NETWORK SECURITY & CRYPTOGRAPHY FINAL RESEARCH PAPER

Parallel & Sequential Jobs.

GROUP MEMBERS

ALIZA ANSARI – B16158009

HIRA KHAN – B16158033

UROOJ FATIMA– B16158101

ABSTRACT

In the research we conducted we emphasize at, we look at a grid system where service is required for both parallel and sequential jobs. Backfilling is used, but a margin of error is added to the prediction of runtime of a job. The effect on the performance of the system will be analyzed and the out comes will be compared with the known optimum runtime case. Two different scheduling methods are considered and the system is evaluated by using a simulation model.

INTRODUCTION

Job scheduling using FCFS algorithm is a challenging task when there’s an unavailability of resources of the grid jobs (gangs) consist of number of parallel tasks. One of the promising solutions of gang scheduling is Backfilling.

Backfilling initiates small jobs before large jobs that require currently unavailable resources. The estimated service times of all jobs must be known during the submission of the jobs or forecasts made by the system base on historic data. However neither of the methods is accurate. This work overcomes such inaccuracies in away that some jobs causing delays will backfill before queued gang can begin processing. The system under study consists of two sites including both parallel and simple jobs. Local jobs (simple jobs) will arrive directly at local scheduler where as grid jobs will have the priority to get dispatched first and foremost, to a site by grid scheduler then further local scheduler will allocate them to a processor. This work extends the study in[1] where power of 2gang sizes were considered.

II. SYSTEM AND WORKLOAD MODELS

There are two sites in the simulation model consisting of a local scheduler with16processors. The work load consists of local and grid jobs& three arrival streams, on eat grid jobs and one inside the local jobs of each both sites. This work extends [1] where gang size was power of 2(2, 4, 8, 16) and follow a uniform distribution with values ranging from 2to 13. Both[1] and[2] results will be compared. The gang sizes were chosen in this study such that average number of tasks per gang in both studies is equal to7.5. The mean arrival time that is inter is the same and locals is exponentially 1/?1,1/?2,1/?3 for locals insite1, site2and gangs respectively where?1, ?2, ?3 are arrival rates for locals insite1andsite2andgangs respectively. We assume that ?1=?2=? and ?3<

Communication held in between the two main sites & GS (gang jobs) relies on message passing there fore its communication time is negligible. However when a grid job is assigned on both sites extra coordination is needed. Figure1shows the configuration of this model.

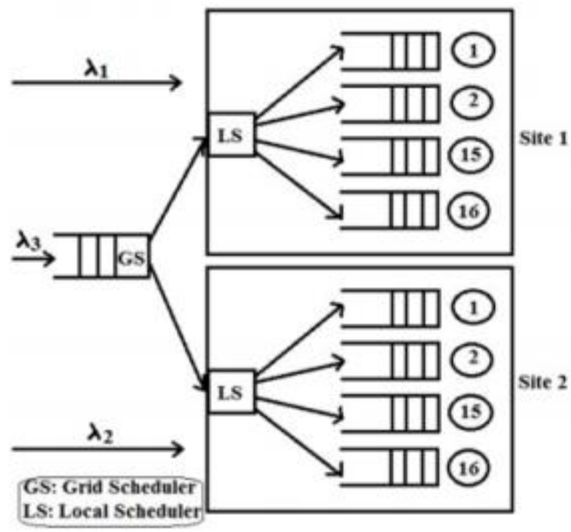

Figure 1. The queuing network model

WORK PROCEEDURE:

III. VARIOUS POLICIES OF SCHEDULING

A. Grid level

1) Allocation policies: The GS will determine whether the arrived gang will be dispatched to a new site or stored in a waiting queue. Two approaches are used for this purpose. Approach1: In this approach first each site is examined for enough idle processors. If this condition is met then GS picks one of the sites and start execution immediately. If no site satisfy this condition then gang is scheduled to the site that has enough empty queues & will start processing when queues become idle. If there is no enough empty queues in both sites then gang waits in GS's waiting queue. Approach2: In this approach the gang tasks can be dispatched to both sites. If approach1conditionsarenot met, then the idle processors of both sites are computed. The gang will be routed to both sites if there are enough idle processors, otherwise it is moved to waiting queue if number of processors in both sites are less than number of gang tasks. 2) Queuing disciplines: It involves twosteps. 1) The tasks of each gang in GS queue are compute they are less or equal to number of empty queues in any site.

The largest gang is chosen among all & the oldest one is chosen if there are more the none largest gangs waiting for the execution.

2) The GS checks for a gang in its queue with a lower or equal number of tasks to the number of idle processors in both sites. The gang found is routed to both sites for processing. If more than one gang is foundthenapproach1strategy is used. B. Local level 1) Allocation policies: The algorithm has twosteps. 1) The local job will be scheduled to shortest queue if all processors are busy. 2) The local job will be scheduled to that processor where execution start immediately if there are queues with idle processors and if the following condition is met: Service Time? Elapsed Time+ T

2) Queuing disciplines: If the processor is idle and queue is empty when local job enters a queue, the job execution will start immediately. In FCFS gangs will delay local jobs services that is why back filling is used in which gang is waiting for service so enough processors are available for local jobs. The local job can start execution before a gang waiting in the queue if following condition is met:

Service Time? Elapsed Time+ T Where service time is the time of run of local job and elapsed time is that time which is remaining until the gang start execution. IV. PERFORMANCE EVALUATION A. Performance metrics To evaluate system performance we compute two different response times: the average response time of the gangs and the average the time of responding by the local jobs. The metric used to measure the response time of a job if m is the number of total processed job is given as:

$$RT = \frac{1}{m}\sum_{j=1}^{m} rj.$$

The response timer j of a job j is the time interval between job arrival and completion. The metric used to measure delay of a job against its actual run time is shown below

$$Sj=\frac{rj}{ej}$$

Where sj is slow down of a job and ej is service time. Averages low down is given as:

$$SLD = \frac{1}{m}\sum_{j=1}^{m} s_j$$

The response time and slow down of parallel jobs is defined by below metrics. Whenever it comes to parallel job search gang need to be weighted with its size.

- The average weighted response time WRT

$$WRT = \frac{\sum_{j=1}^{m} p(x_j) r_j}{\sum_{j=1}^{m} p(x_j)}$$

- The average weighted slow down WSLD:

$$WSLD = \frac{\sum_{j=1}^{m} p(x_j) s_j}{\sum_{j=1}^{m} p(x_j)}$$

Where P(x) is supposed to be that number of the processor required by a particular job. Table I contains the parameters used in simulation computations . B. Input parameters The arrival rate for gang is static for all ten experiments and equal to0.5. The mean interval time for gangis1/?3=2. Number of users and amount of processors will change arrival rate of local jobs. Mean interval timeis1/?3 =0.08, 0.1and 0.12for locals in simulation experiments. These mean interval time correspond to arrival rate?=12.5, 10and8.3 respectively. The threshold and overhead of a gang is T=0and 10% respectively. Threshold is the maximum time a back filling local cande lay a gang. The error percentage of runtime calculation is t=0, 10and30

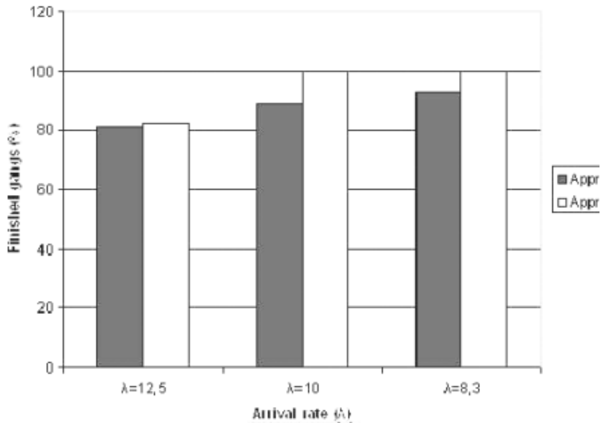

Figure 3. Percentage of finished gangs versus λ for power of 2 gang sizes

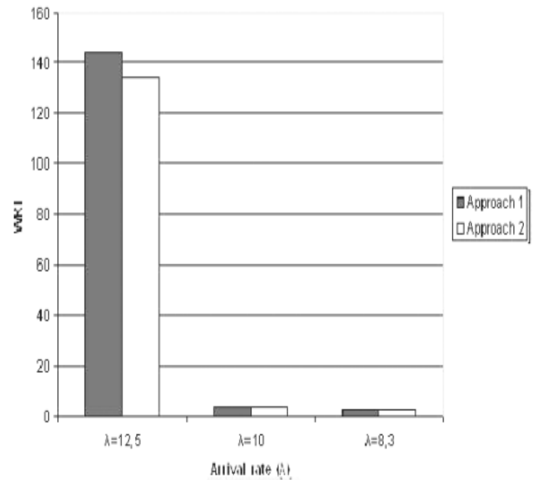

Figure 4. WRT versus λ for uniform gang sizes

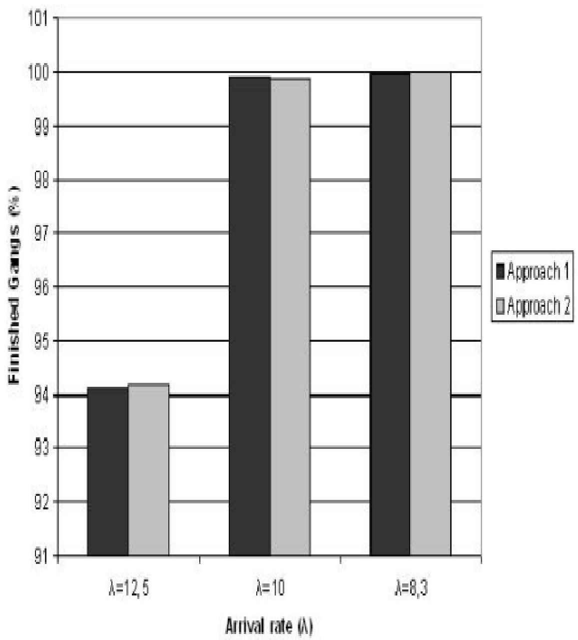

Figure 2. Percentage of finished gangs versus λ for uniform gang sizes

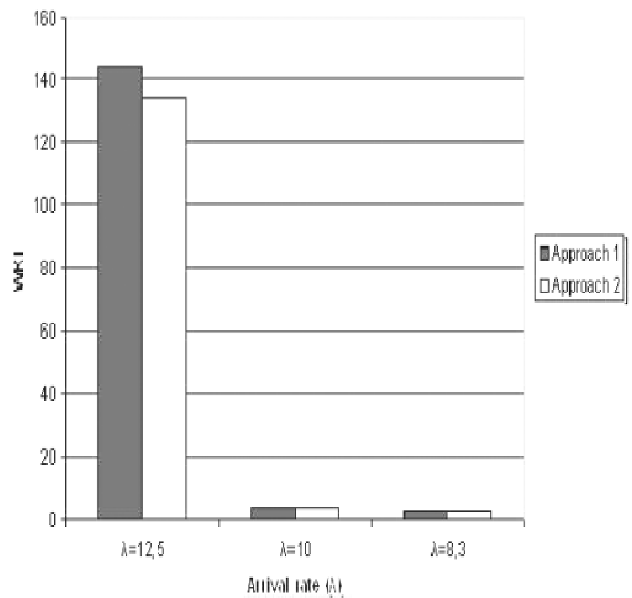

Figure 4. WRT versus λ for uniform gang sizes

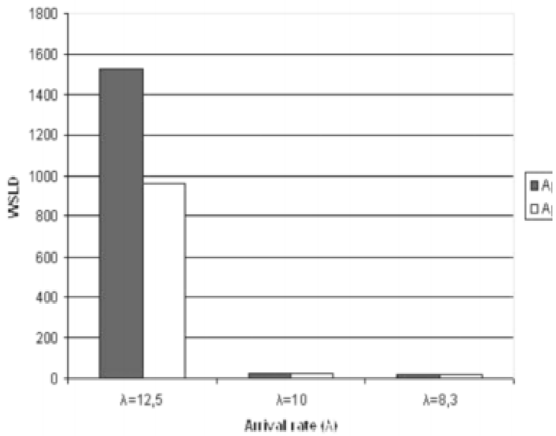

Figure 5. WSLD versus λ for uniform gang sizes

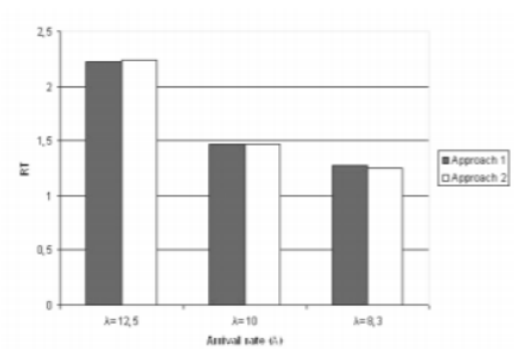

Figure 6. RT versus λ for uniform gang sizes

C. Simulation experiments

Figure2haveahigher completion of gangs for uniform gang sizes and in both approaches; Approach1and2, all gangs complete execution for medium and high workload. In Figure3approach2produce much better results for gang size of powers of two as compared to uniform gang sizes infigure2. The result for Approach1inboth Figure2and3 areal most same. Approach2results in the completion of 10% moregangthanapproach1.

Figure4and5shows that when work load is high approach 2results in lower WRT and WSLD. The reason behind is that approach2 minimizes waiting time as gangs tasks can be routed to both sites for executionopposetoapproach1.

Figure6and7shows that for higher workloads, RT and SLD for local jobs are bit higher for approach2comparedto Approach1 while gang sizes are uniformed.

Table I
MEAN PROCESSOR UTILIZATION

	Approach1	Approach2
λ = 12.5	0.87999	0.88304
λ = 10	0.7375	0.73791
λ = 8.3	0.63484	0.6390

Table I depicts that higher workloads result in higher mean utilization. As shown above simulated experiments both approaches produce similar results when uniform gang sizes are used in low / medium workloads. Approach2results in slightly higher system utilization as compared to Approach1 as in high work load it complete a larger percentage of gangs with low WRT and WSLD as it assign gangs to both sites. Figure8and9shows that change in WRT for different values of error is very small for some work load. This is because of least need of backfilling.

Figure10and11shows similar result for WSLD.

Figure12and13shows that error in local jobs runtime prediction has little or no effect on their mean response time.

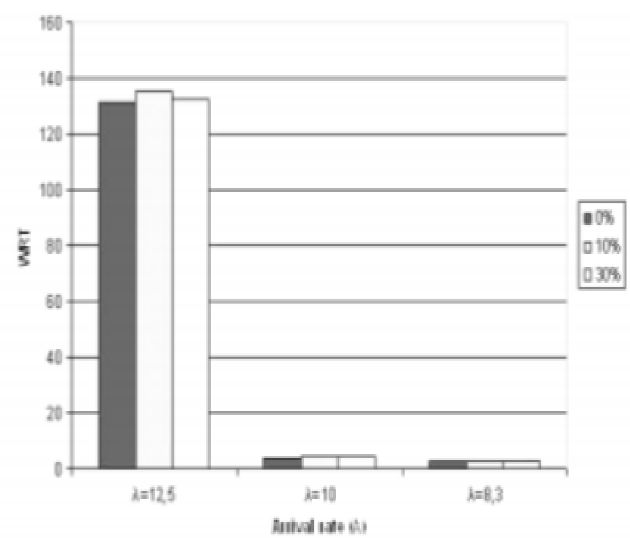

Figure 8. WRT versus λ for uniform gang sizes using

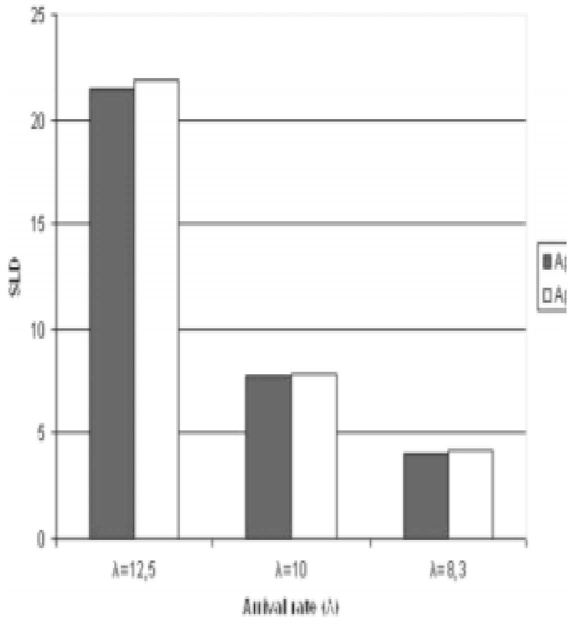

Figure 7. SLD versus λ for uniform gang sizes

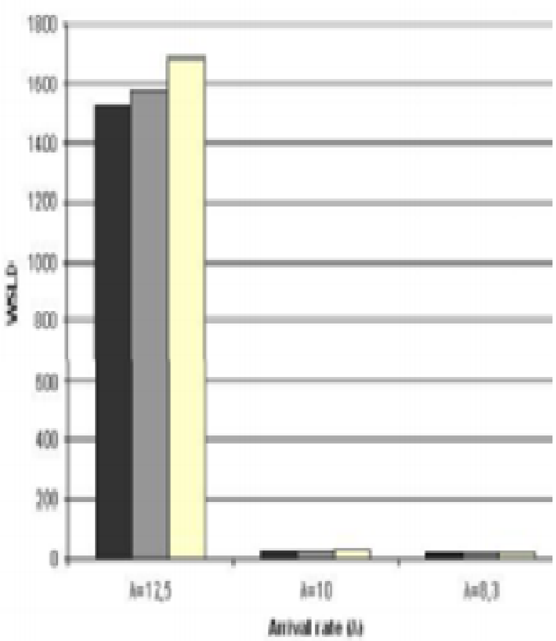

Figure 10. WSLD versus λ for uniform gang sizes using approach 1

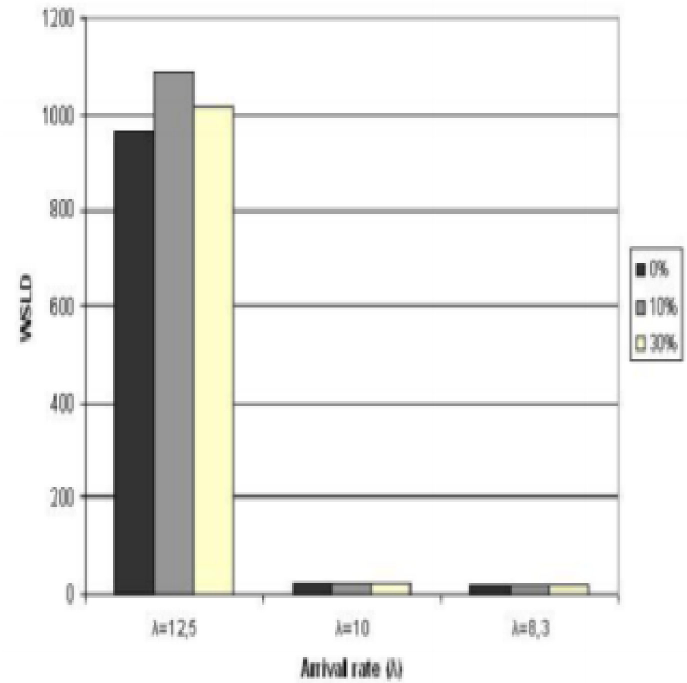

Figure 11. WSLD versus λ for uniform gang sizes using approach 2

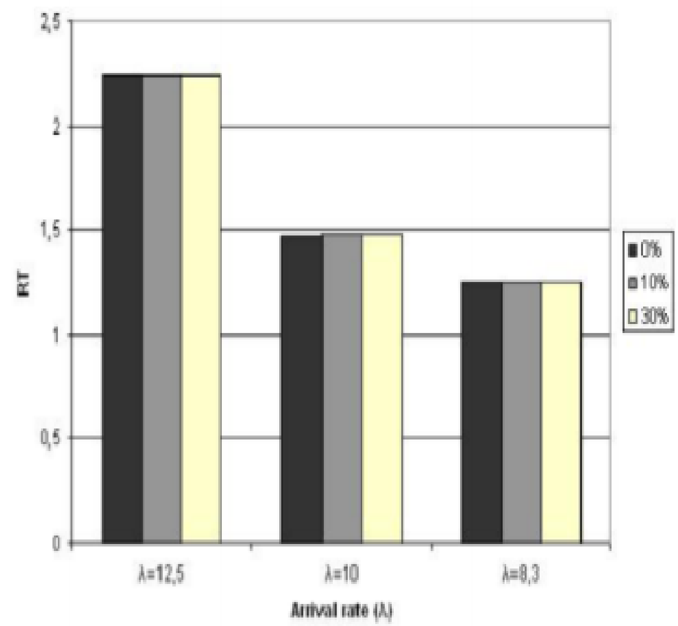

Figure 13. RT versus λ for uniform gang sizes using approach 2

CONCLUSION:

This research studies and implements two scheduling techniques in a system where grid and local jobs compete for same resources. Their performance is evaluated and back filing is used to avoid fragmentation. Two experiments were conducted using a simulation model under various workloads. Results show that back filling is an excellent gang scheduling technique for this particular system. This study could be extended in several ways. For example, we assumed homogeneous sites, when heterogeneity is one of the main characteristics of the grid. Sites with different number of processors could be considered.

REFERNCES:

Hasasn Rajaei, COMPARISON OF BACKFILLING ALGORITHMSFOR JOBSCHEDULING IN DISTRIBUTED MEMORYPARALLEL SYSTEM?, Computer Science Department at Bowling Green State University. 2006-737 https://en.wikipedia.org/wiki/Gang_scheduling. Eitan Frachtenberg , Dror G. Feitelson , Juan Fernandez , and Fabrizio Petrini, ? Parallel Job Scheduling Under Dynamic Workloads?, 1CCS-3 Modeling, Algorithms, and Informatics Group Computer and Computational Sciences (CCS) Division Los Alamos National Laboratory (LANL), Science Foundation (grant no. 219/99).