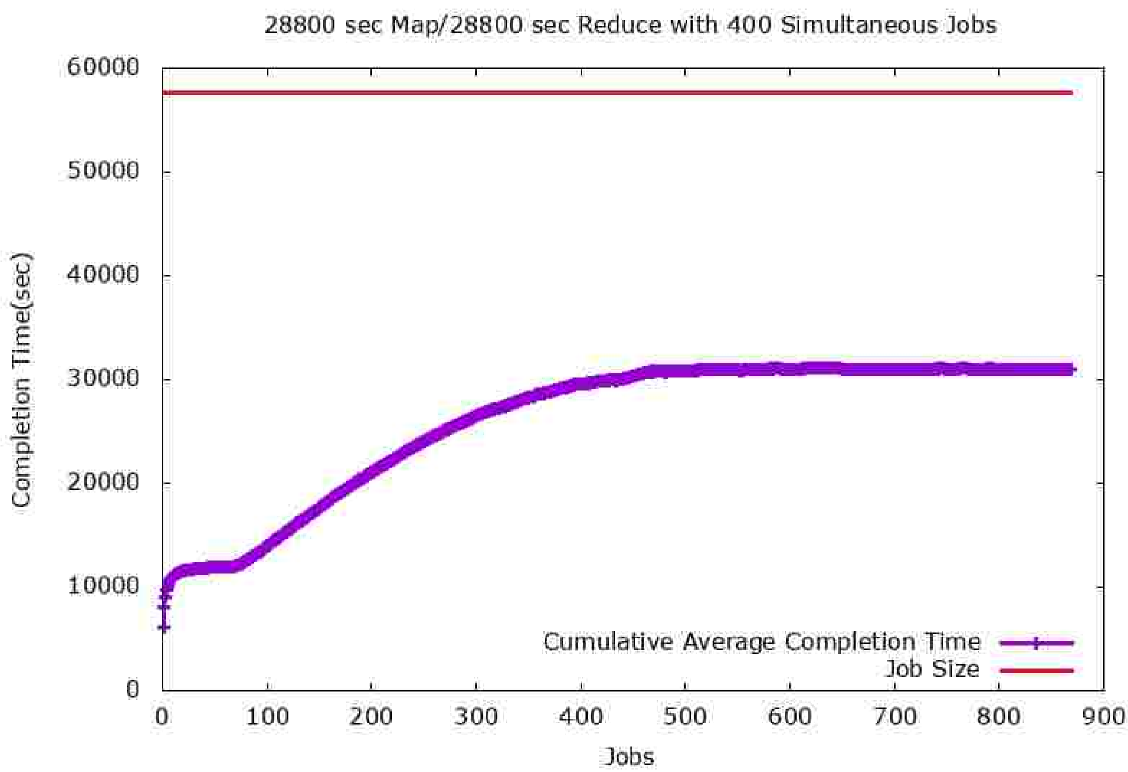
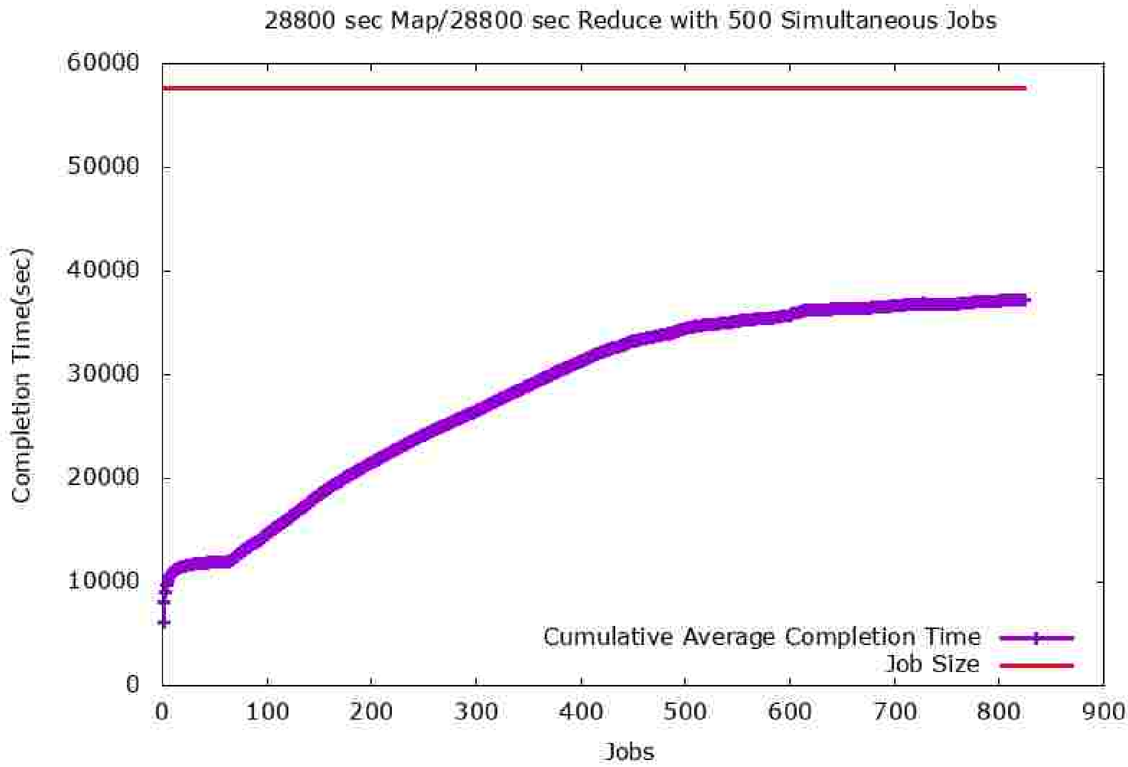


Figure 54: Job Completion Times



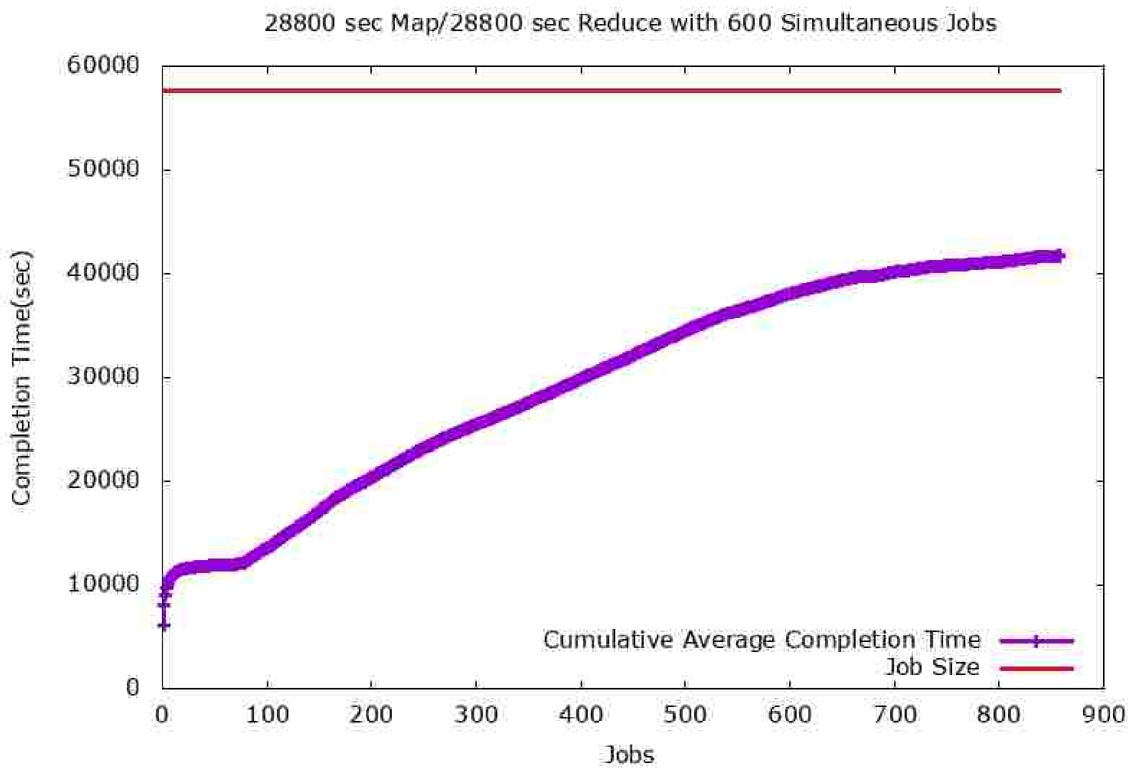
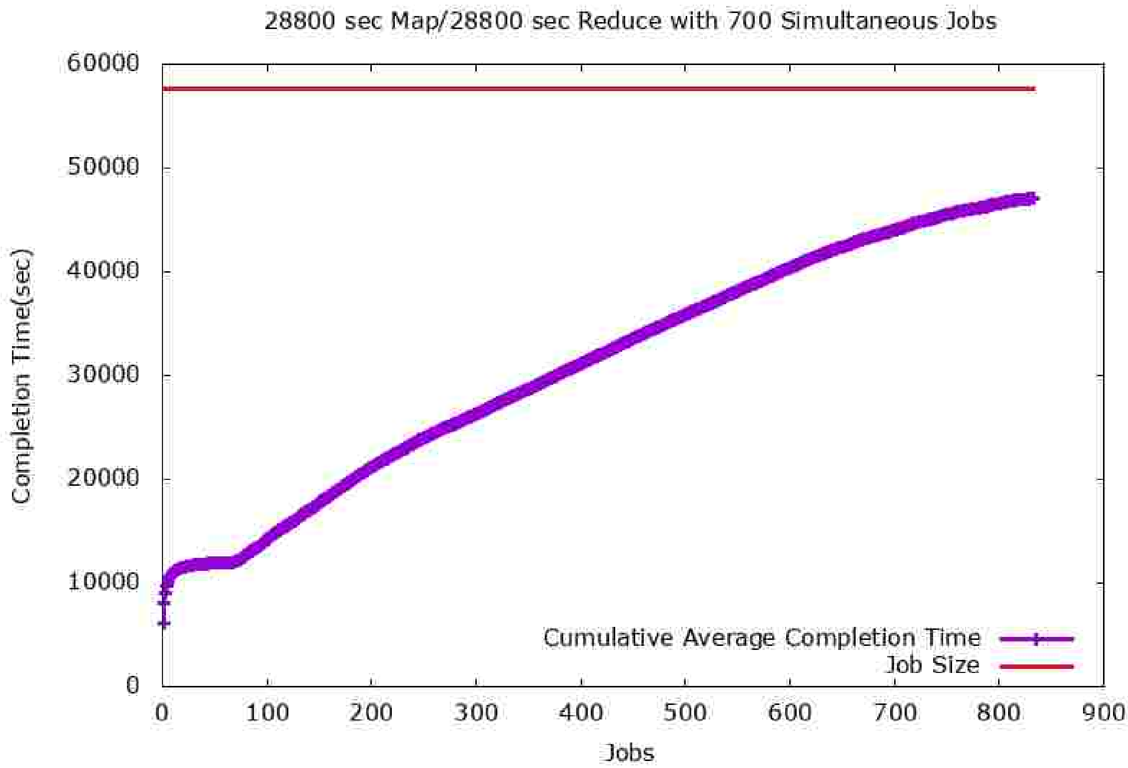


Figure 55: Job Completion Times



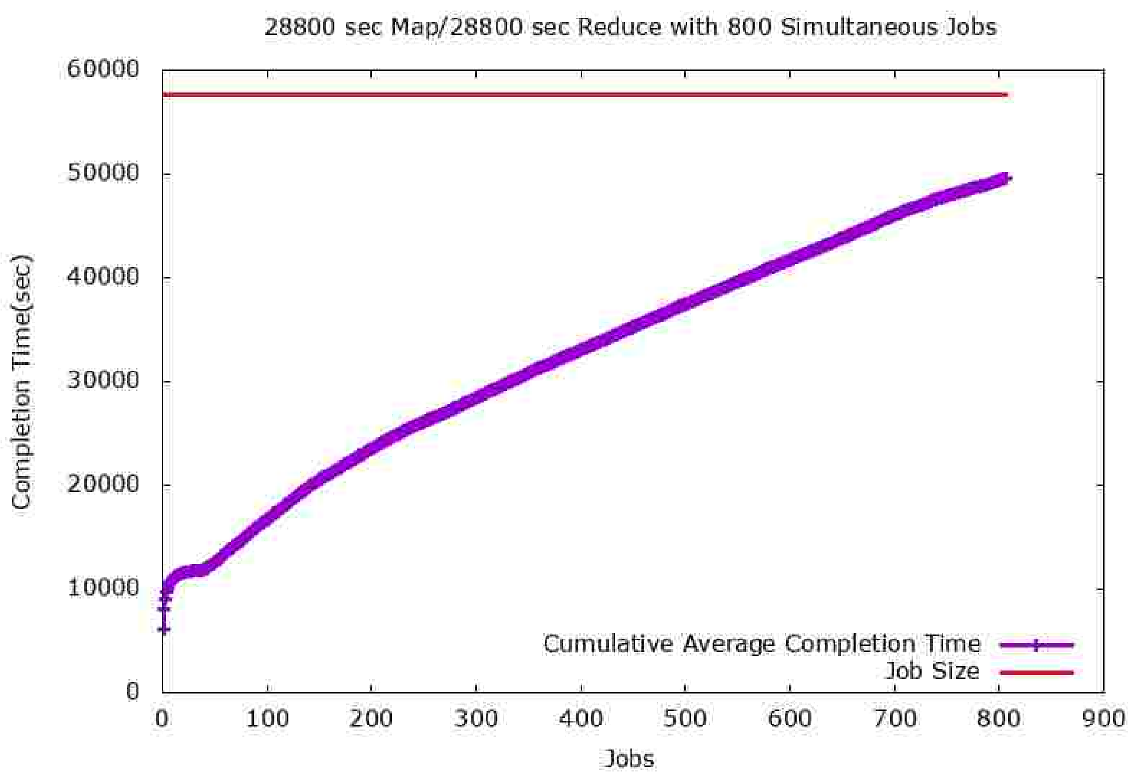
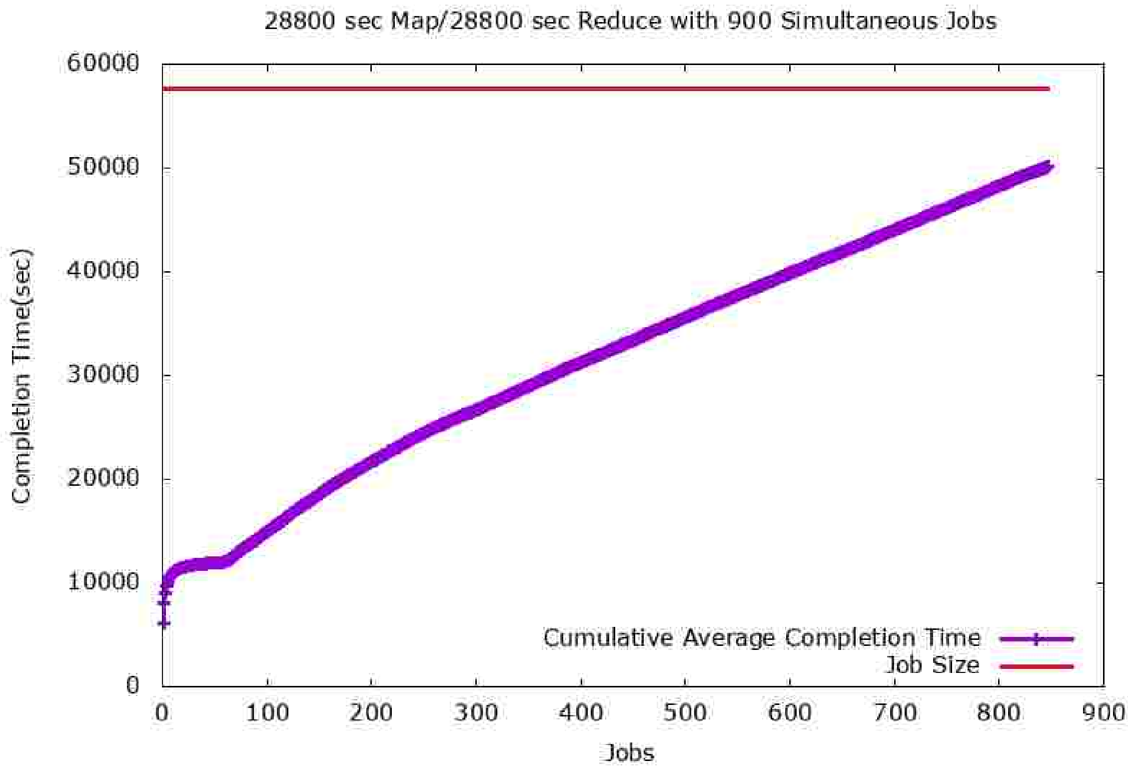


Figure 56: Job Completion Times



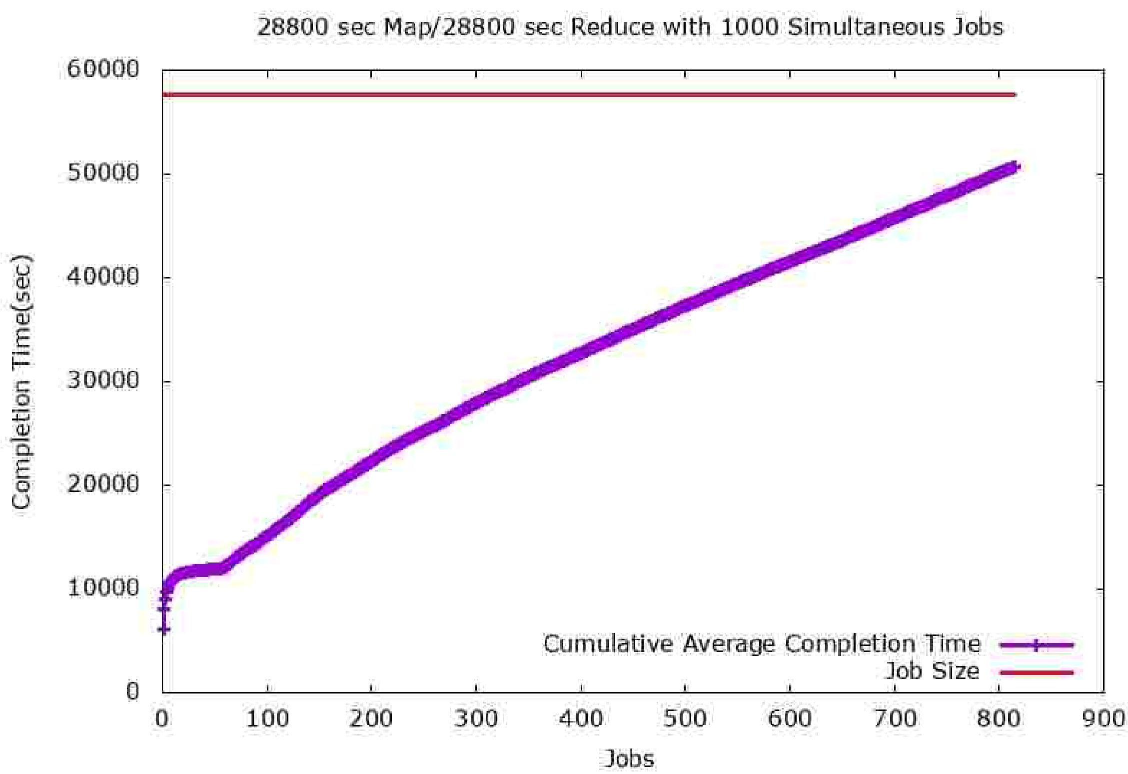
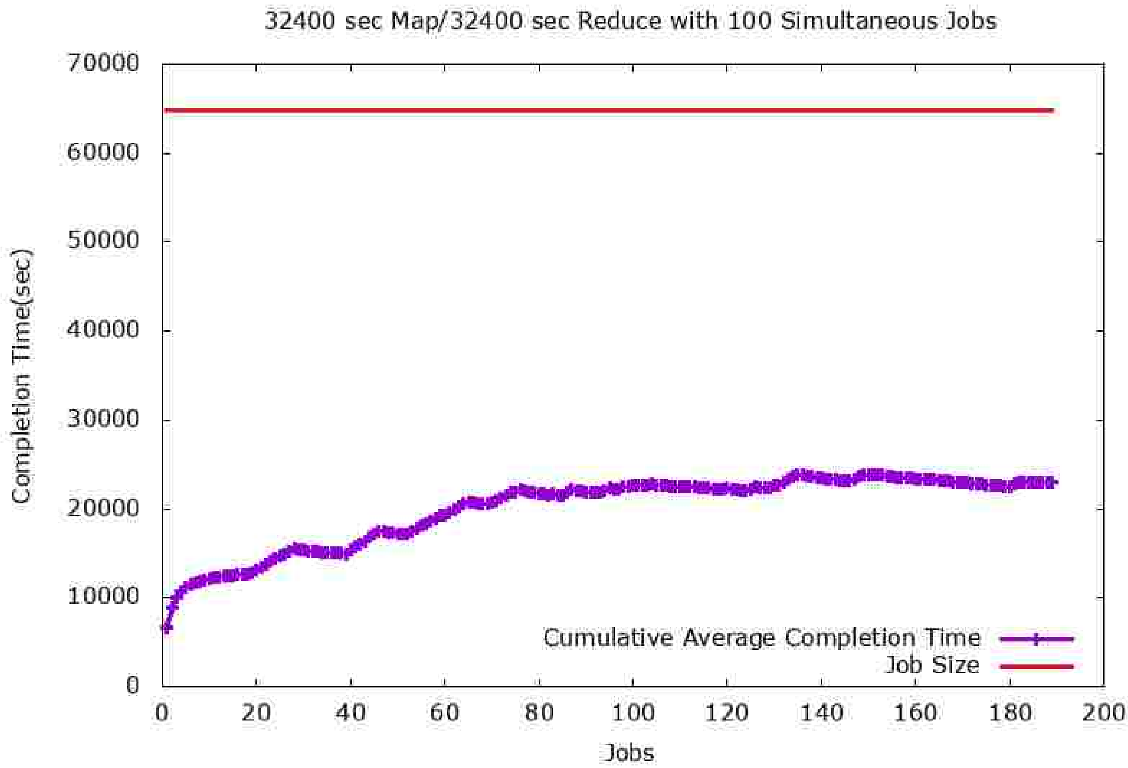


Figure 57: Job Completion Times



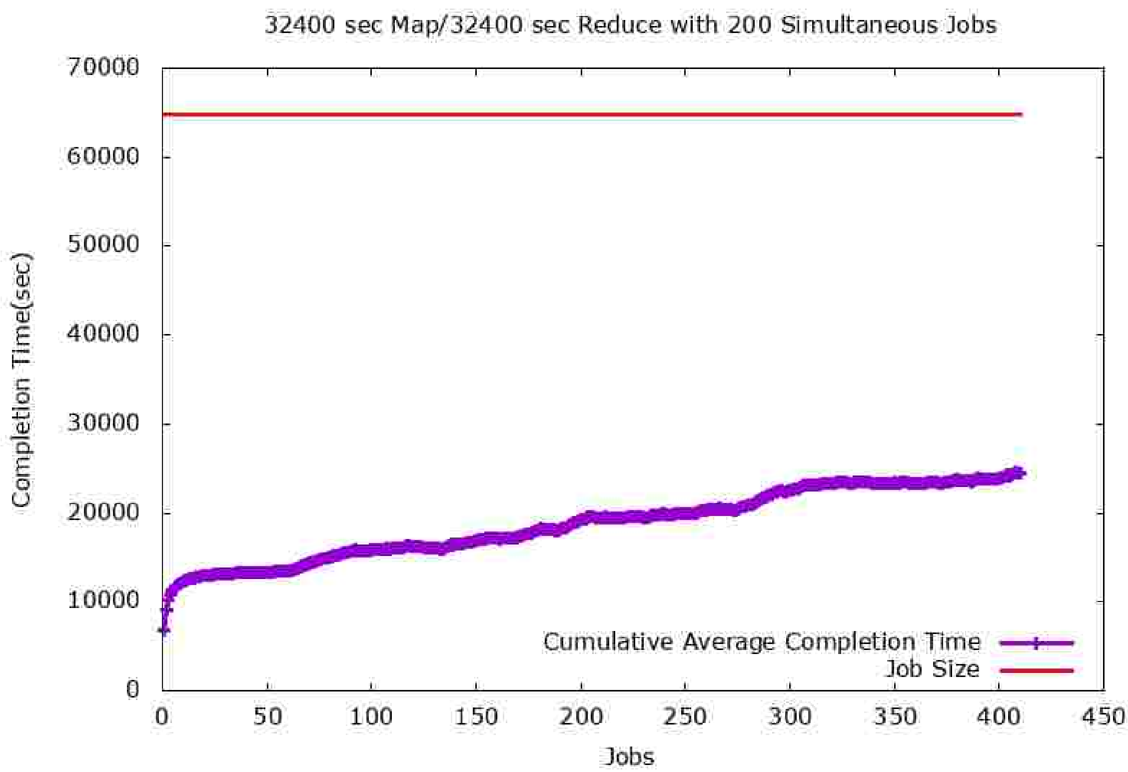
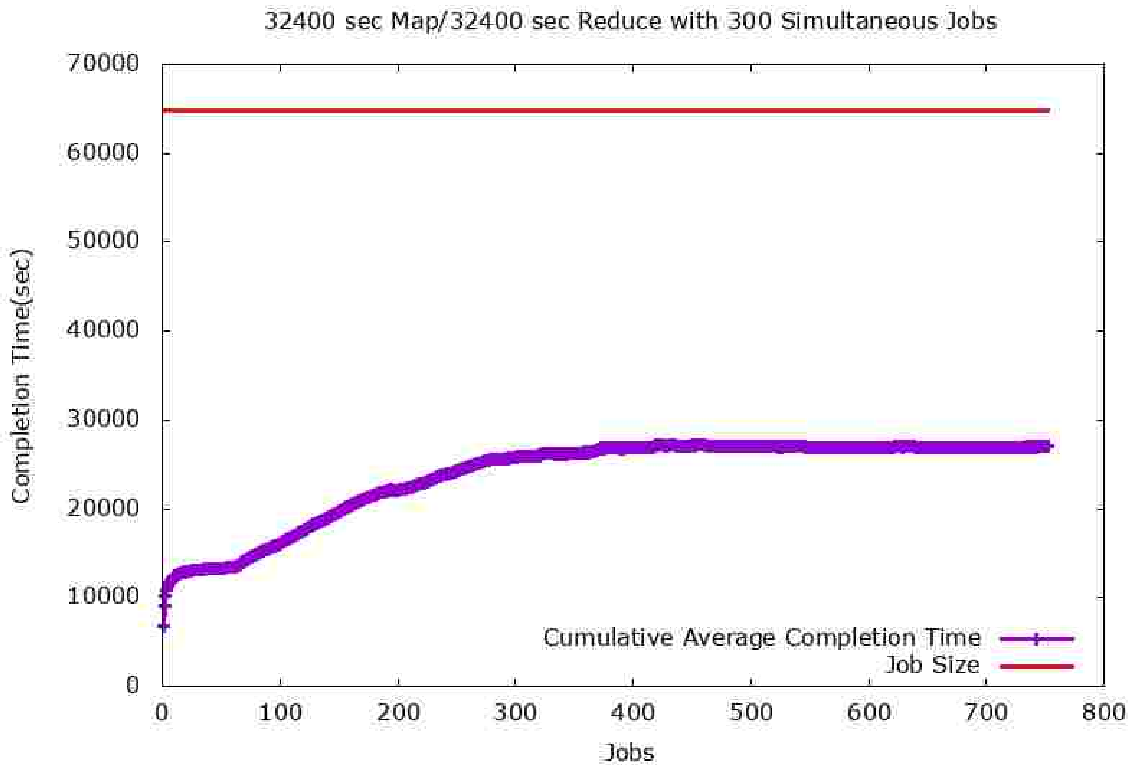


Figure 58: Job Completion Times



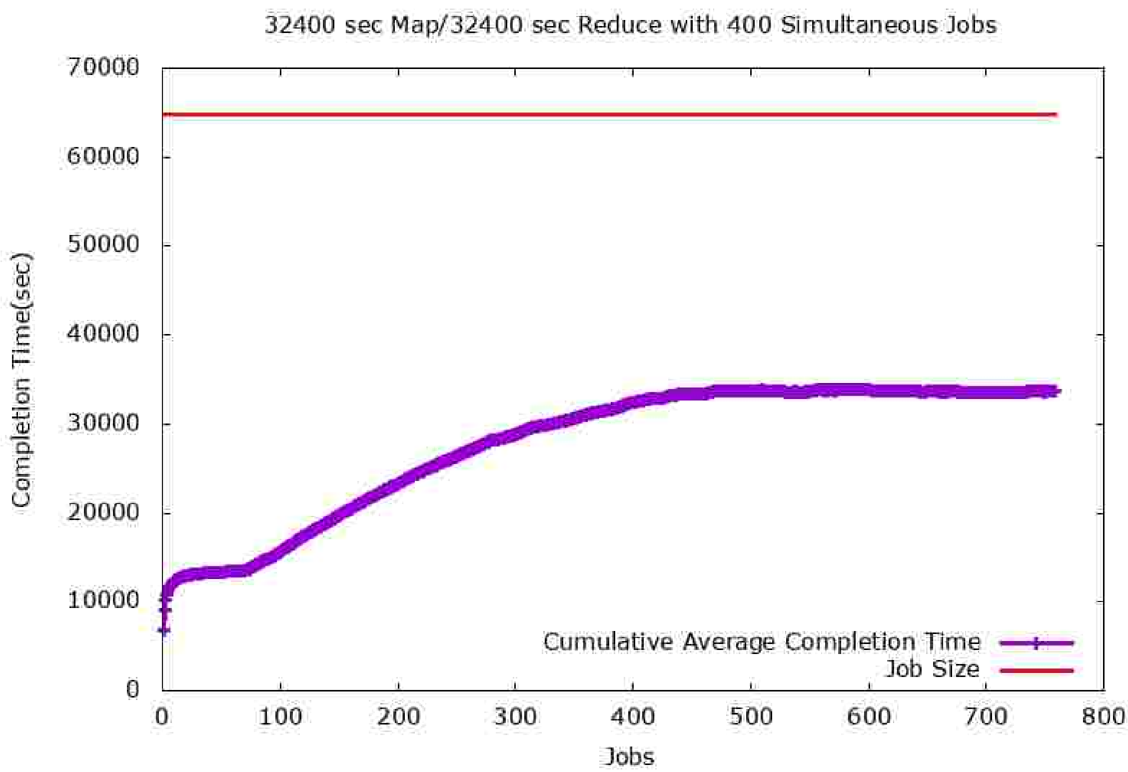
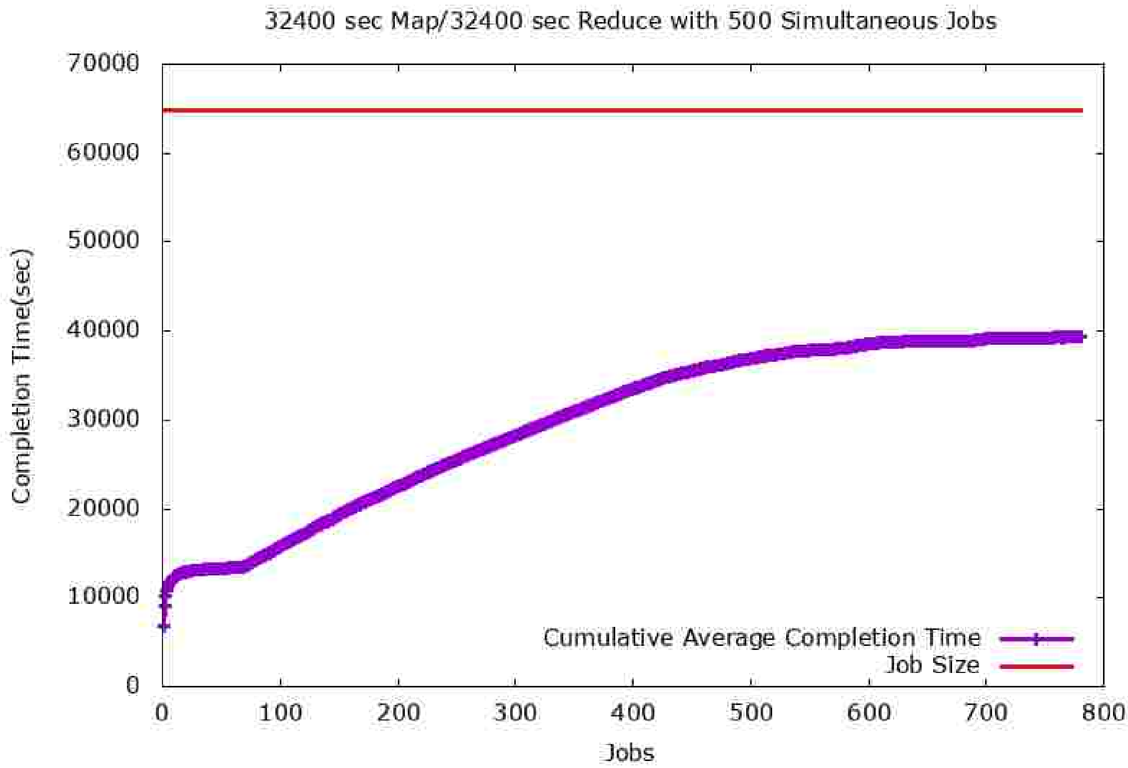


Figure 59: Job Completion Times



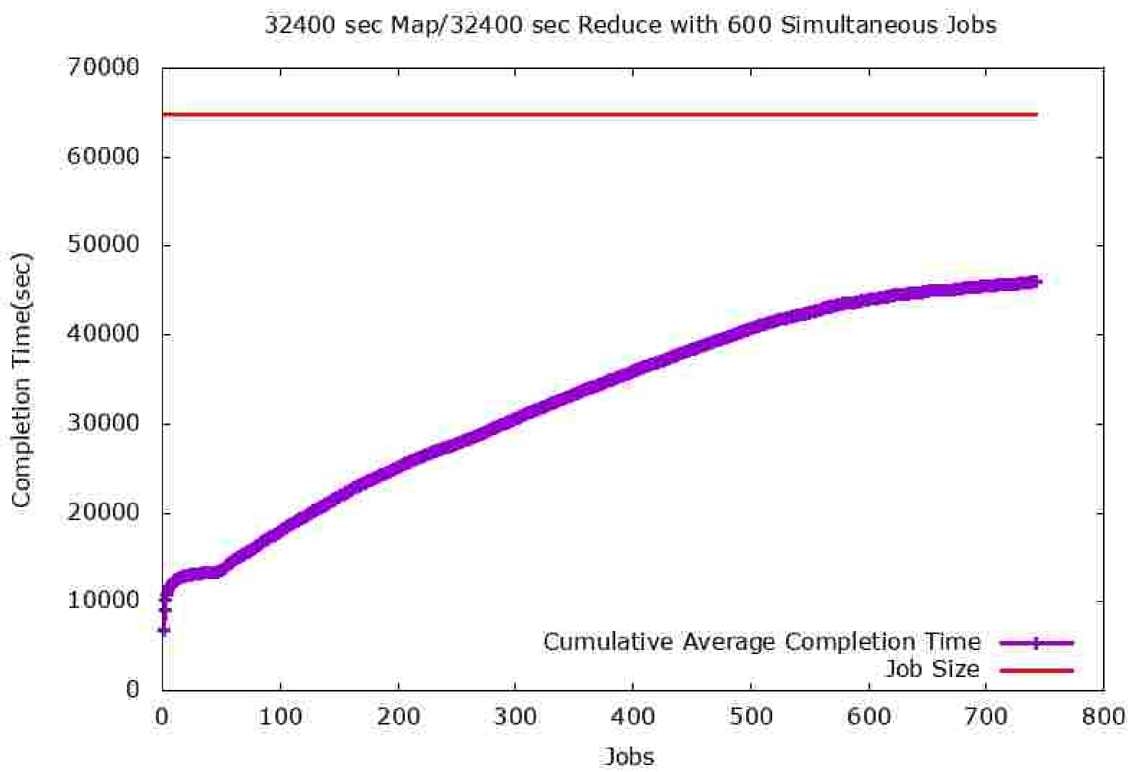


Figure 60: Job Completion Times

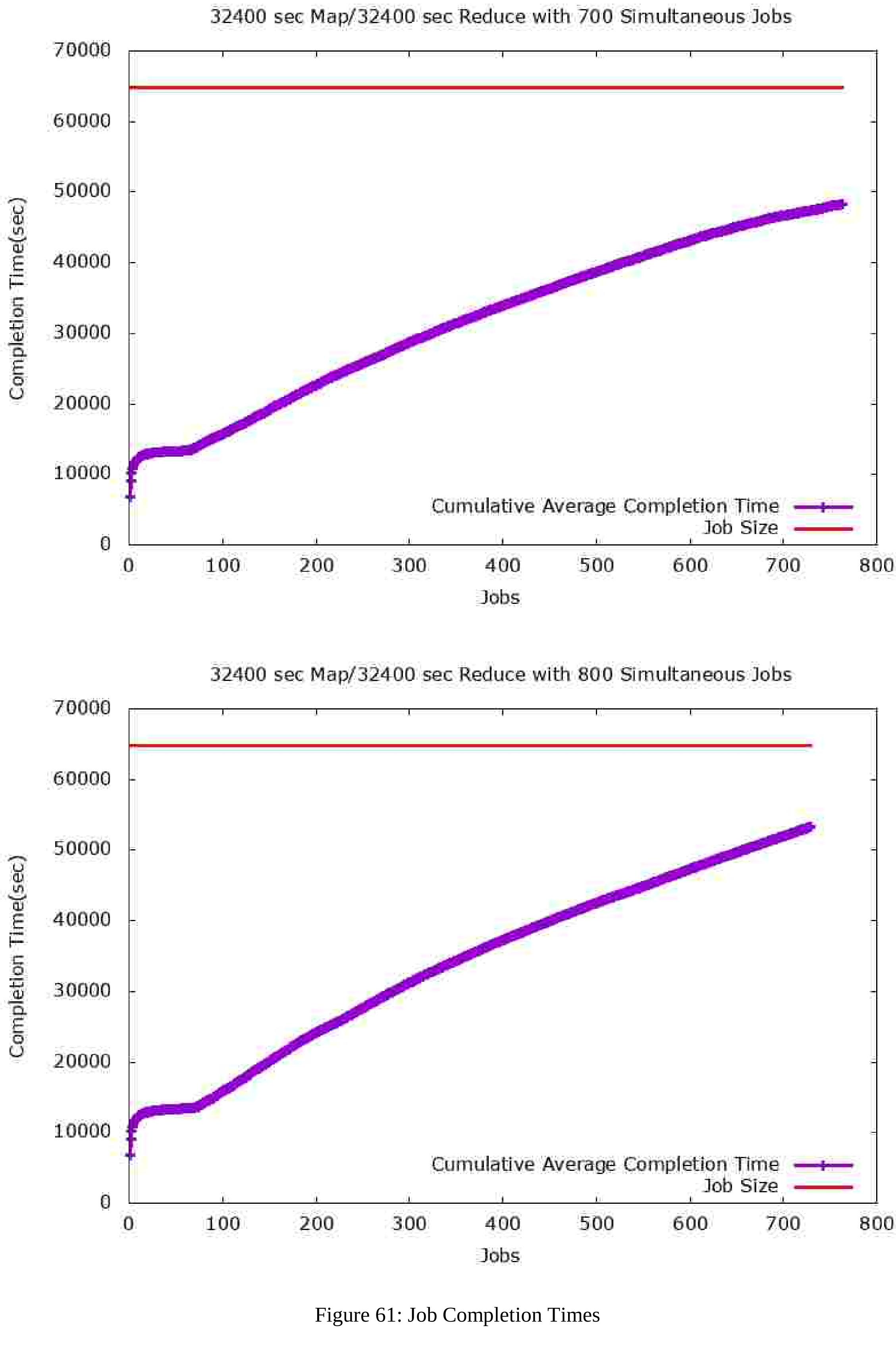


Figure 61: Job Completion Times

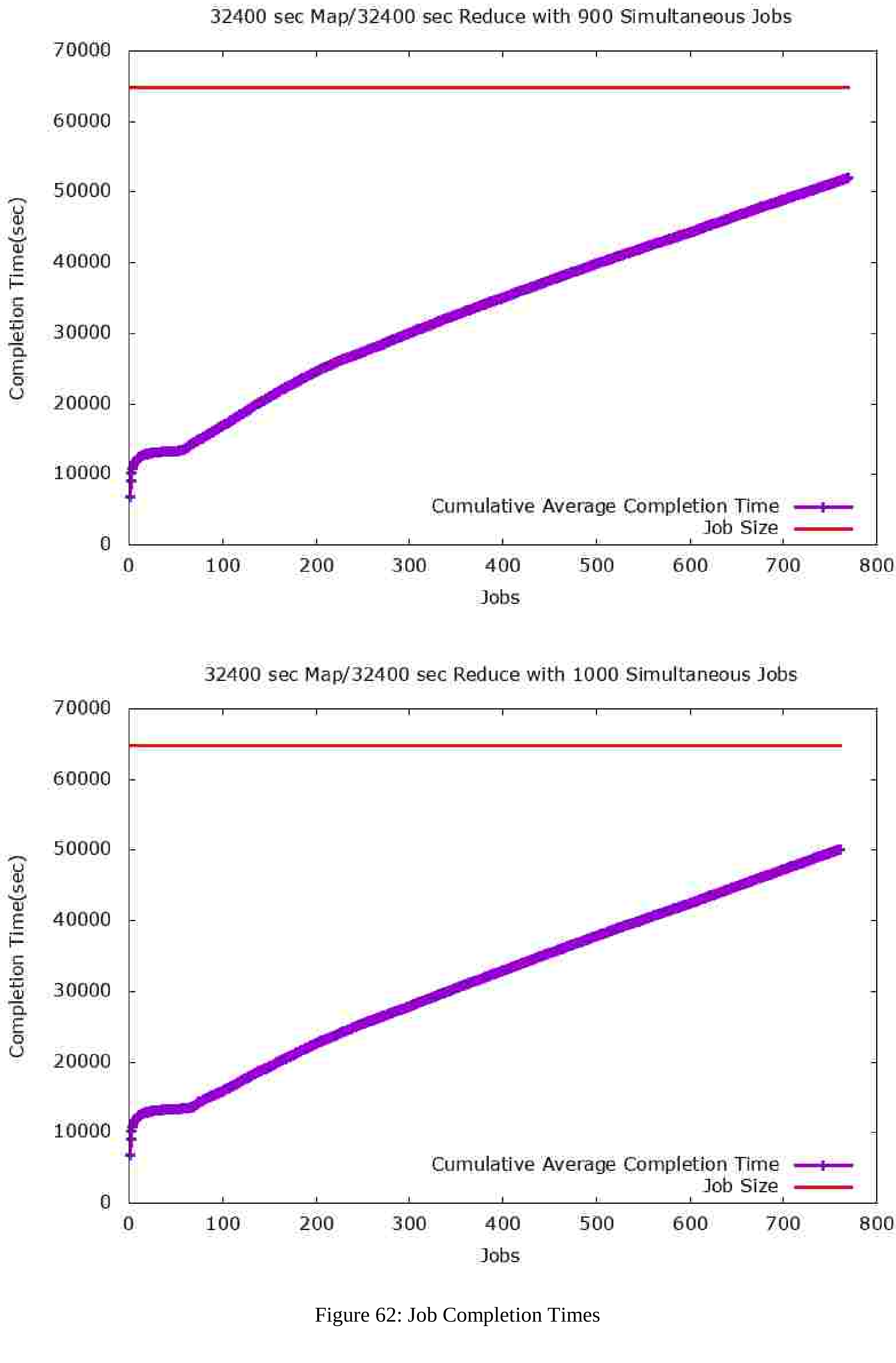
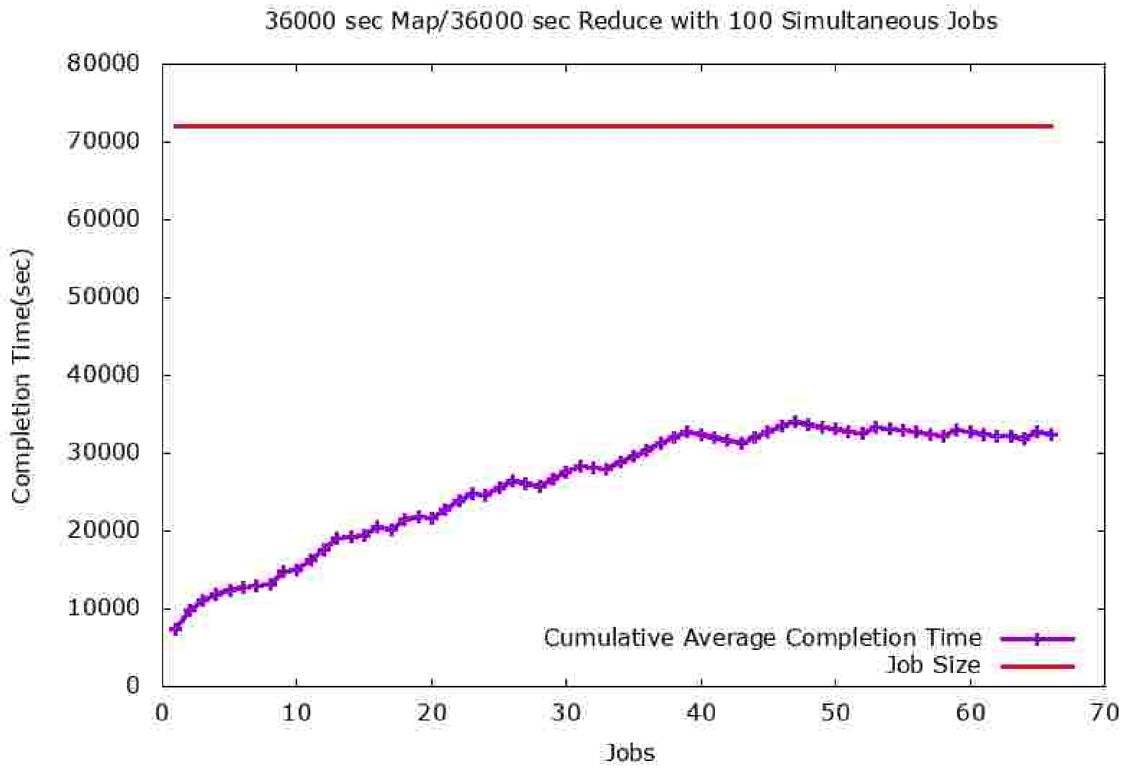


Figure 62: Job Completion Times



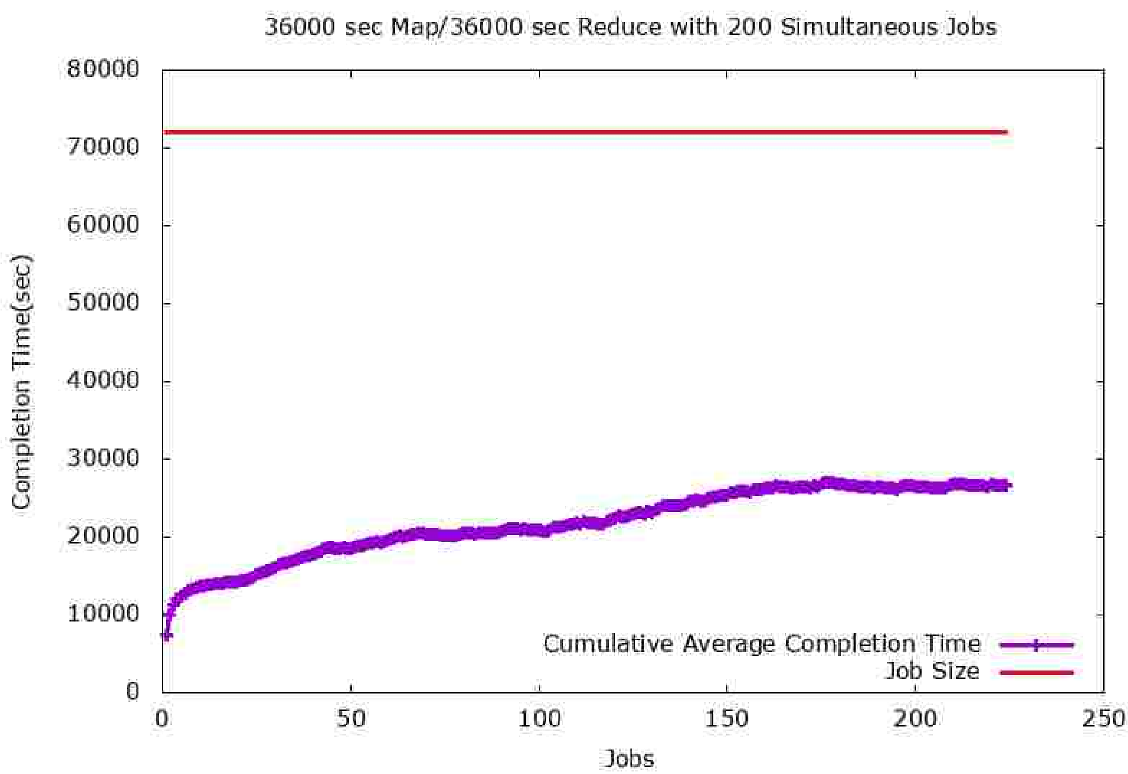
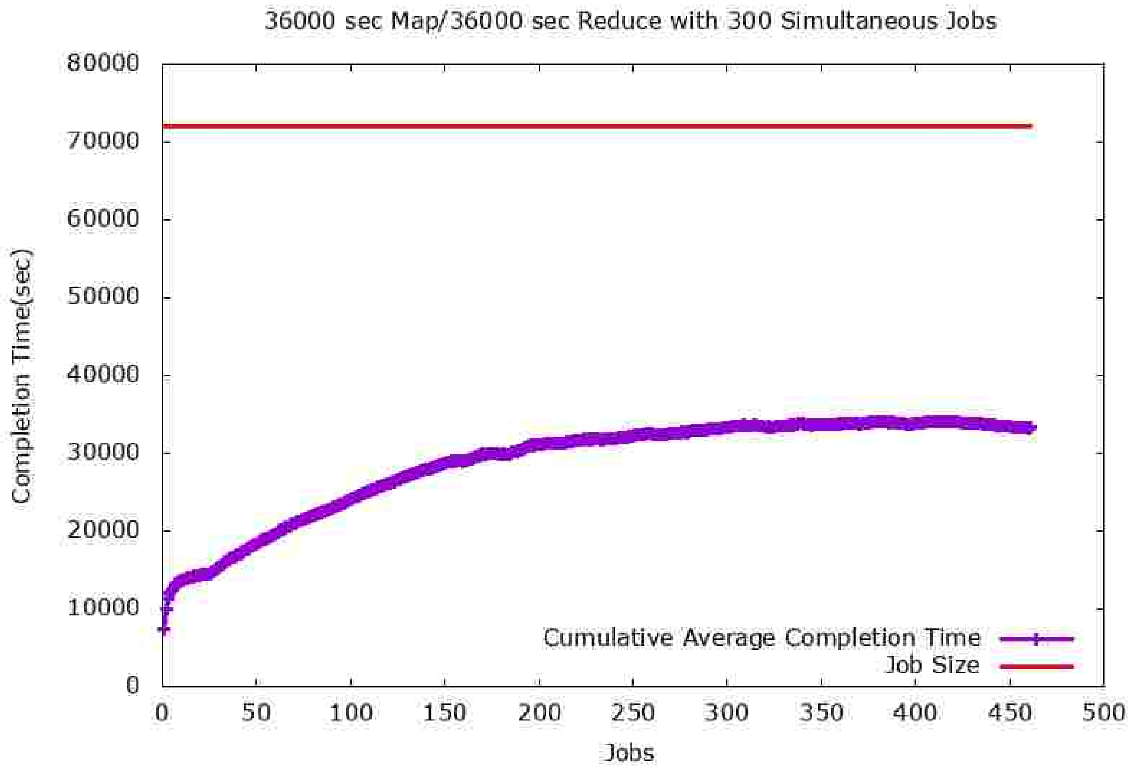


Figure 63: Job Completion Times



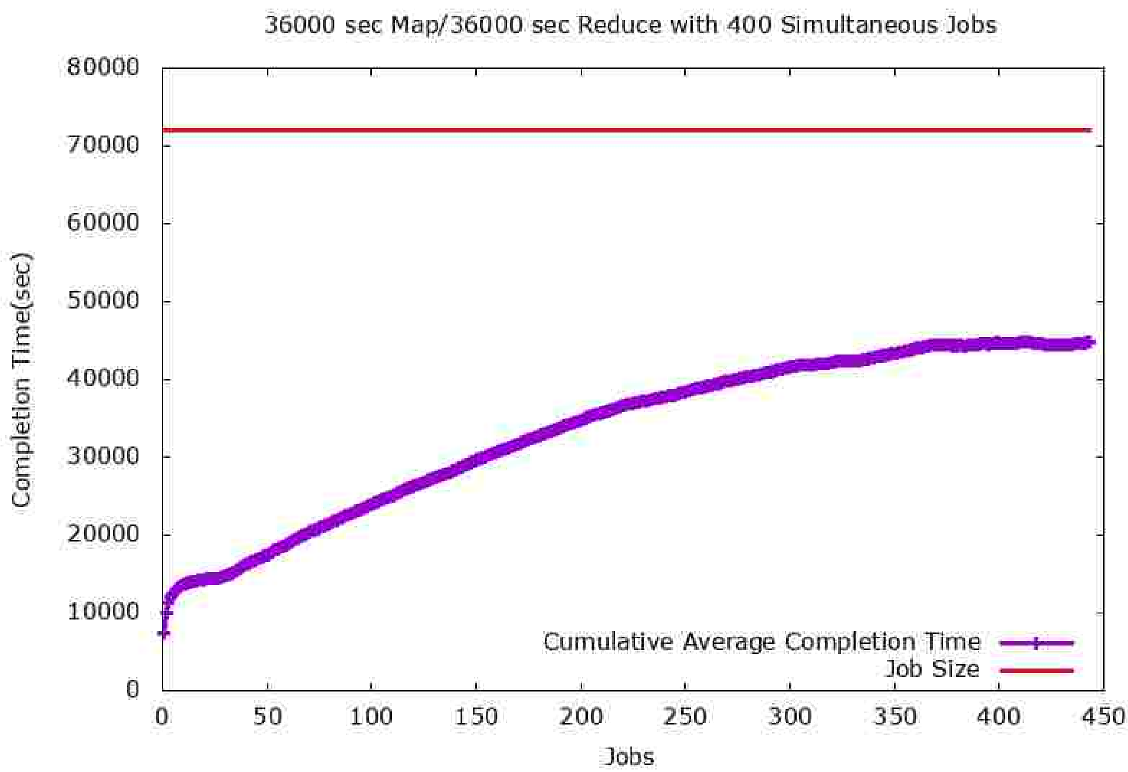
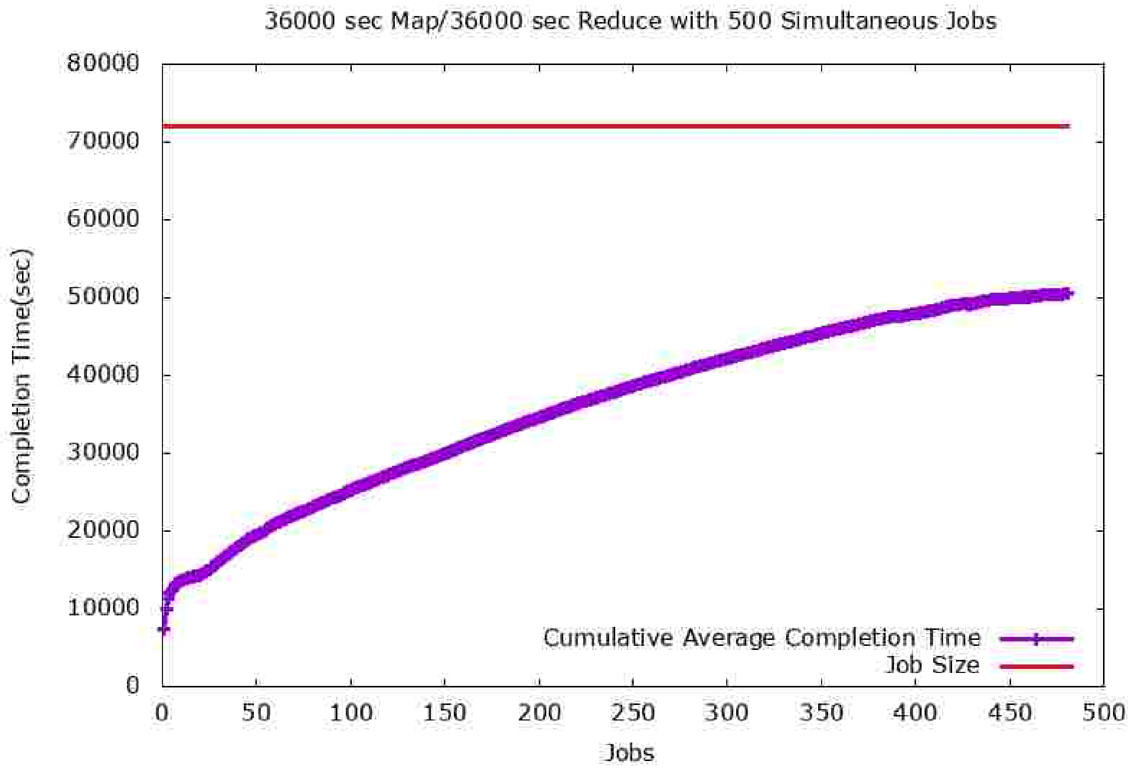


Figure 64: Job Completion Times



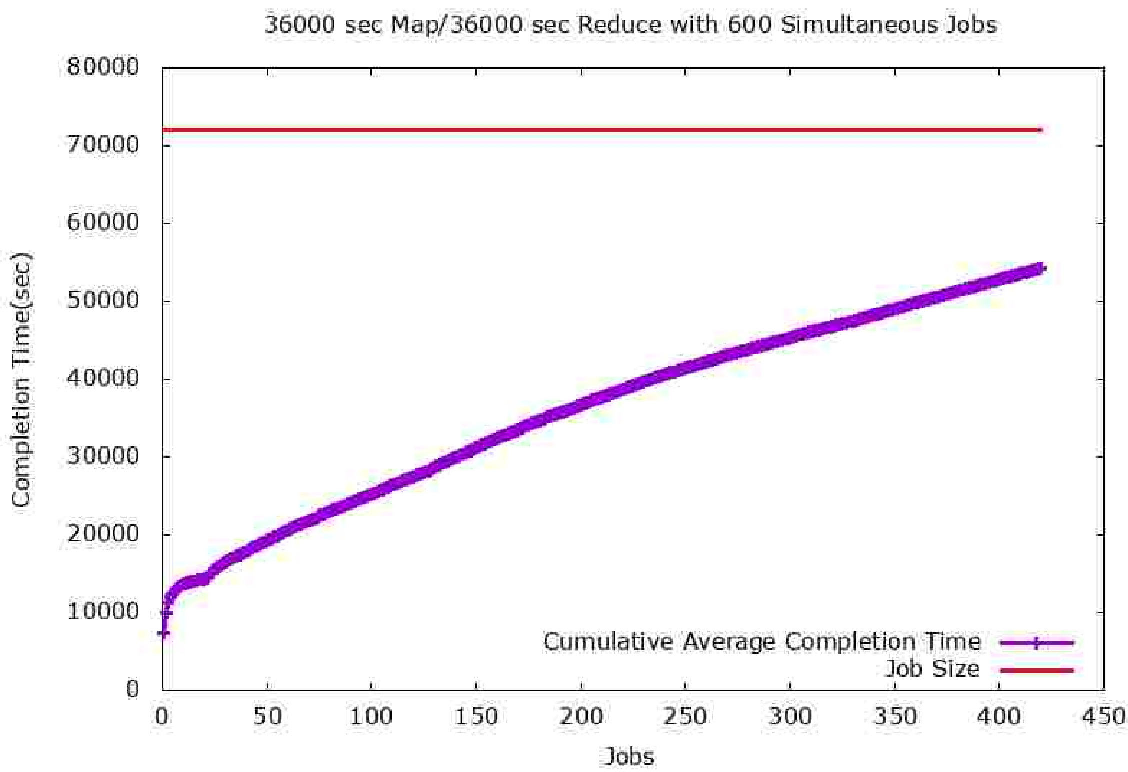
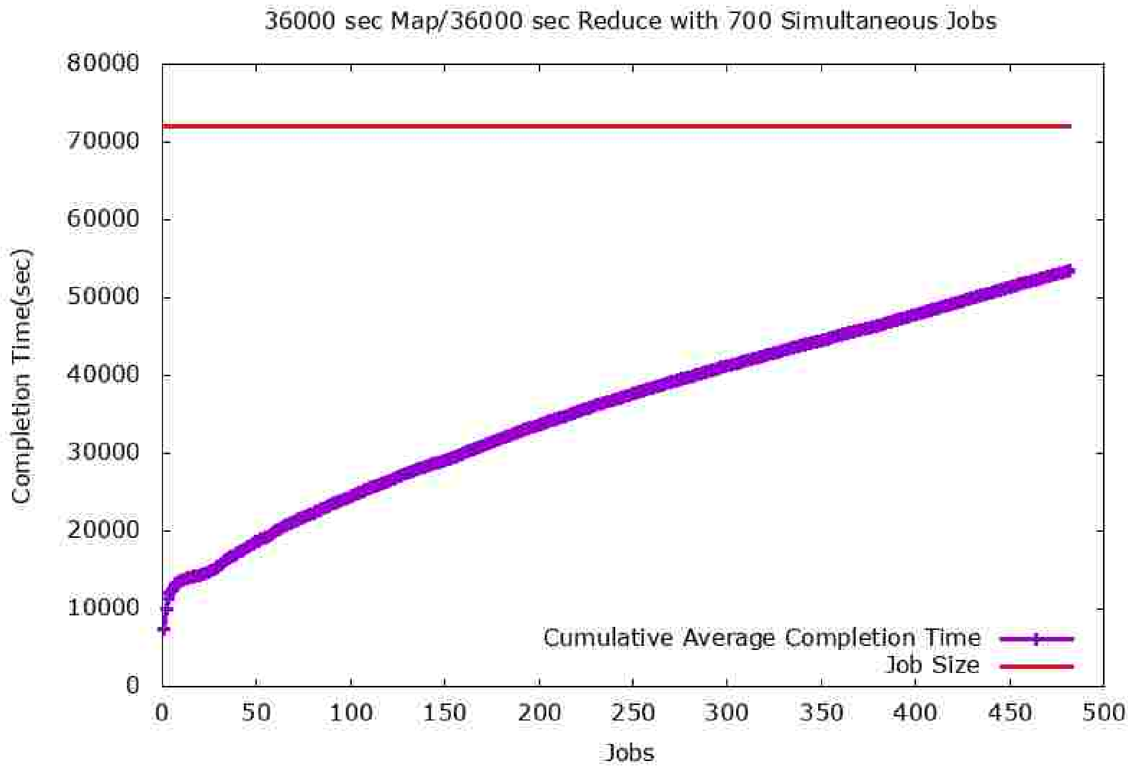


Figure 65: Job Completion Times



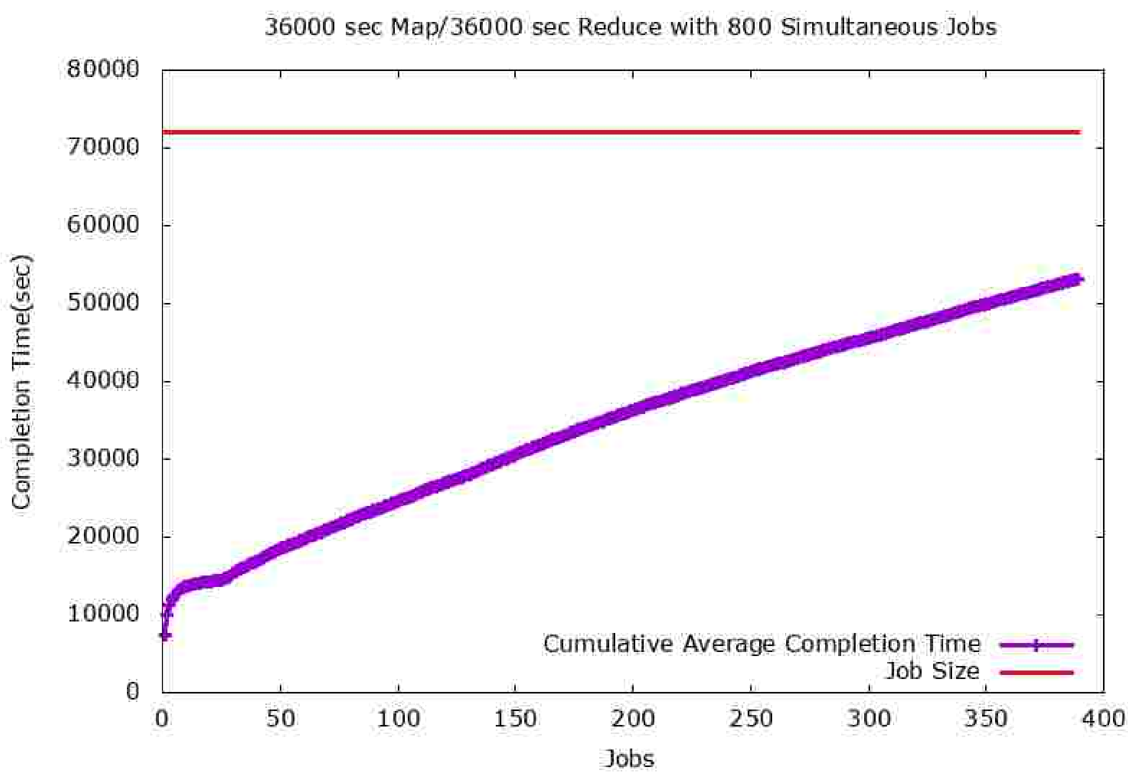
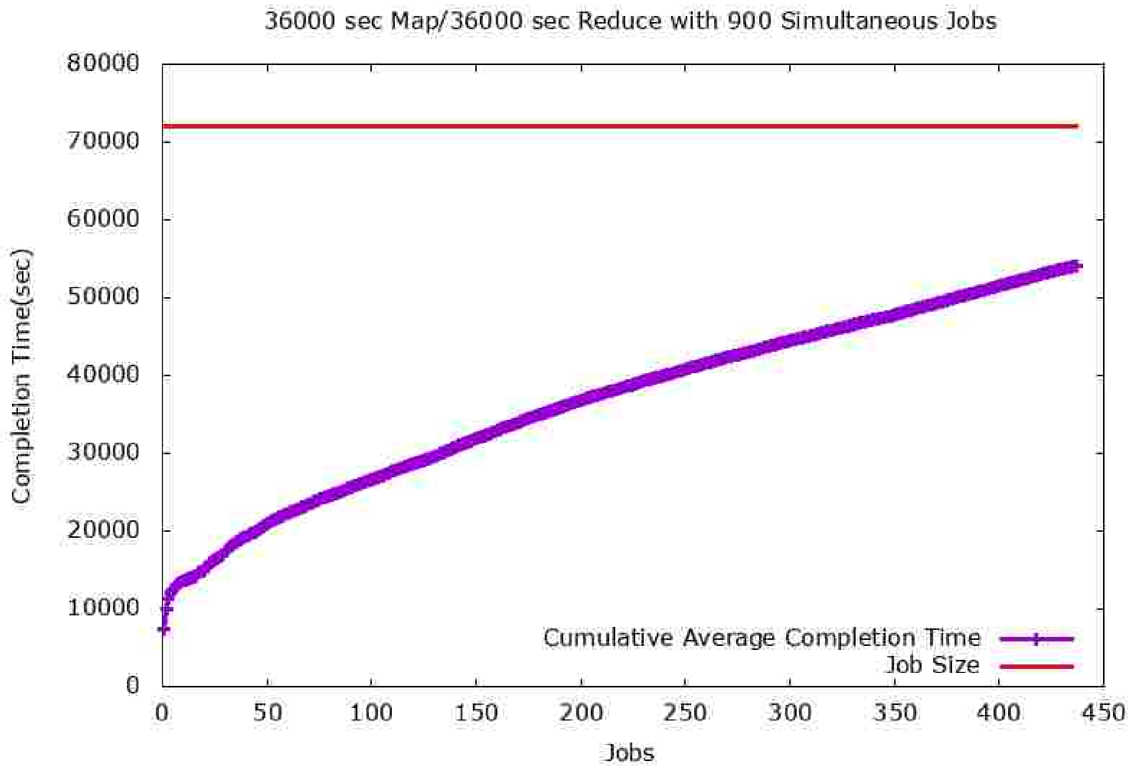


Figure 66: Job Completion Times



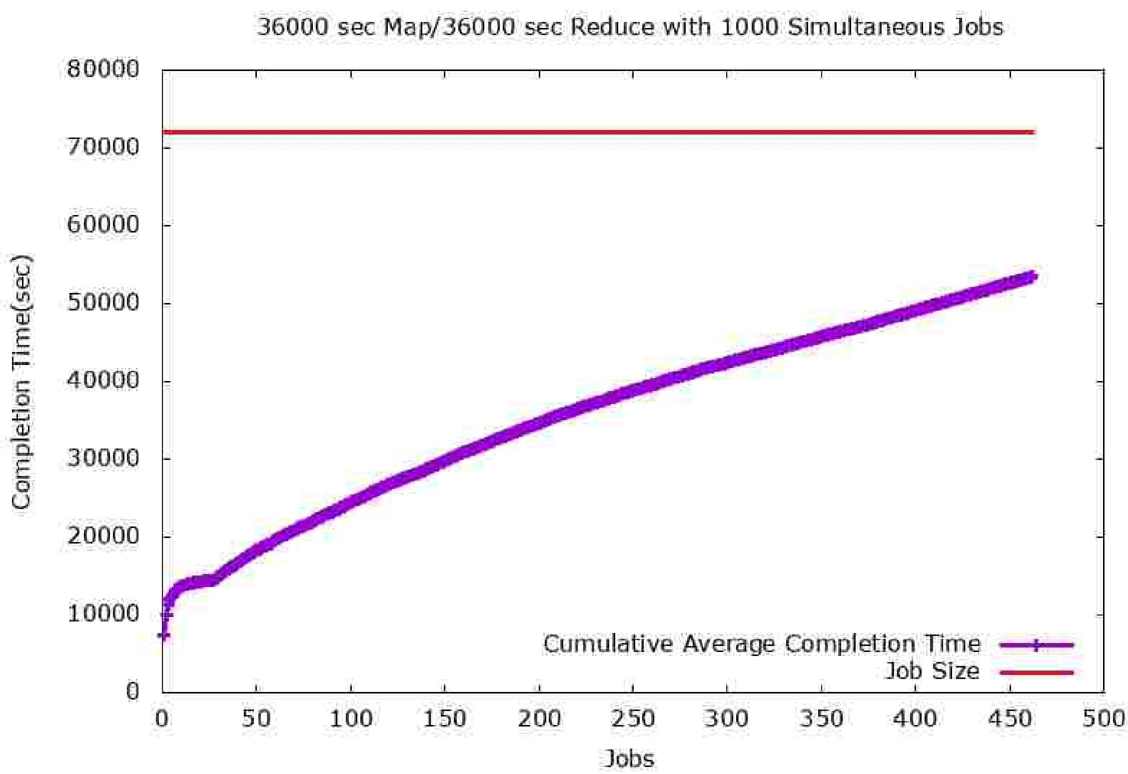


Figure 67: Job Completion Times

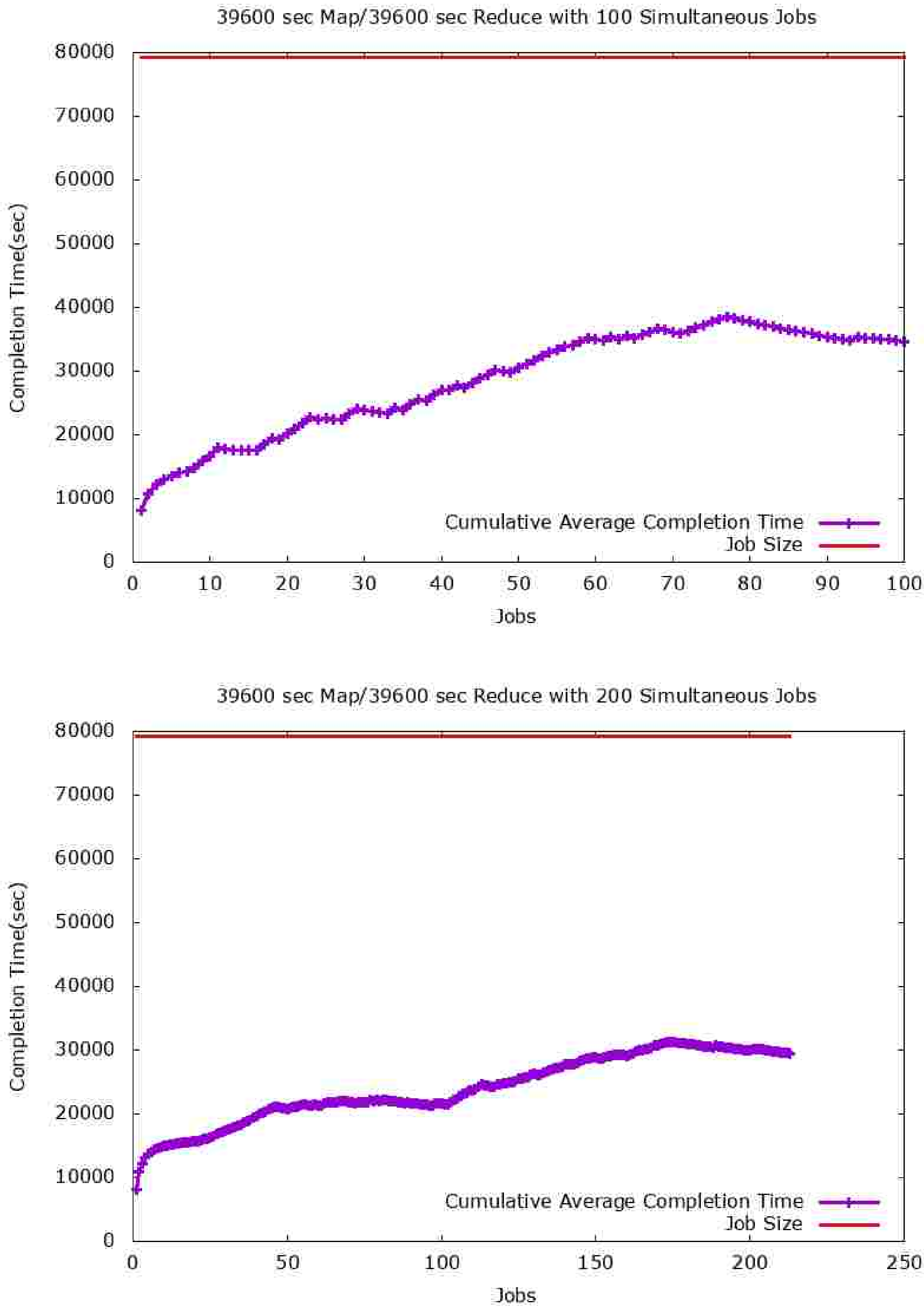
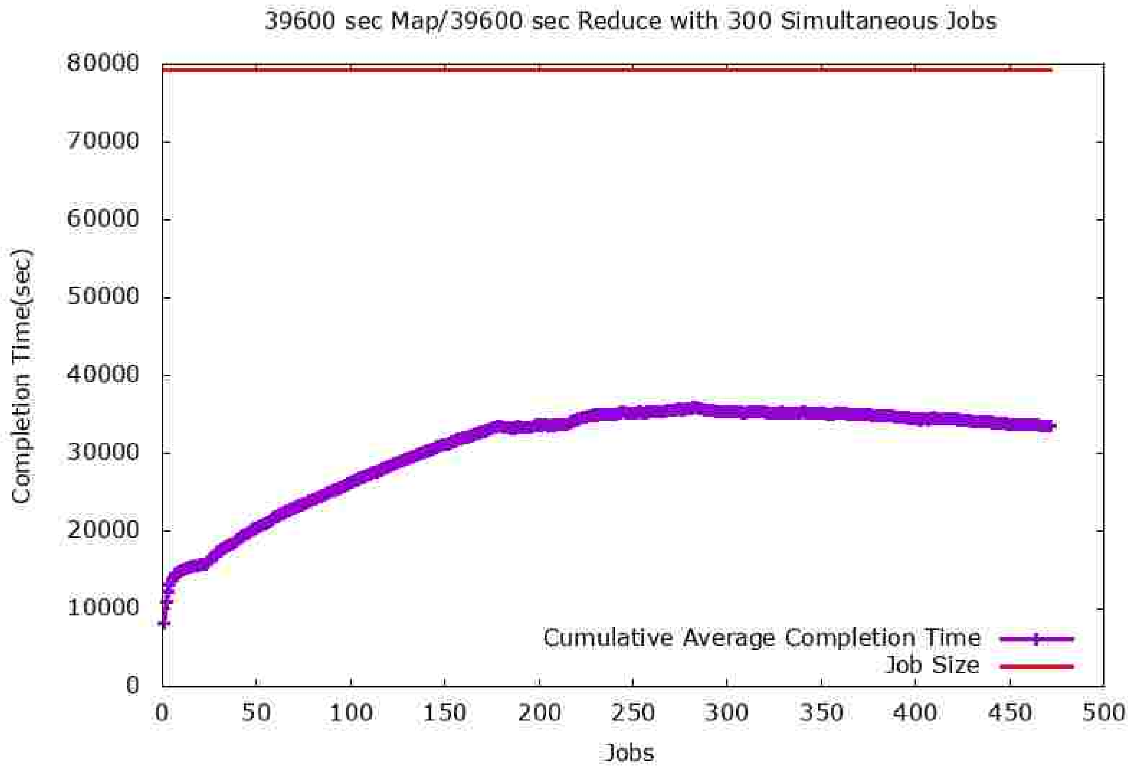


Figure 68: Job Completion Times



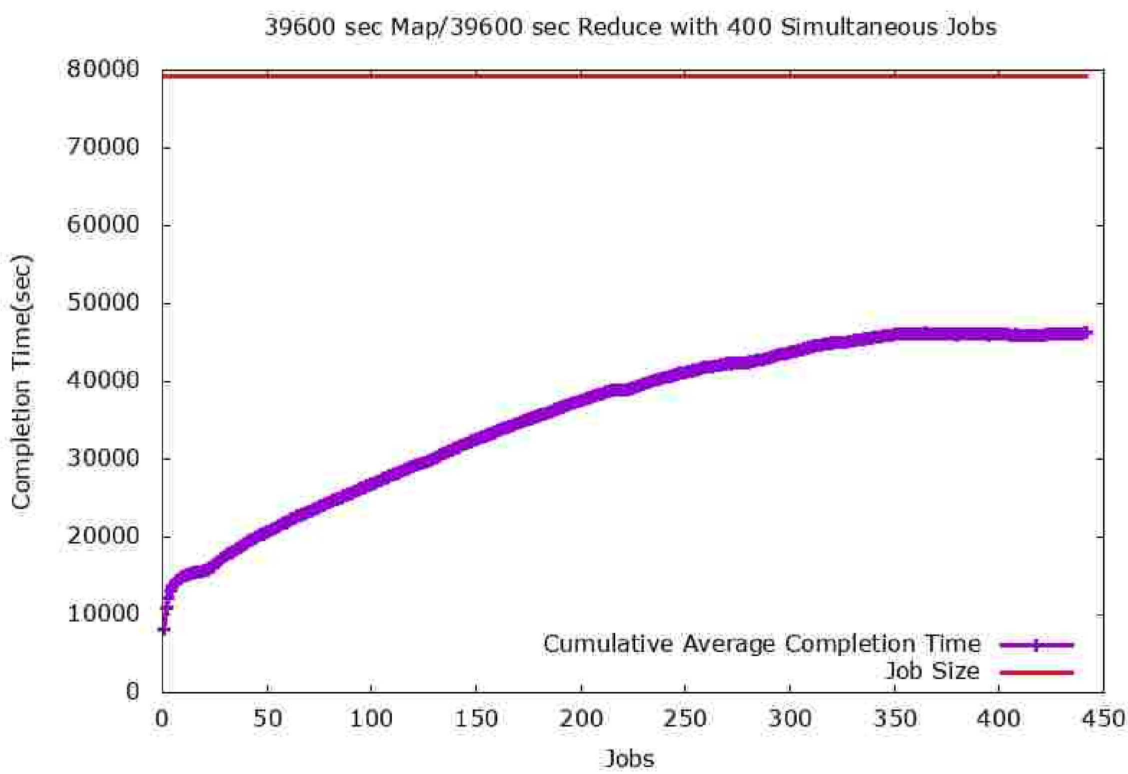
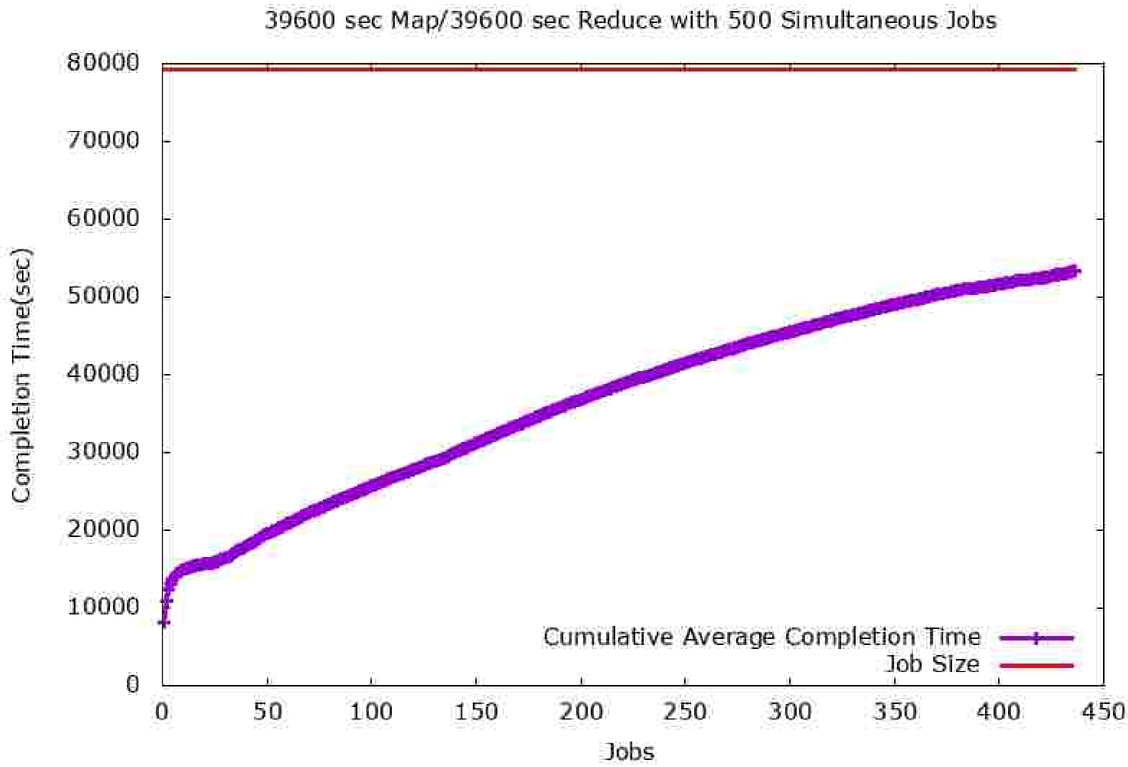


Figure 69: Job Completion Times



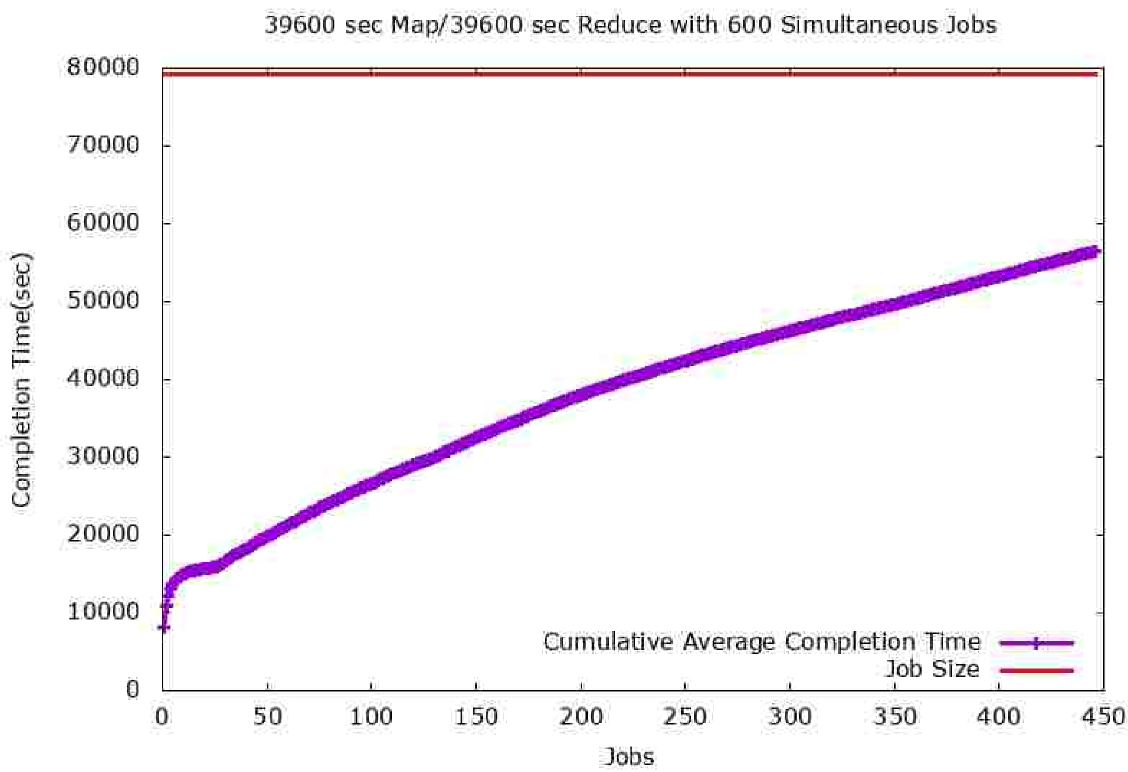


Figure 70: Job Completion Times

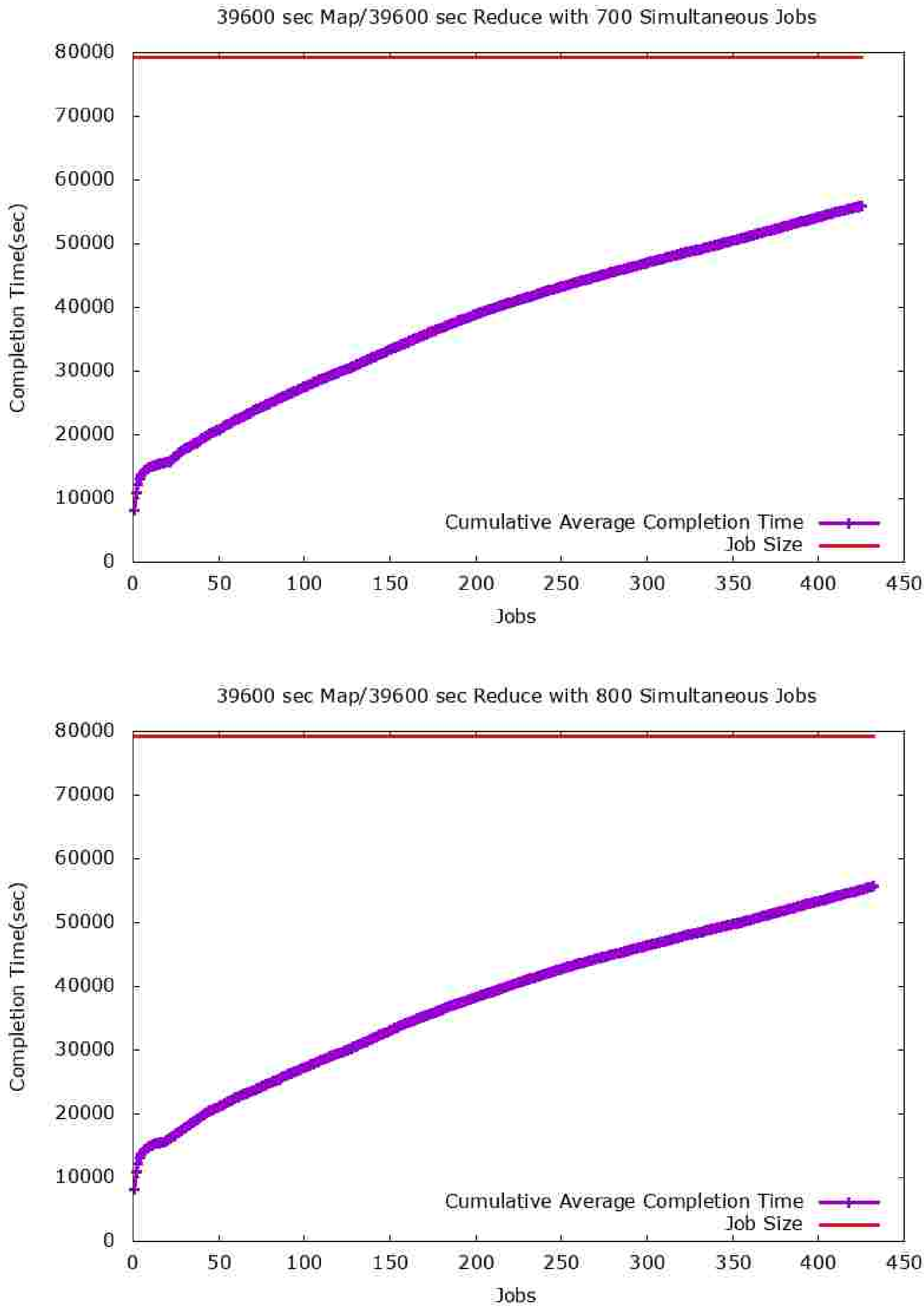
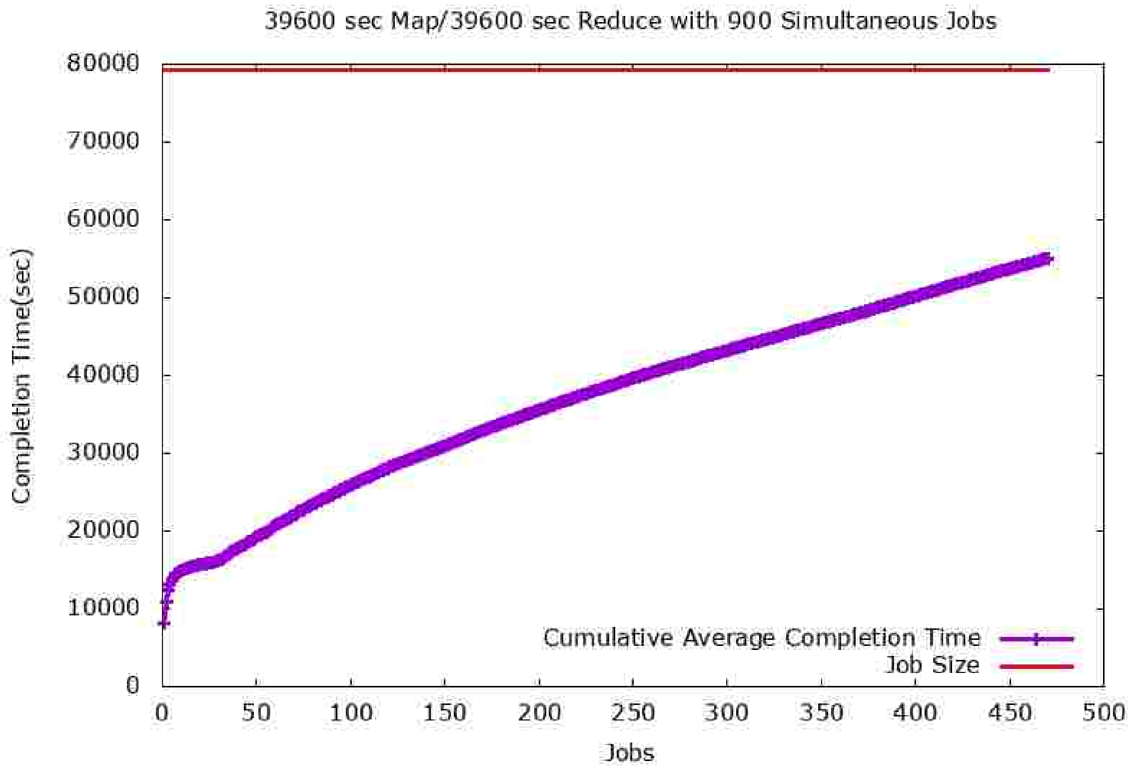


Figure 71: Job Completion Times



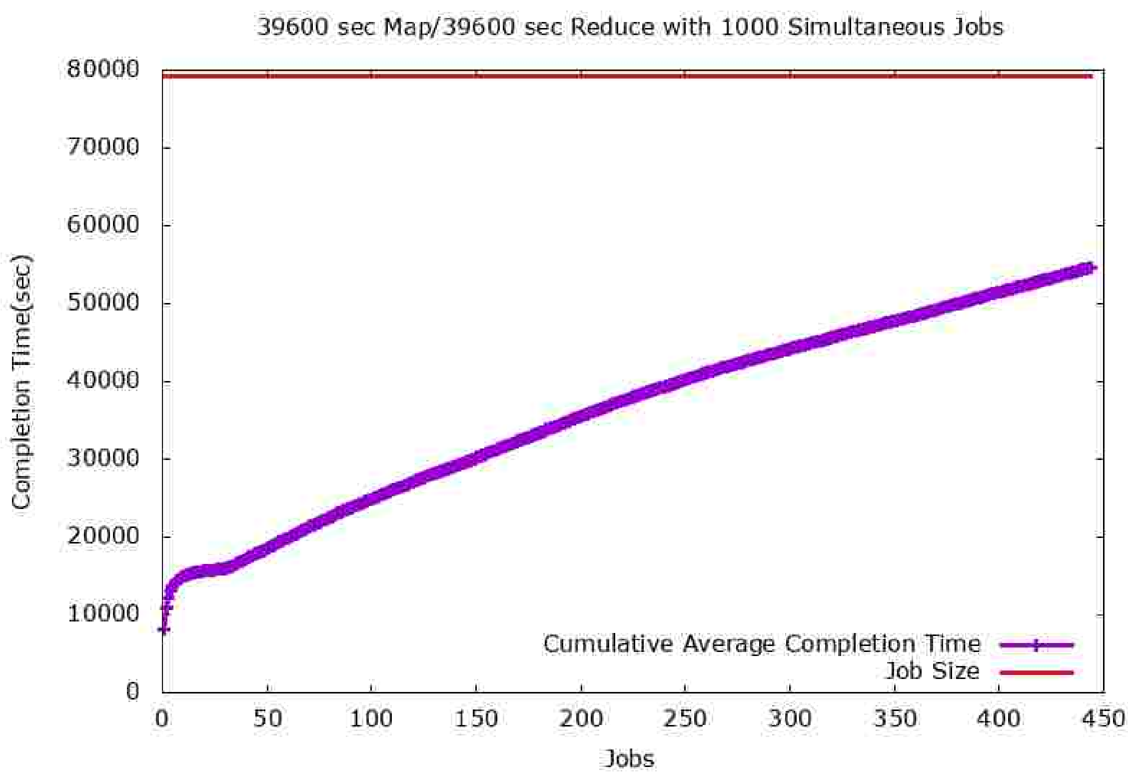
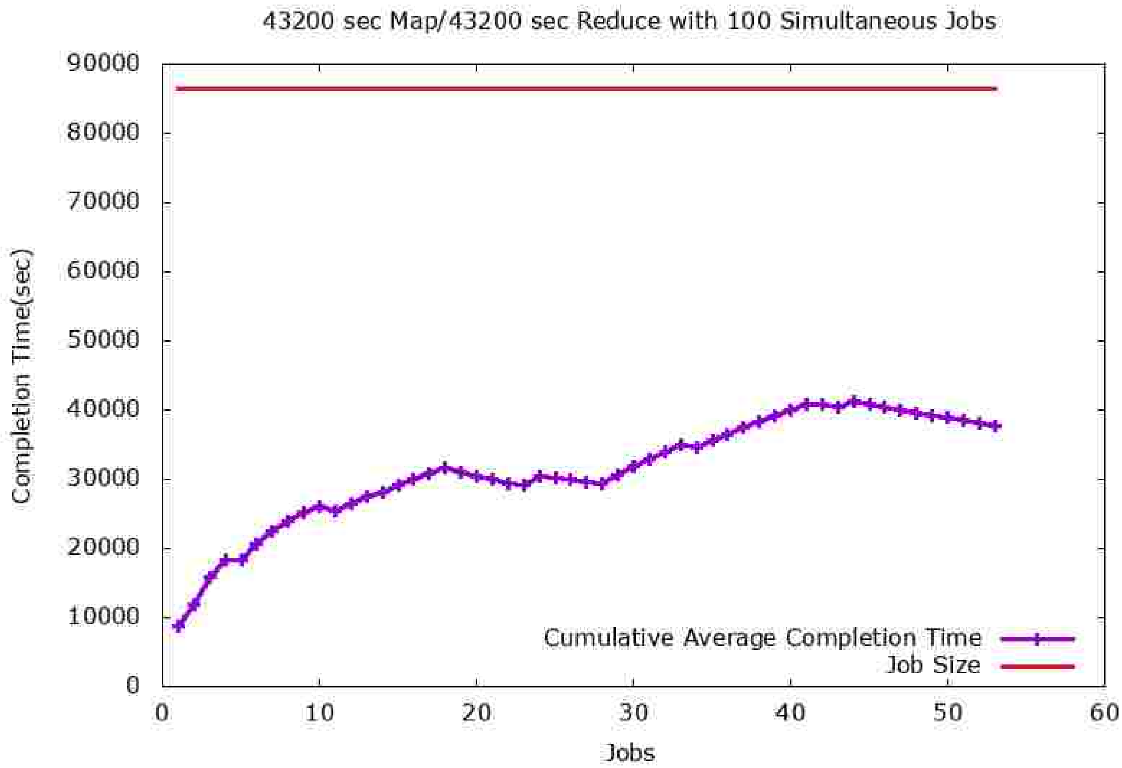


Figure 72: Job Completion Times



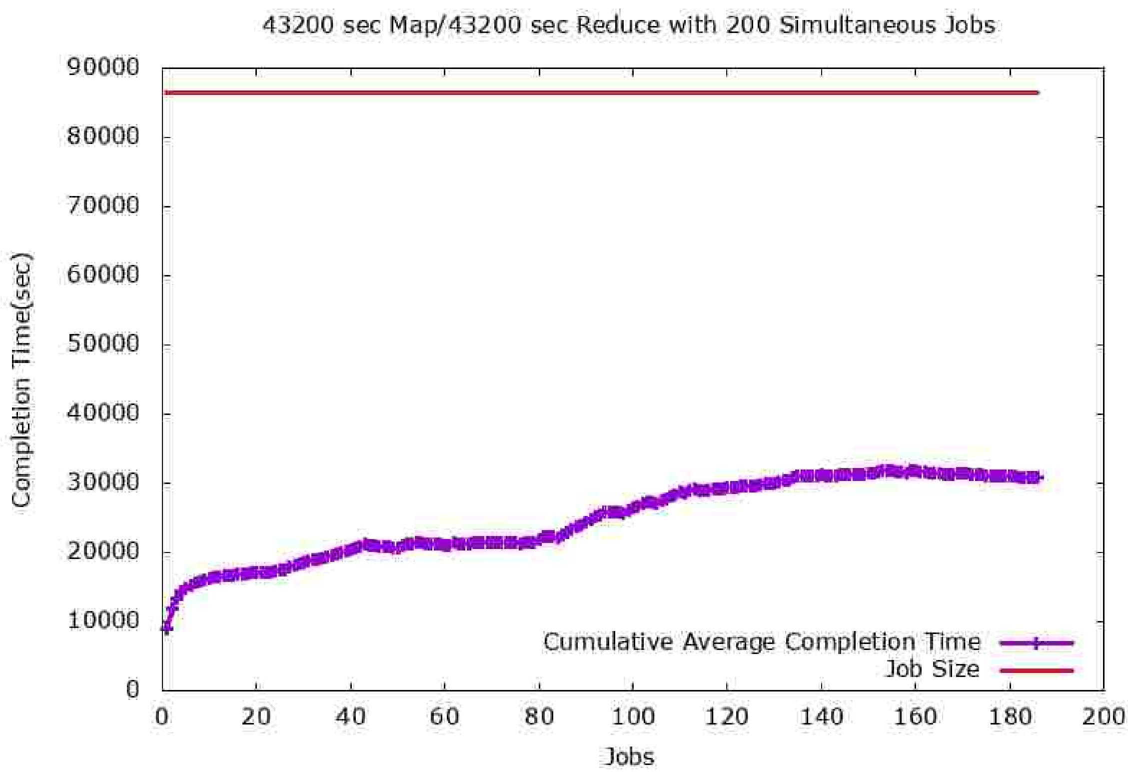


Figure 73: Job Completion Times

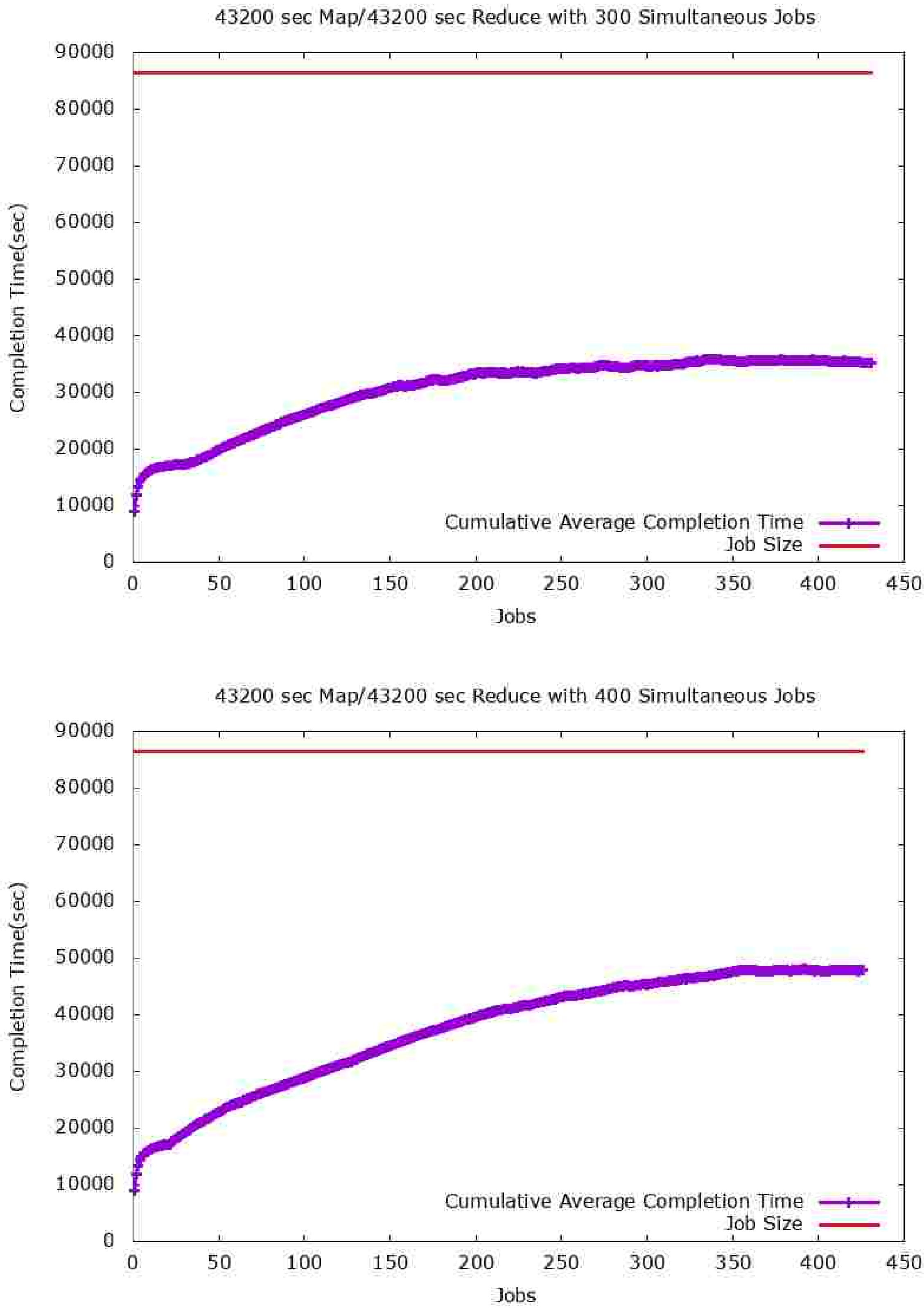
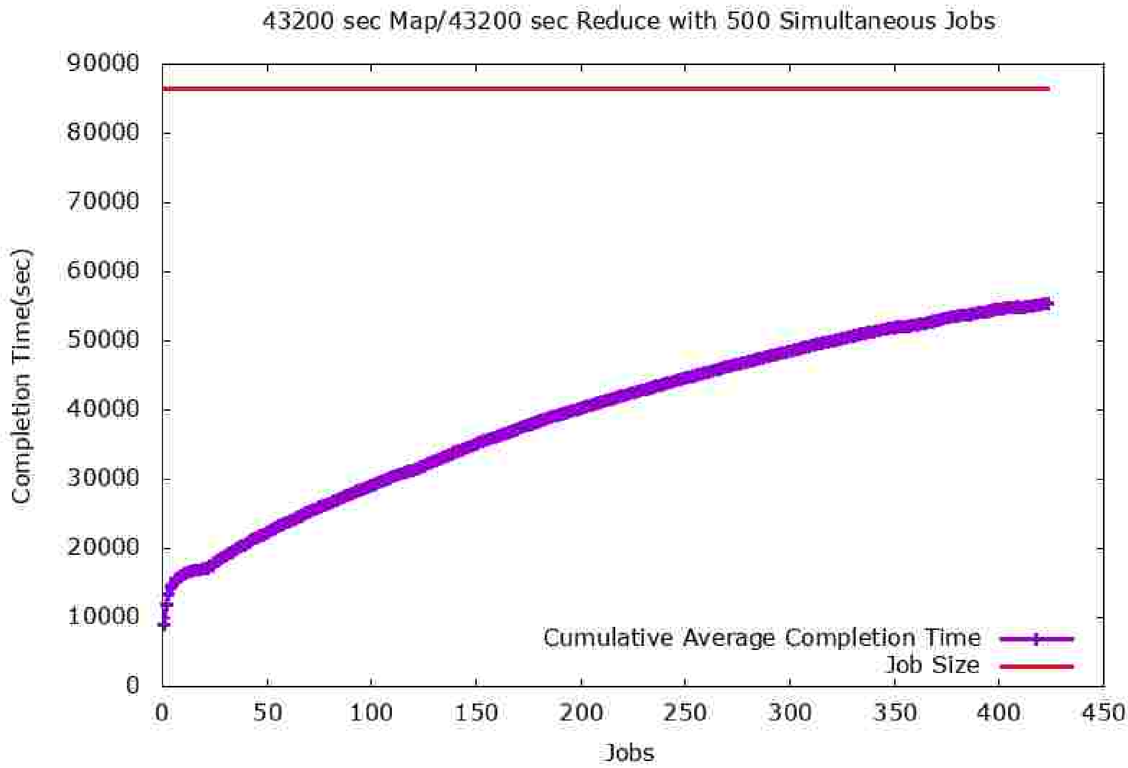


Figure 74: Job Completion Times



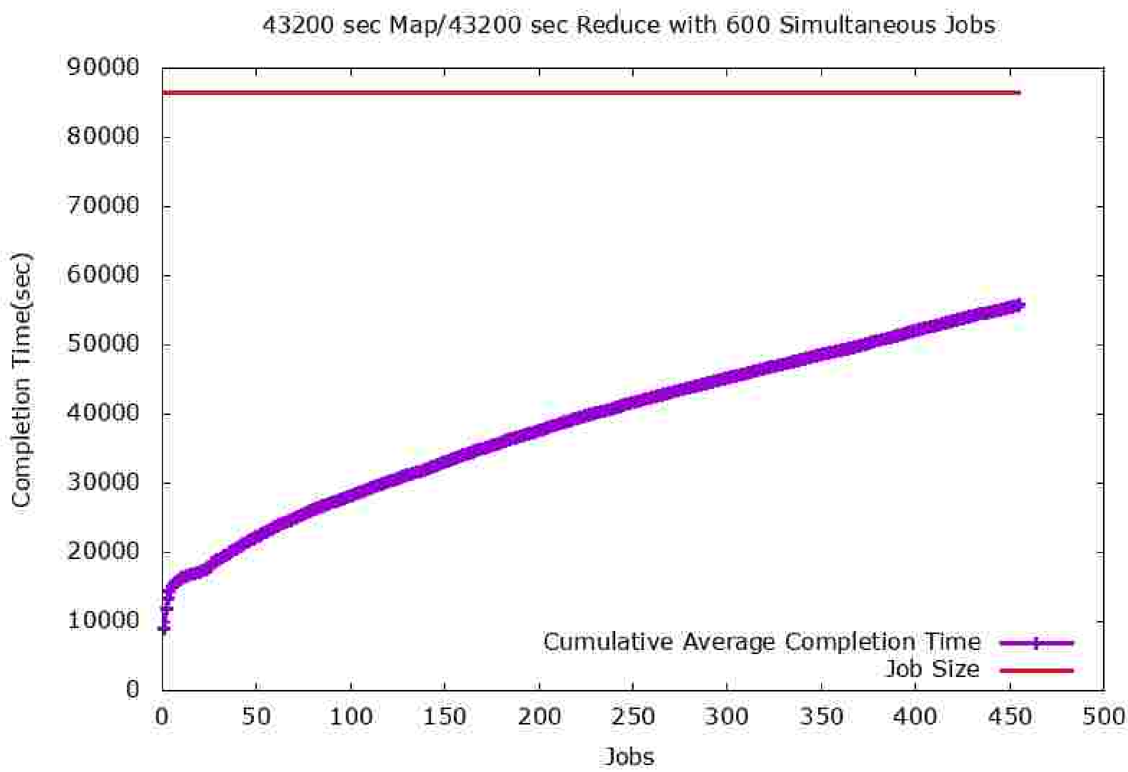
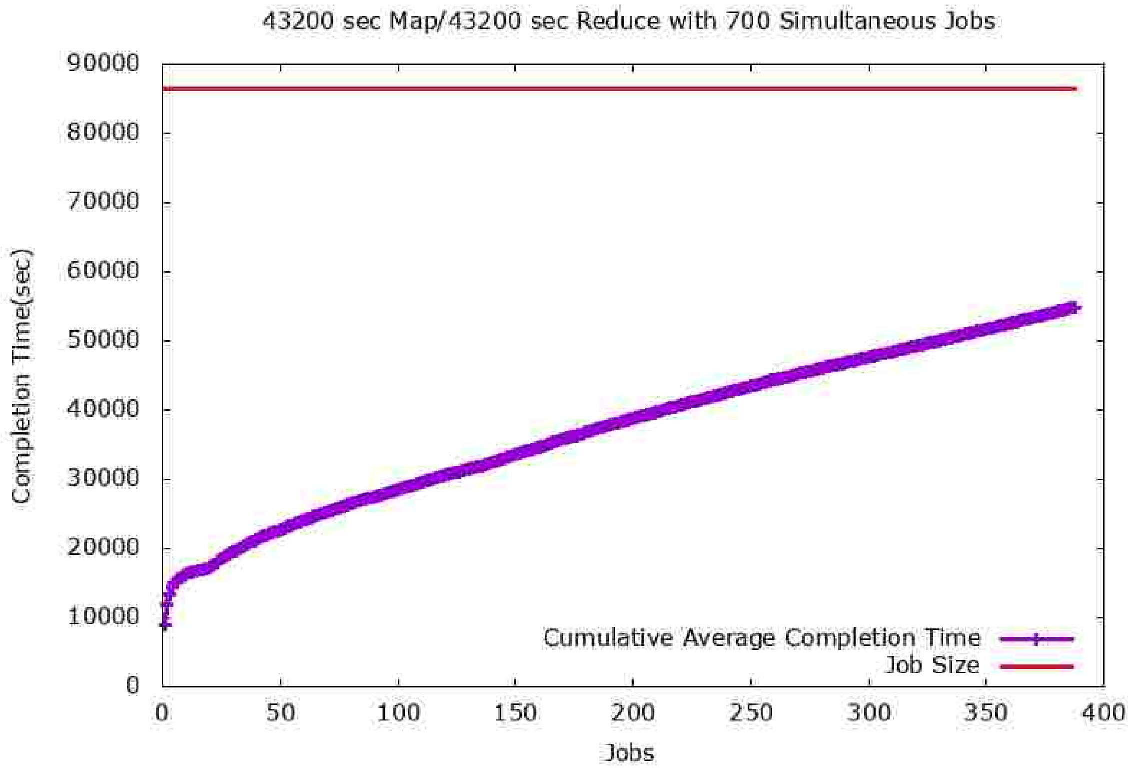


Figure 75: Job Completion Times



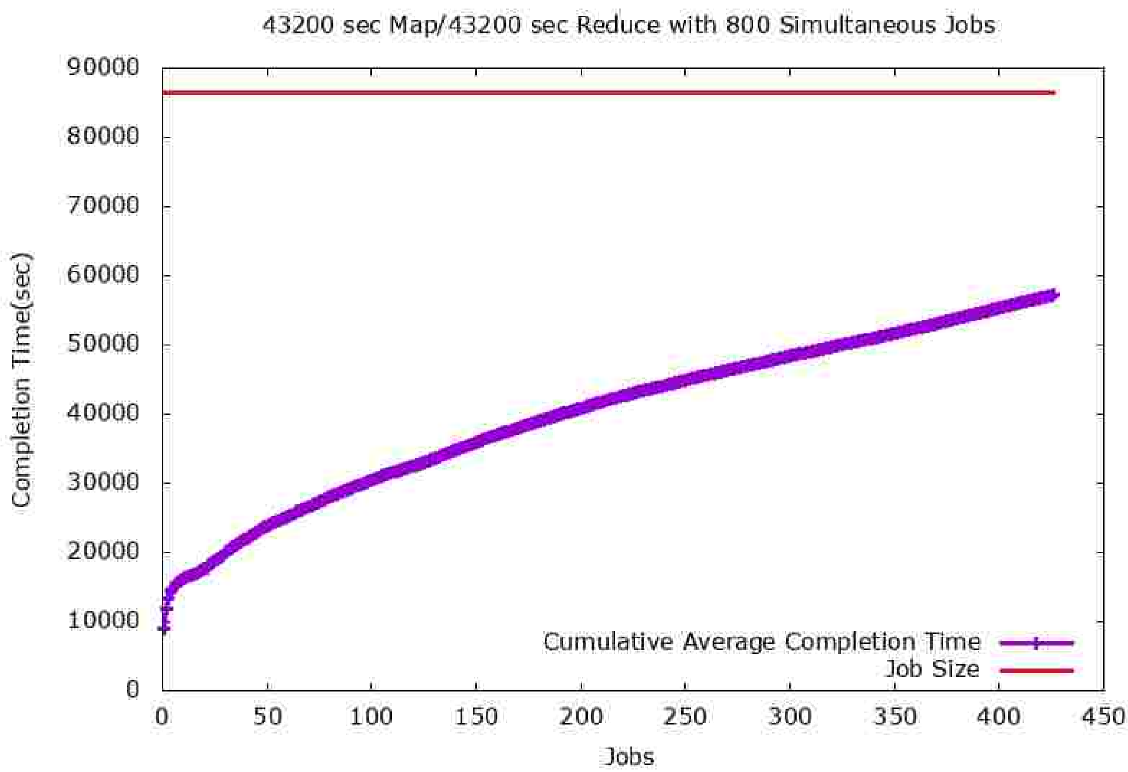


Figure 76: Job Completion Times

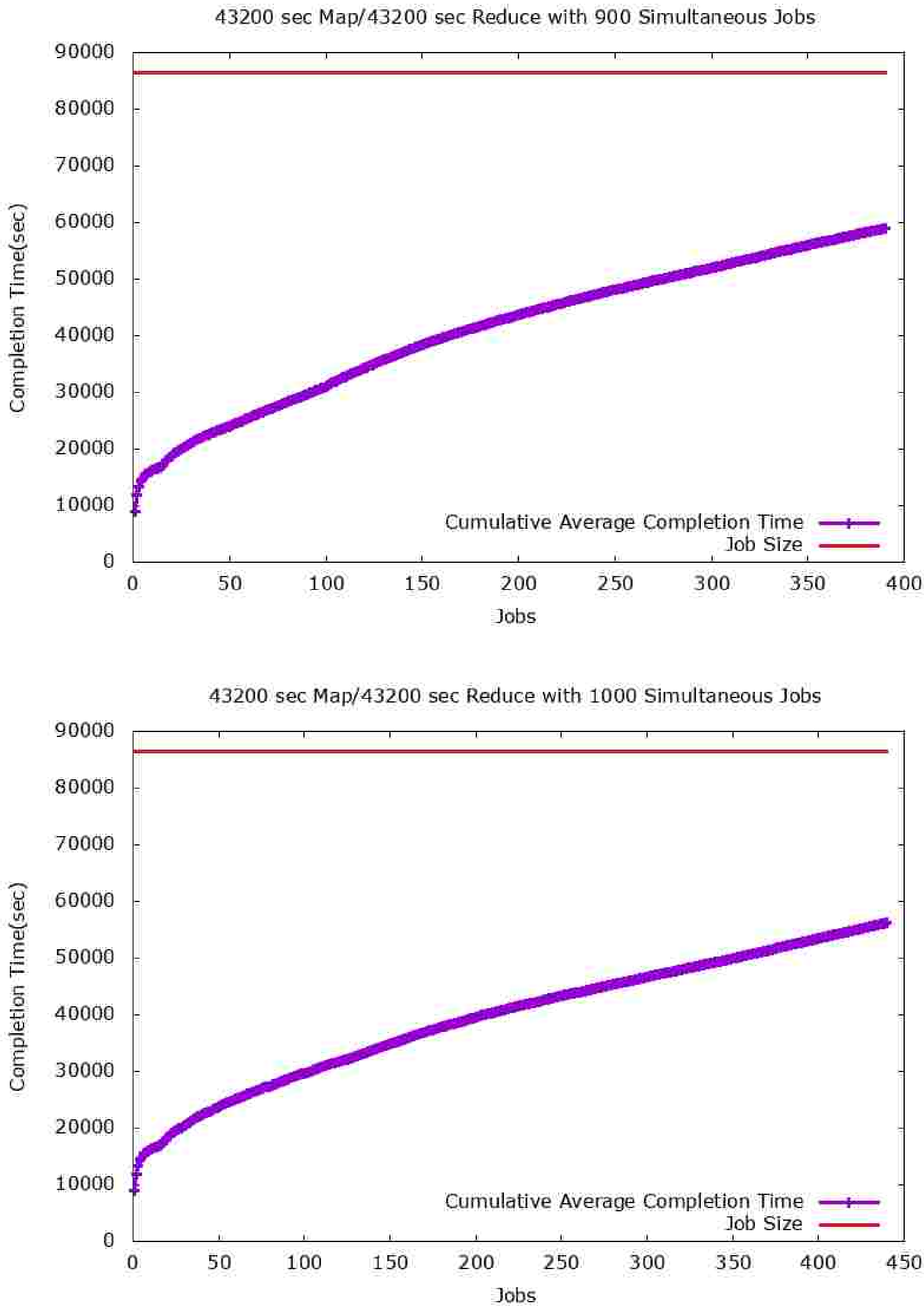
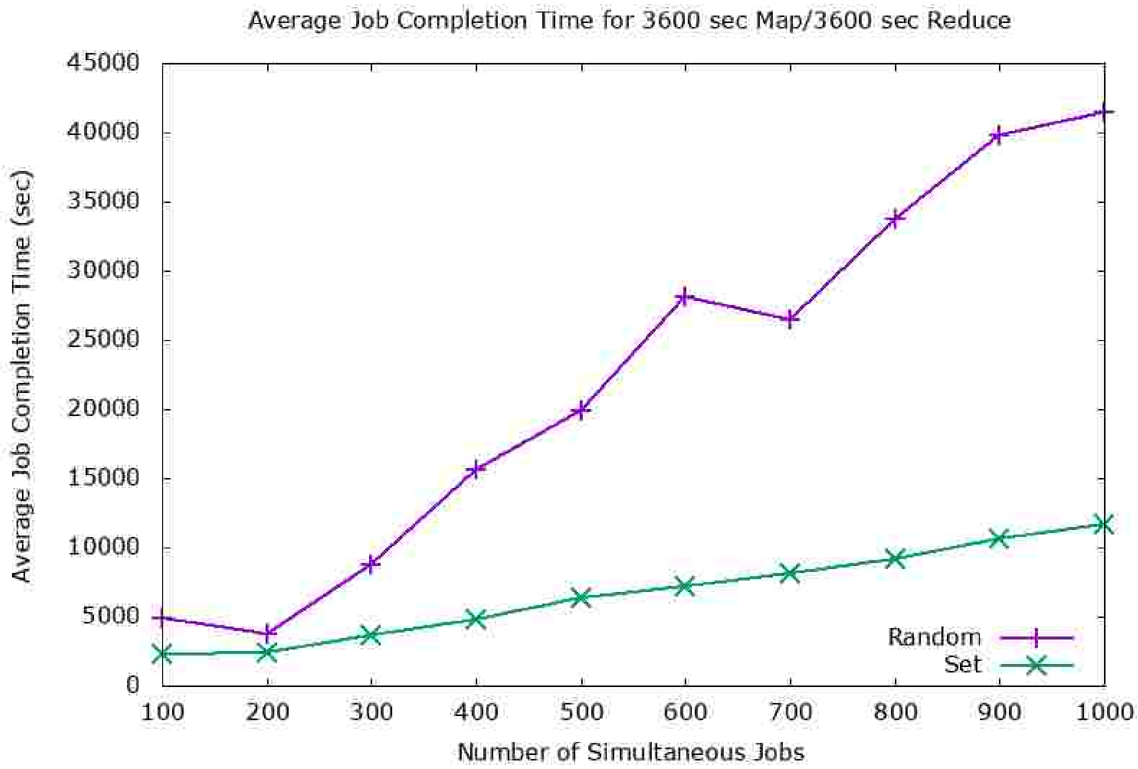


Figure 77: Job Completion Times

APPENDIX B: RANDOM AND SET JOB SIZES



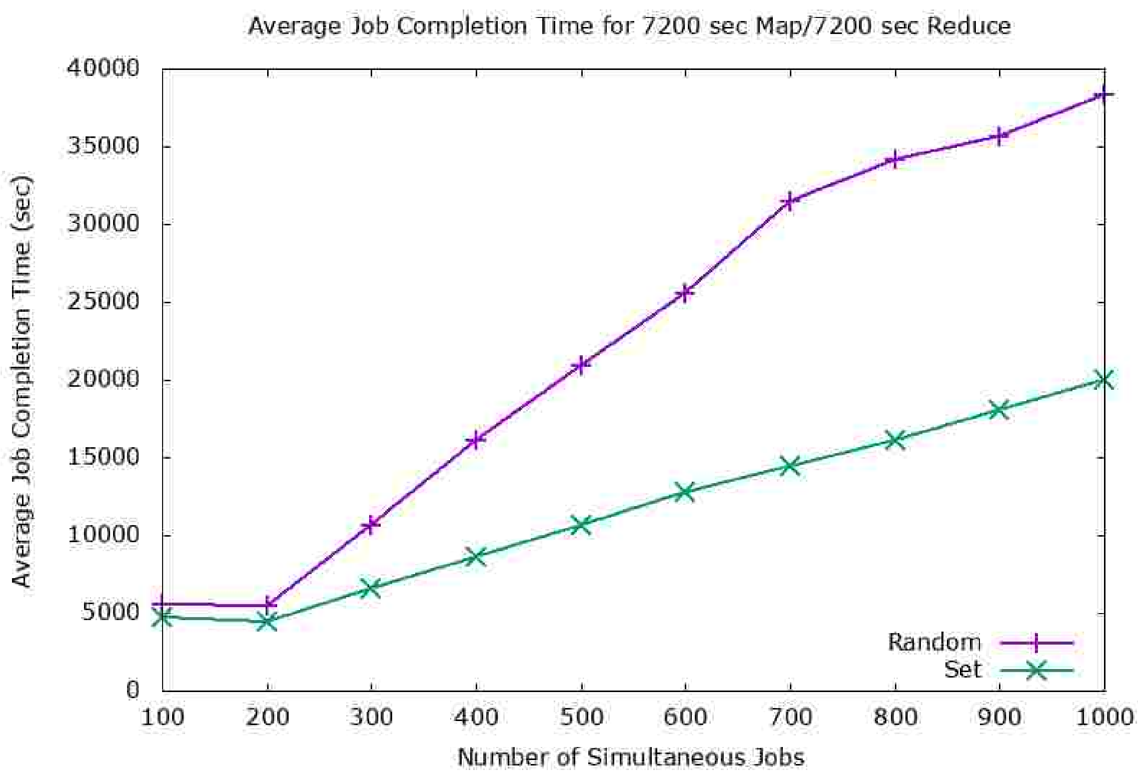
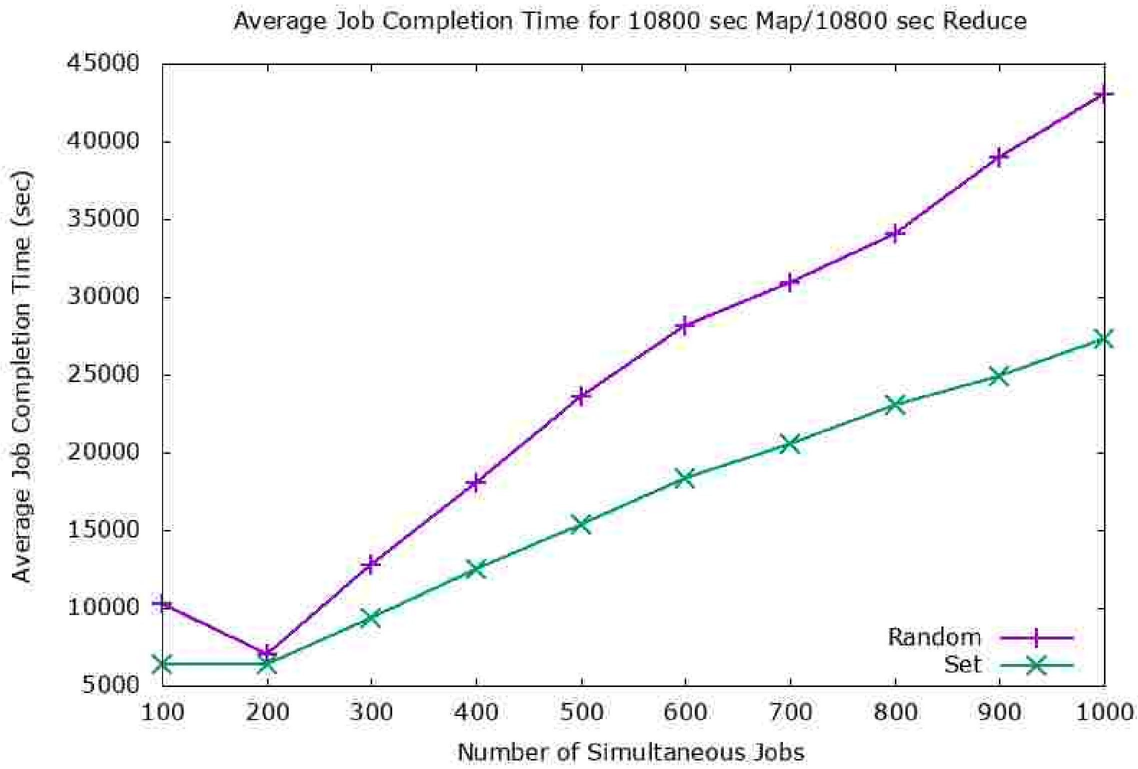


Figure 78: Average Job Completion Times



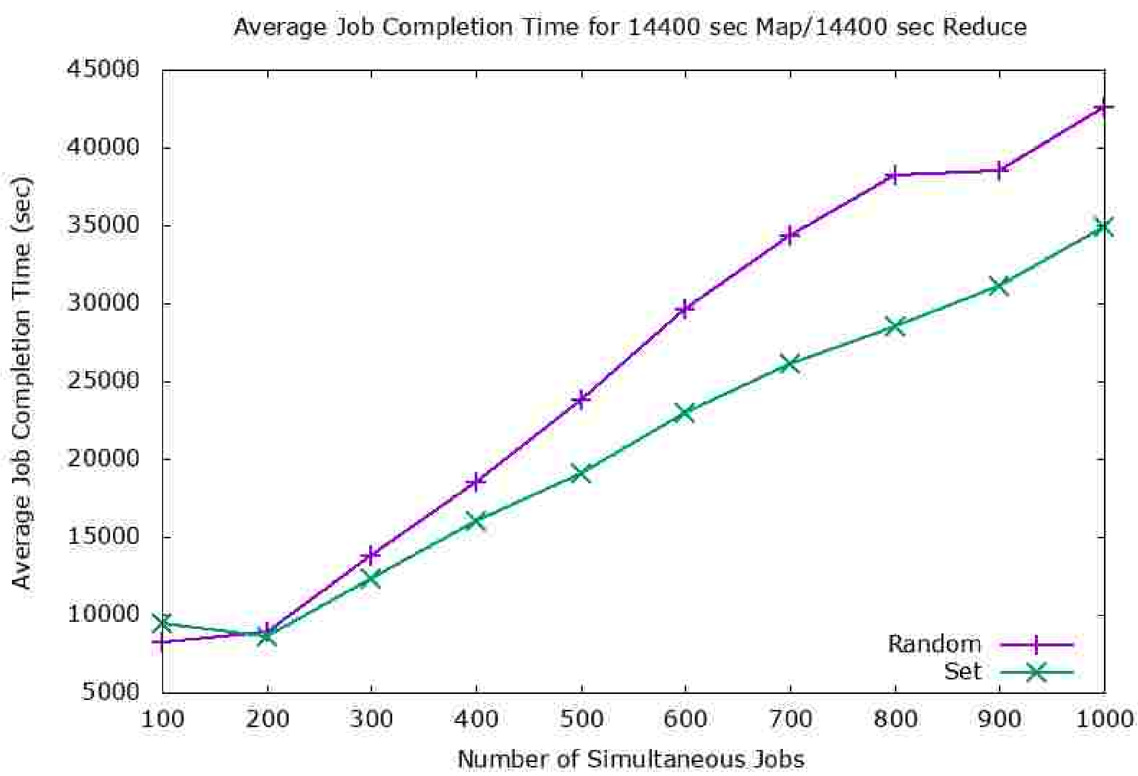
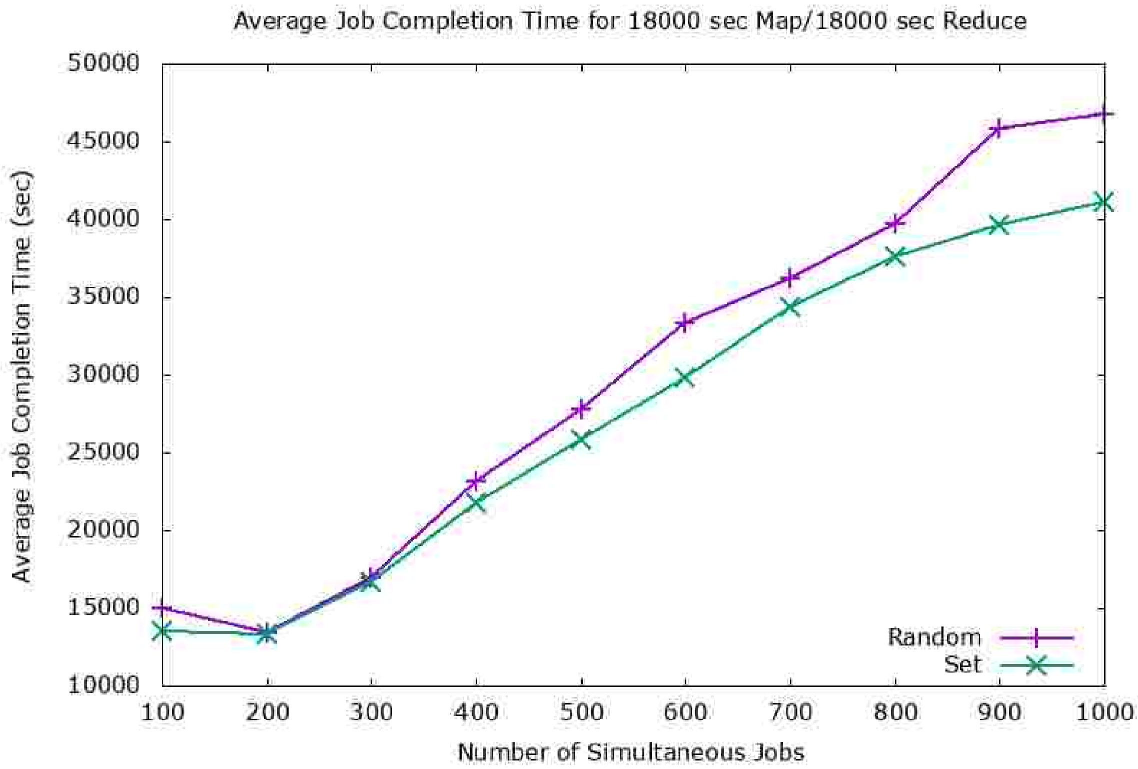


Figure 79: Average Job Completion Times



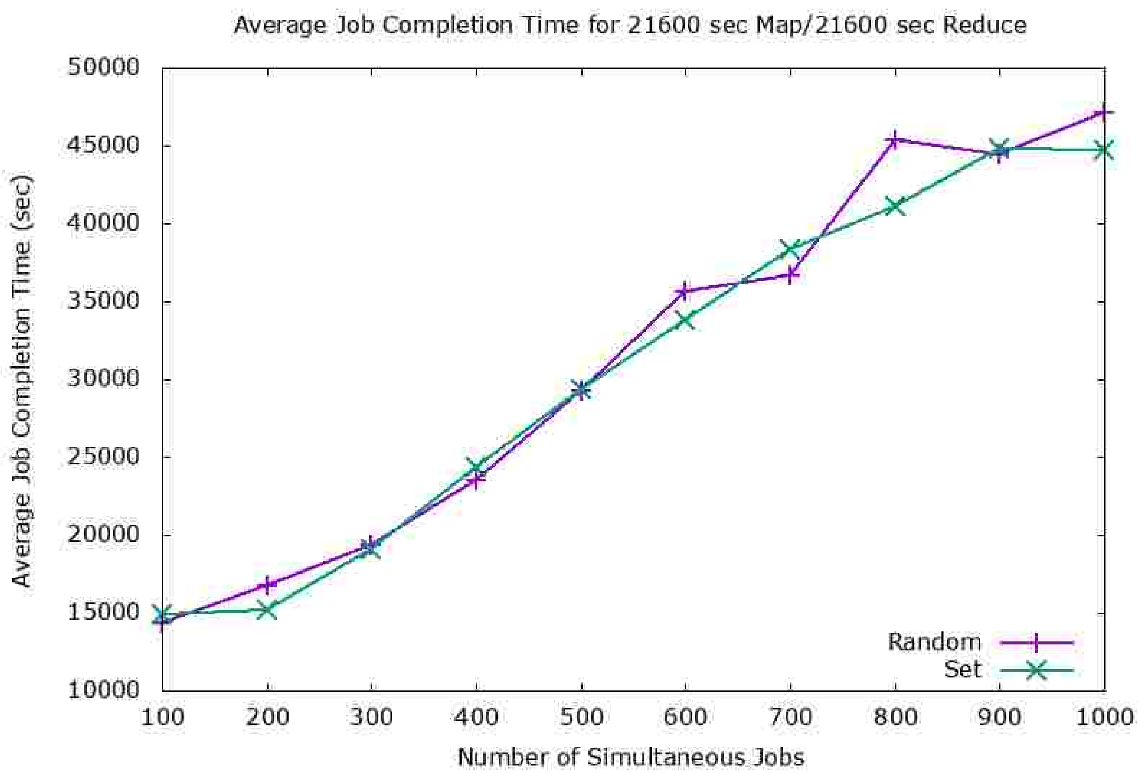


Figure 80: Average Job Completion Times

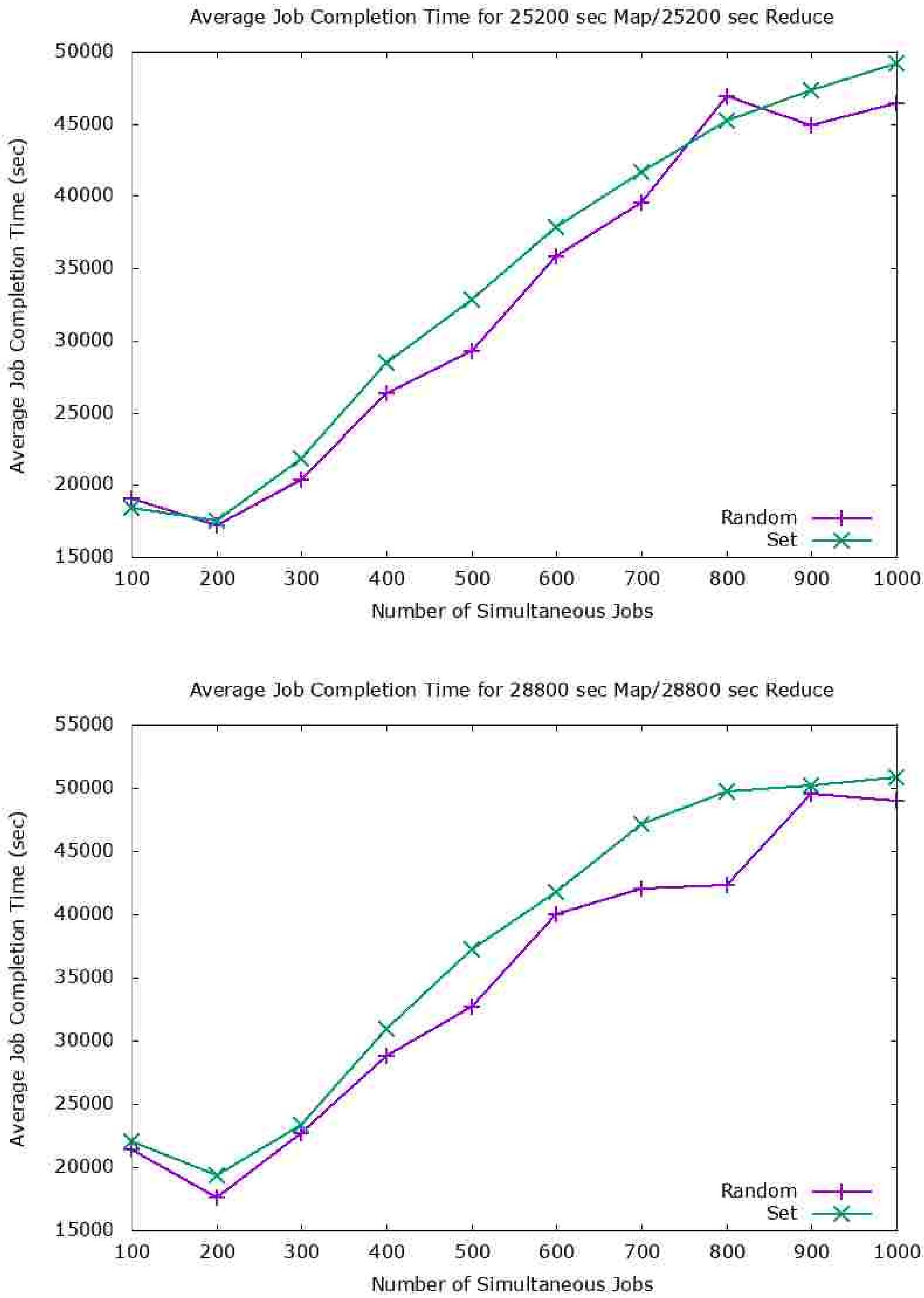
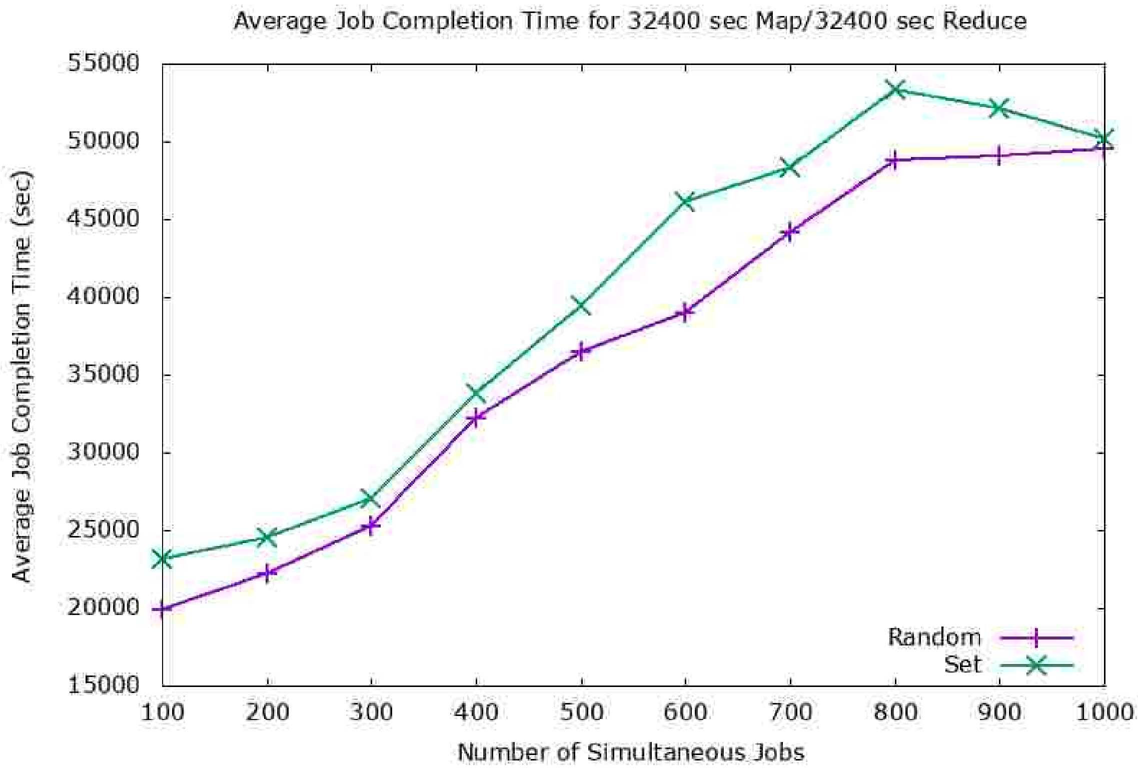


Figure 81: Average Job Completion Times



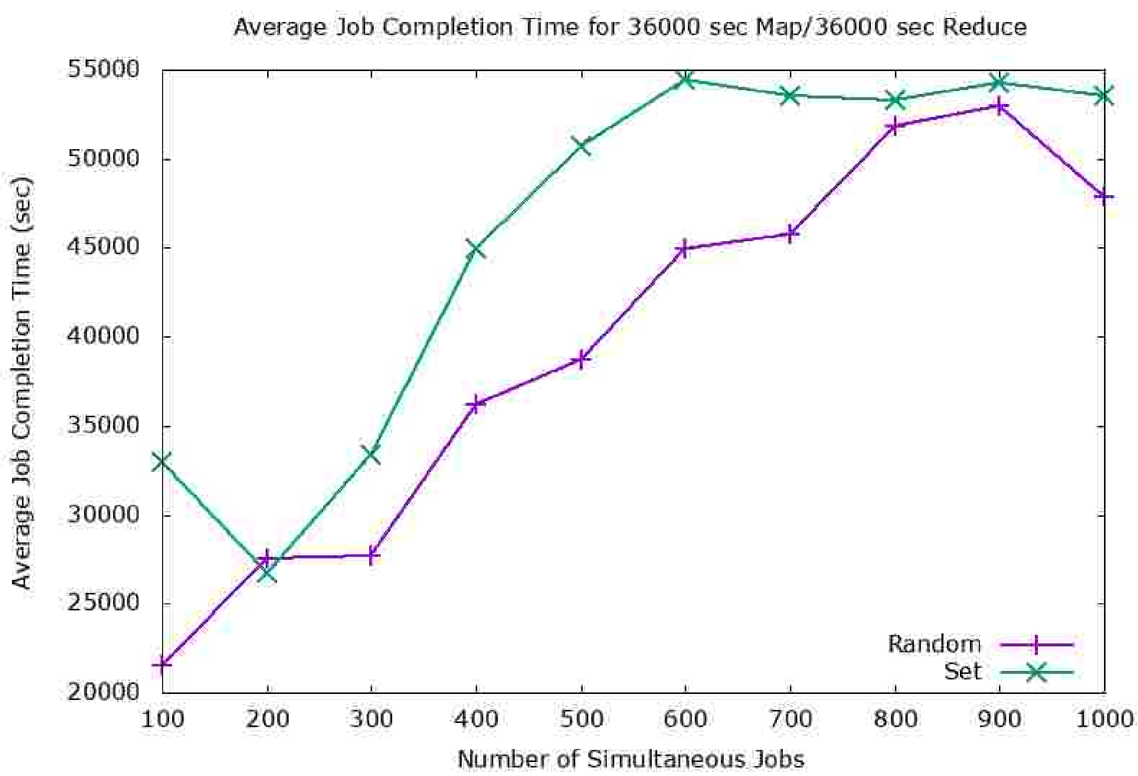
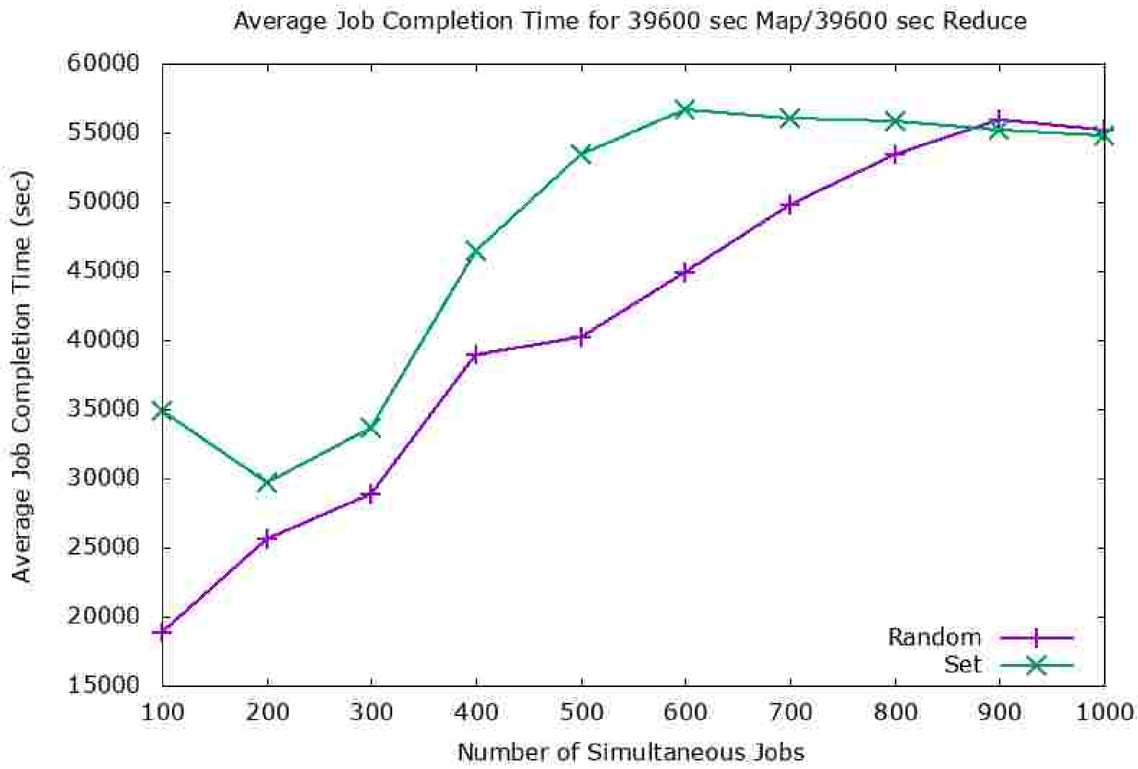


Figure 82: Average Job Completion Times



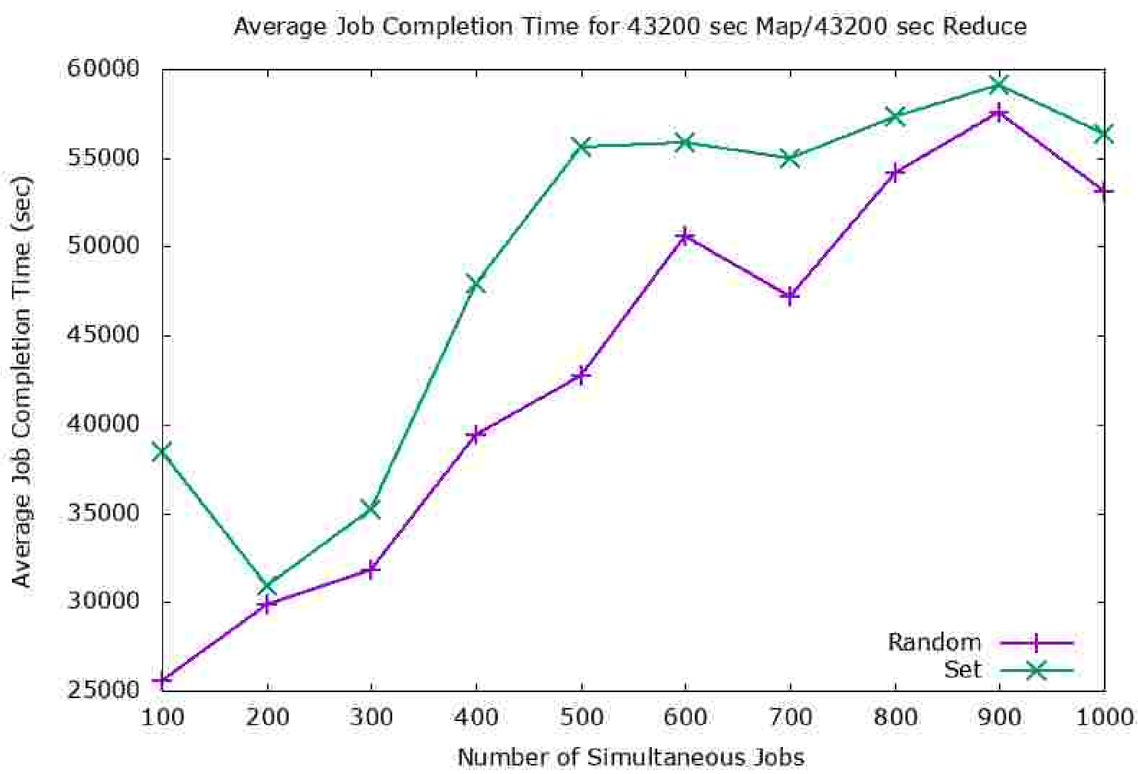


Figure 83: Average Job Completion Times

VITA

Lloyd Decker
Department of Computer Science
Old Dominion University
Norfolk, VA 23529

EDUCATION

Master’s in Information Technology at Virginia Tech, May 2010

Bachelor of Science in Computer Science at Old Dominion University, May 2015

Bachelor of Science in Physics at Virginia Tech, May 1994

EMPLOYMENT

Adjunct Instructor, Old Dominion University, 2016 – present

CS 150 Problem Solving and Programming with C++

CS 170 Computer Architecture I

CS 300T Computers in Society

Operations Research Analyst, Department of the Navy, 2003 – present

Microsoft Word (Office 365) was used to create this document.