Clairvoyant Site Allocation of Jobs with Highly Variable Service Demands in a Computational Grid

Stylianos Zikos and Helen Karatza

Department of Informatics

Aristotle University of Thessaloniki

54124 Thessaloniki, Greece

PMEO 2010

Atlanta, USA

Outline

- In this paper we evaluate performance of three different site allocation policies in a 2-level computational grid with heterogeneous sites.

- A simulation model is used to evaluate performance in terms of the response time and slowdown, under medium and high load.

Structure of the presentation

- Introduction
- System and workload models
- Scheduling policies
- Performance metrics
- Experimental setup
- Experimental results
- Conclusions and future directions

Introduction

- Computational grids are very common and useful nowadays.
- Efficient scheduling of jobs is essential in a grid due to the heterogeneous distributed resources and the number of users involved.
- In general, scheduling algorithms have to deal with resource assignment and queue ordering. In this paper we focus on the resource assignment part.

Introduction

- A scheduling algorithm can be classified into clairvoyant or nonclairvoyant with regard to knowledge about characteristics of jobs.
- A clairvoyant scheduling algorithm may use information of jobs' characteristics such as service time, whereas a nonclairvoyant algorithm assumes nothing about the characteristics of the jobs.
- In this paper we assume that job service demands are known to schedulers.

Introduction

- The present paper focuses on site allocation policies in a 2-level heterogeneous grid, where job service demands are highly variable following the Bounded Pareto distribution.

System and Workload Models

- An open queueing network model of a 2-level grid with heterogeneous sites is considered.
- There are totally four sites.
- The Grid Scheduler (GS) dispatches submitted jobs to the geographically distributed sites.
- Each site consists of a set of processors and a Local Scheduler (LS).
- LS and processors are connected via a high speed local network.

System and Workload Models

- When a job arrives, LS routes the job to a processor, according to a policy.
- There are totally **80** processors in the model, with each site consisting of different number of processors.

  *Site #1 → **8** processors*

  *Site #2 → **16** processors*

  *Site #3 → **24** processors*

  *Site #4 → **32** processors*

- All processors have the same computational power.

System and Workload Models

- There are no jobs locally submitted.
- Jobs are atomic, as they can not be further divided into tasks that can be executed in parallel.
- Jobs are nonpreemptable: their execution on a processor can not be suspended until completion.
- Jobs are clairvoyant as their service demand times are known to schedulers.

System and Workload Models

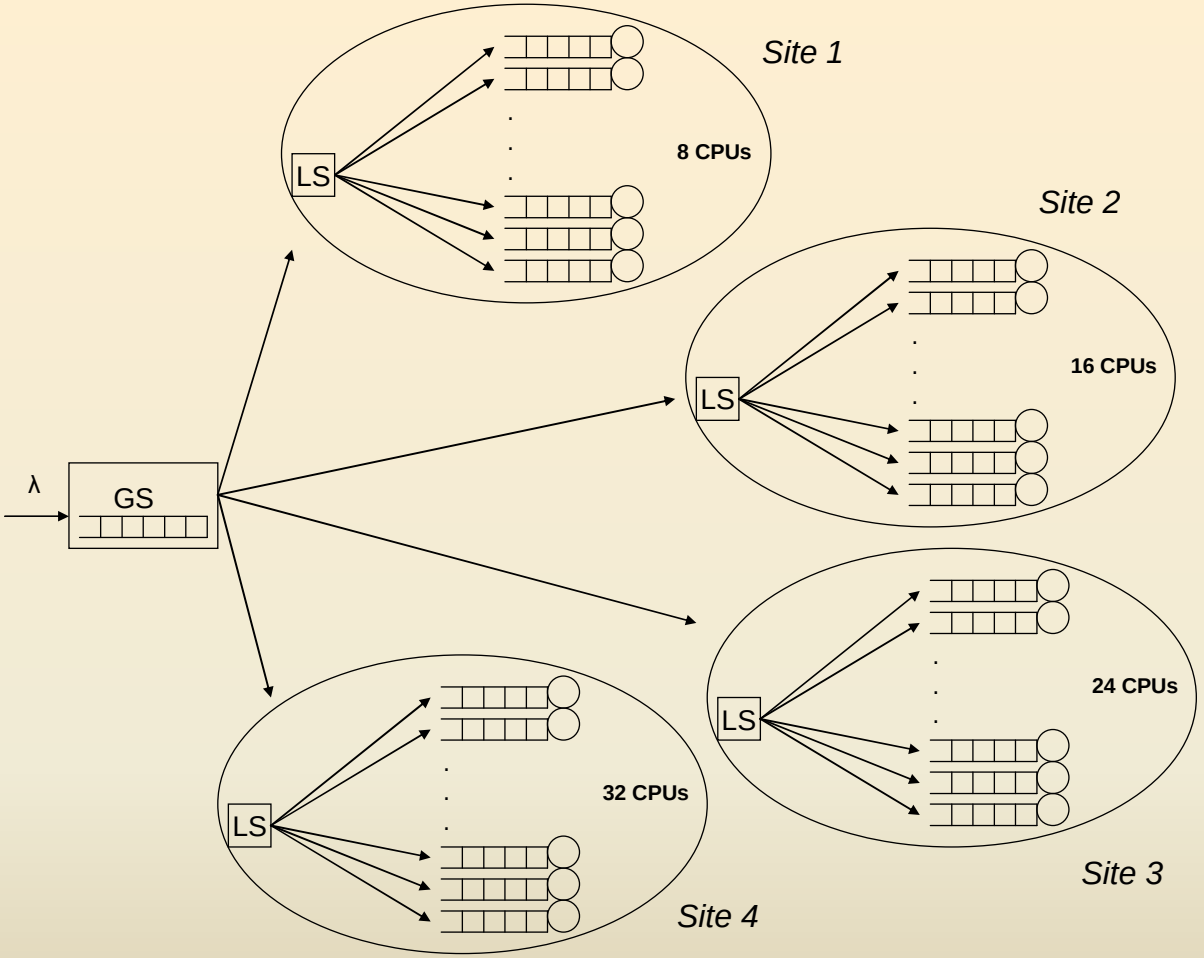

Figure 1. The queueing network model

System and Workload Models

- The inter-arrival times of jobs are exponential random variables with mean of $1/\lambda$.
- The Bounded Pareto distribution is used, in order to generate highly variable job service demand times :

    High number of service demands that are very small compared to the mean service time, and few service demands that are much larger than the mean service time.
- The Bounded Pareto distribution is characterized by the three following parameters:

    α (shape parameter – determines the level of variability)

    L (Lowest bound: minimum service demand)

    H (Highest bound: maximum service demand)

Site allocation policies

- The applied policy determines the way a site is selected for a job.
- *Random*
  - GS instantly routes a job to a randomly selected site.
  - It uses static site information to create approximate selection probabilities about each site.
  - A site's selection probability is proportional to its computational capability.
  - GS does not exploit the knowledge about each job's service demand.

Site allocation policies

- *Deferred*
  - Based on dynamic site load information that the GS periodically receives from the LSs.
  - The information is available to GS at every specified time interval that we call Allocation Interval (A_I).
  - The GS dispatches all jobs in the queue at the end of each A_I.
  - For each job, the site with the minimum load is selected.
  - We define load as the average remaining work per processor in a site.
  - The total remaining work for a site is divided by the number of processors in the site, in order to calculate the average remaining work per processor.

Site allocation policies

- *Size-Based Deferred (SB-Deferred)*
  - We introduce this policy which combines the two policies presented above, the Random and the Deferred.
  - GS uses the Service Demand Threshold (SDT) parameter to apply either the Random or the Deferred policy.
  - If a job's service demand is larger than SDT, then the job is considered as demanding, its scheduling is deferred and it is stored in GS's queue. Otherwise, a site is selected for the job according to the Random policy.
  - The objective of SB-Deferred is twofold: **1)** to avoid the delay of small-sized jobs in GS's queue and **2)** to dispatch the large jobs to the most appropriate sites since they constitute a large fraction of the total load.

Local policy

- The LS applies a policy which determines the method a processor is selected in order to serve an incoming job.
- We have chosen the ***Least Work Remaining*** (LWR) policy.
- LSs are aware of service demands of jobs, monitor the remaining work in each local queue, and select the processor with the least remaining work.
- We have chosen LWR in order to minimize the delay of jobs in local queues.
- The FCFS policy is applied in local queues.

Performance metrics

- **Response time** of a job is the time period from the arrival to the GS to the time service completion of the job.

- **Slowdown** of a job is the job's response time divided by its service time.
  - The importance of the slowdown metric is increased in a system at which job service demands are highly variable, due to the fact that relatively long delays for demanding jobs can be acceptable.

Performance metrics

P	number of processors in system
λ	mean arrival rate
1/λ	mean inter-arrival time of jobs
μ	mean service rate
1/μ	mean service demand of jobs
A_I	allocation interval
SDT	service demand threshold
α	shape of Pareto
L	lowest bound of Bounded Pareto
H	highest bound of Bounded Pareto
U	average system utilization
RT	average response time of jobs
MaxRT	maximum RT
SLD	average slowdown

TABLE I. NOTATIONS OF THE PARAMETERS

Experimental setup

- We developed a simulation application in C programming language.
- The application operates according to the discrete event simulation technique.
- Each simulation experiment ends when 80000 jobs' executions are completed.
- We used a warm-up period of 5000 job executions.
- Each result presented is the average value that is derived from 100 simulation experiments with different seeds of random numbers.

Experimental setup

- Inter-arrival times
  - Two cases for the mean job inter-arrival time are considered in this paper:

    $1/\lambda = 0.028, 0.014$

  - The mean arrival rates of jobs are respectively:

    $\lambda = 35.71, 71.43$

  - An approximation of the corresponding average system utilization values is the following:

    $U = 45\%, 90\%$

Experimental setup

- Service demand times
  - We chose the mean service demand of jobs to be equal to 1 ($1/\mu = 1$).
  - We vary α in order to examine the impact of different levels of variability on system's performance.
  - Table below presents the L and H parameters for various α values that we examine.

α	**2**	**1.75**	**1.5**	**1.25**
H	100	100	100	100
L	0.502	0.436	0.354	0.258

Regarding A_I, we chose to be equal to the mean service demand of jobs (A_I=1) in the sets of experiments that we conducted.

Experimental results

Impact of Service Demand Variability (α)

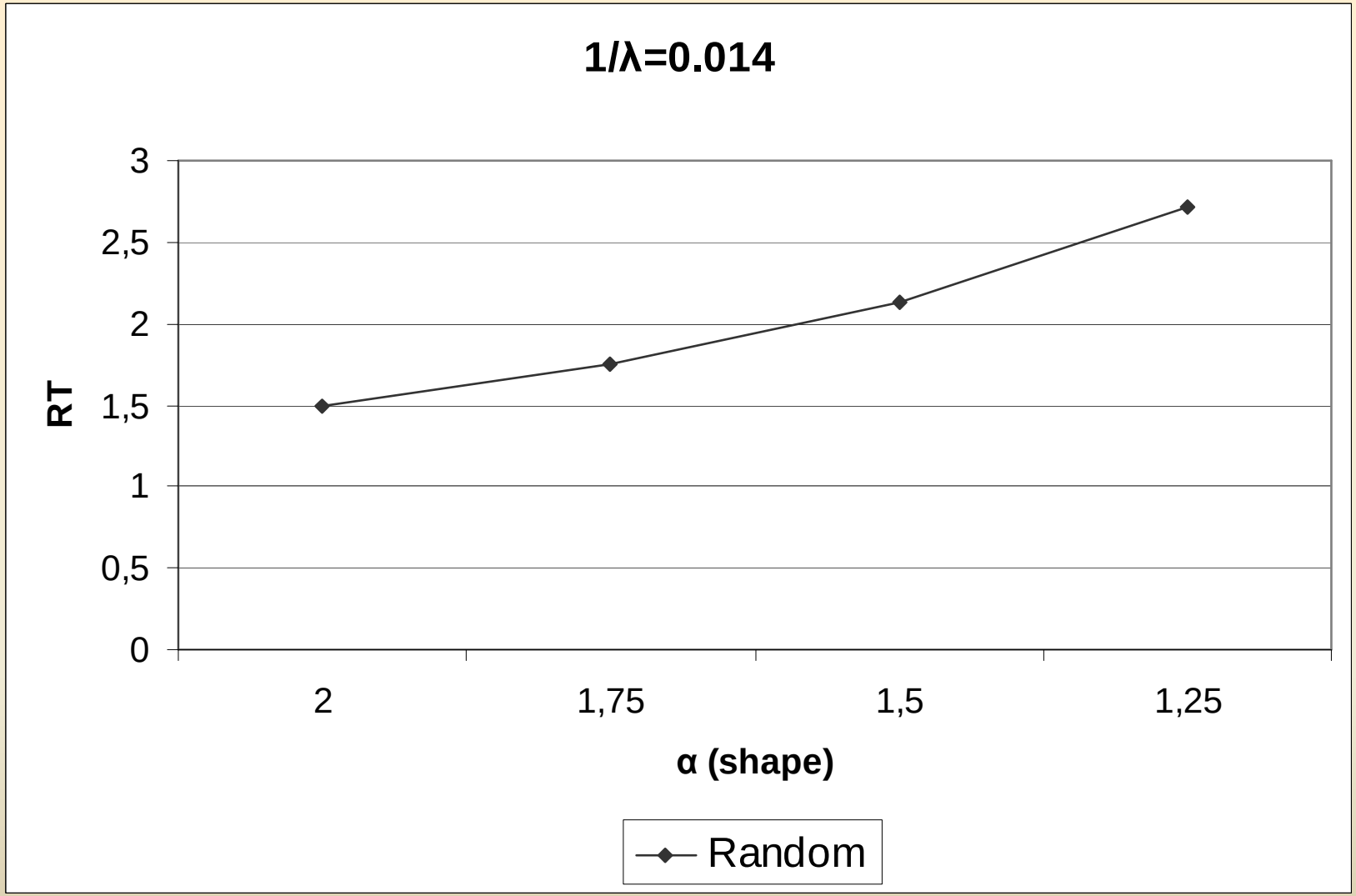

Figure 3. RT versus α when 1/λ=0.014 for Random policy

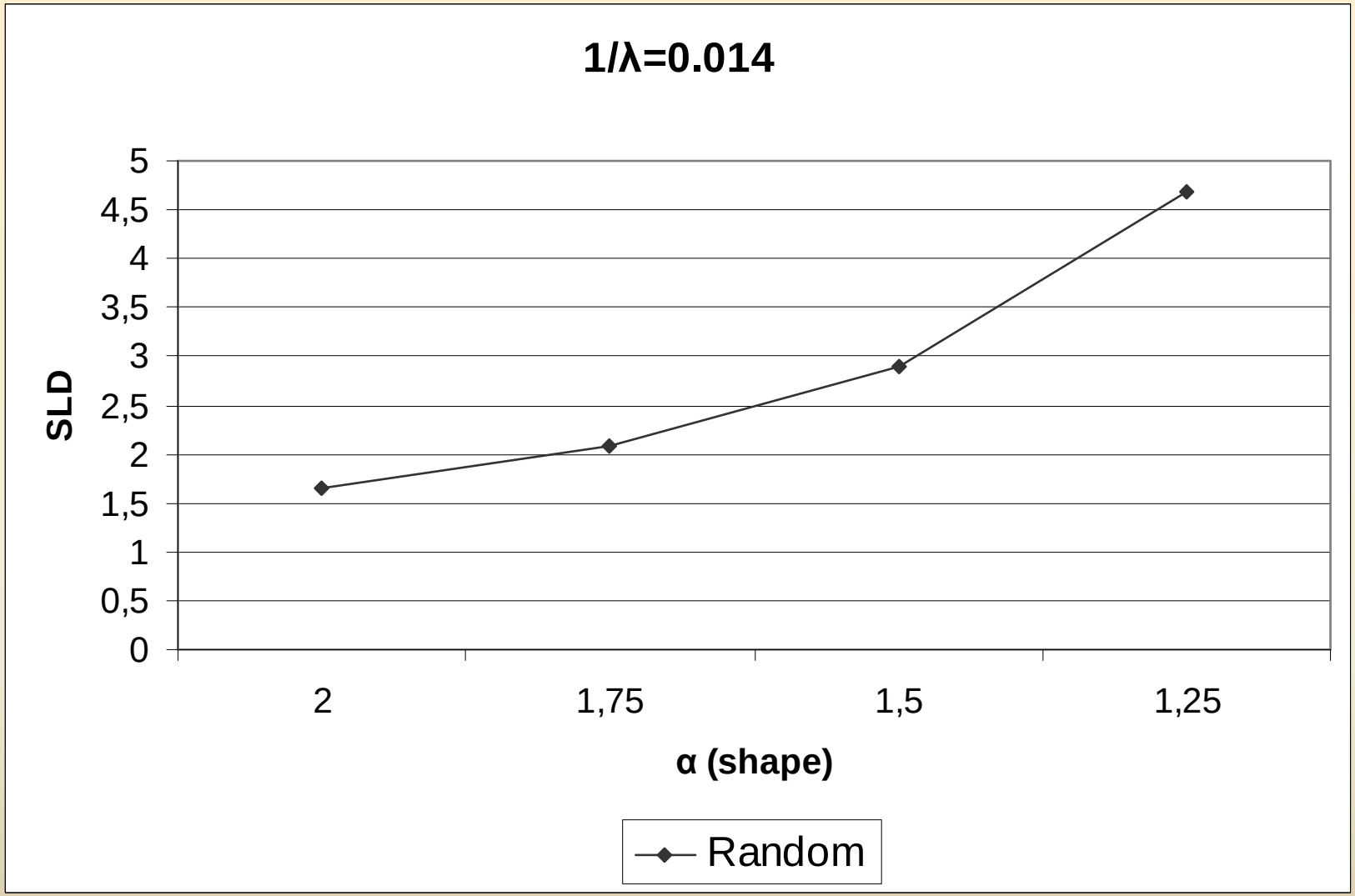

Figure 4. SLD versus α when 1/λ=0.014 for Random policy

Experimental results

Impact of SDT

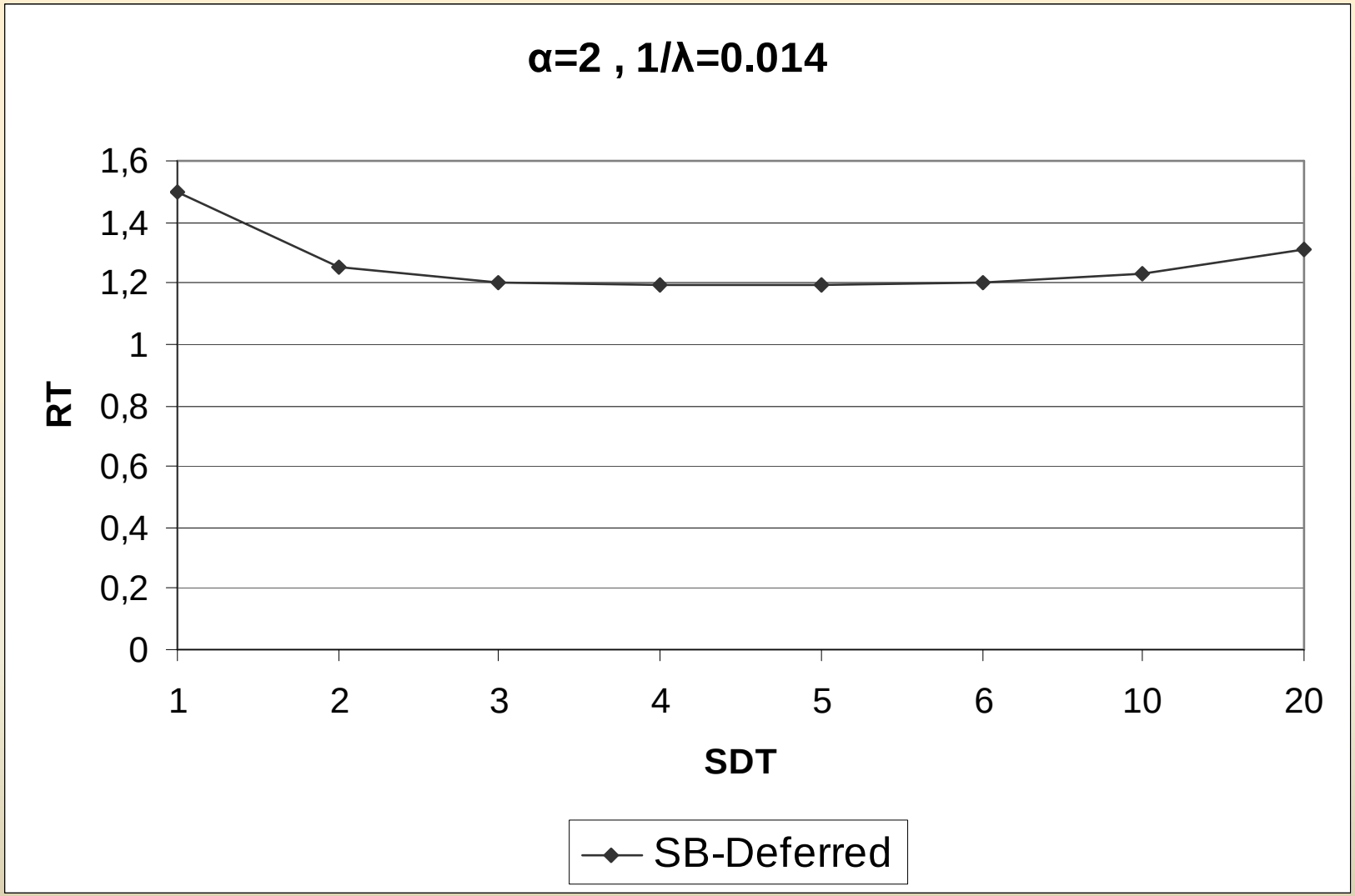

Figure 5. RT versus SDT when α=2 for SB-Deferred policy

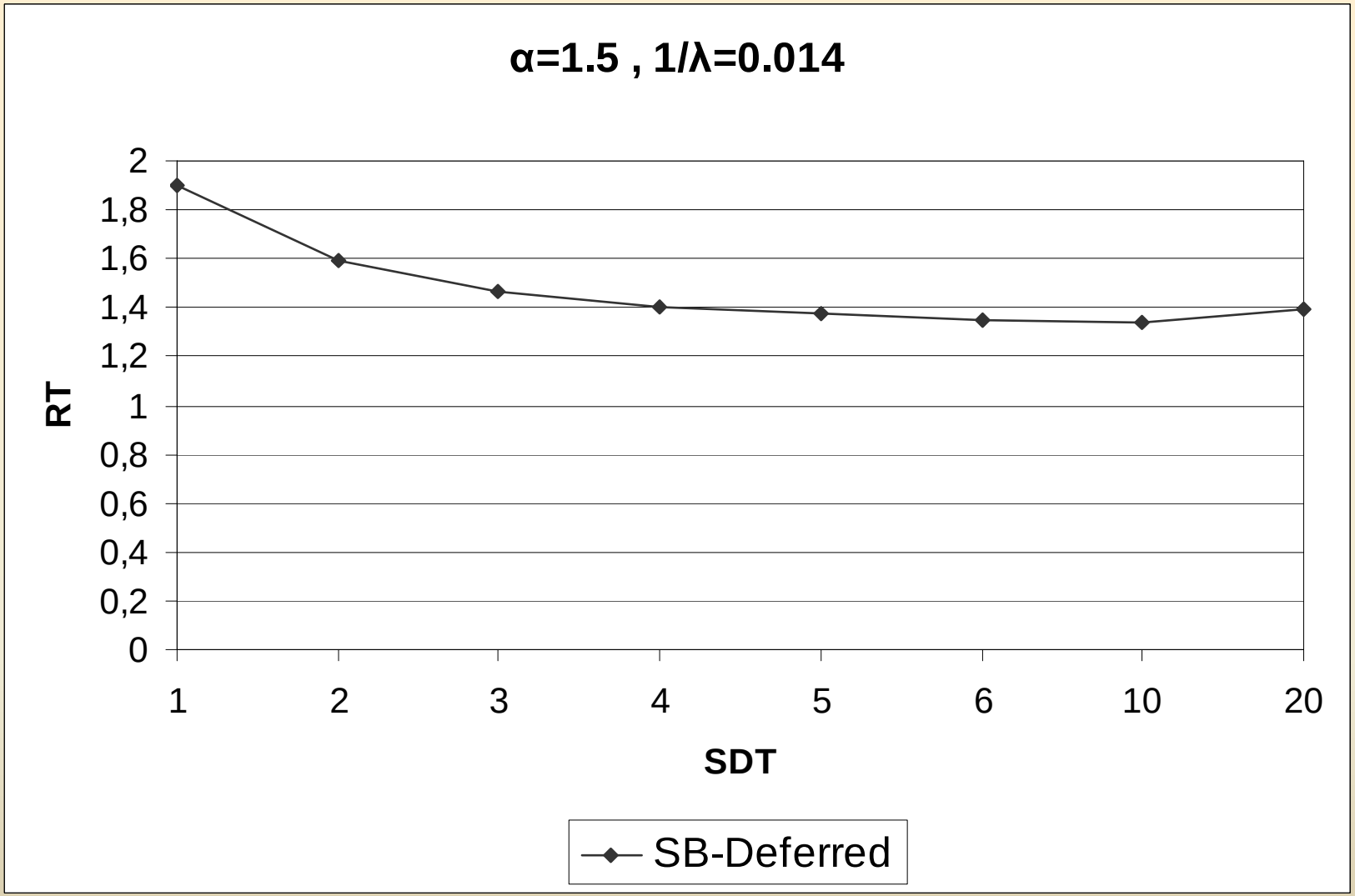

Figure 6. RT versus SDT when α=1.5 for SB-Deferred policy

Experimental results

Performance Evaluation of the Policies

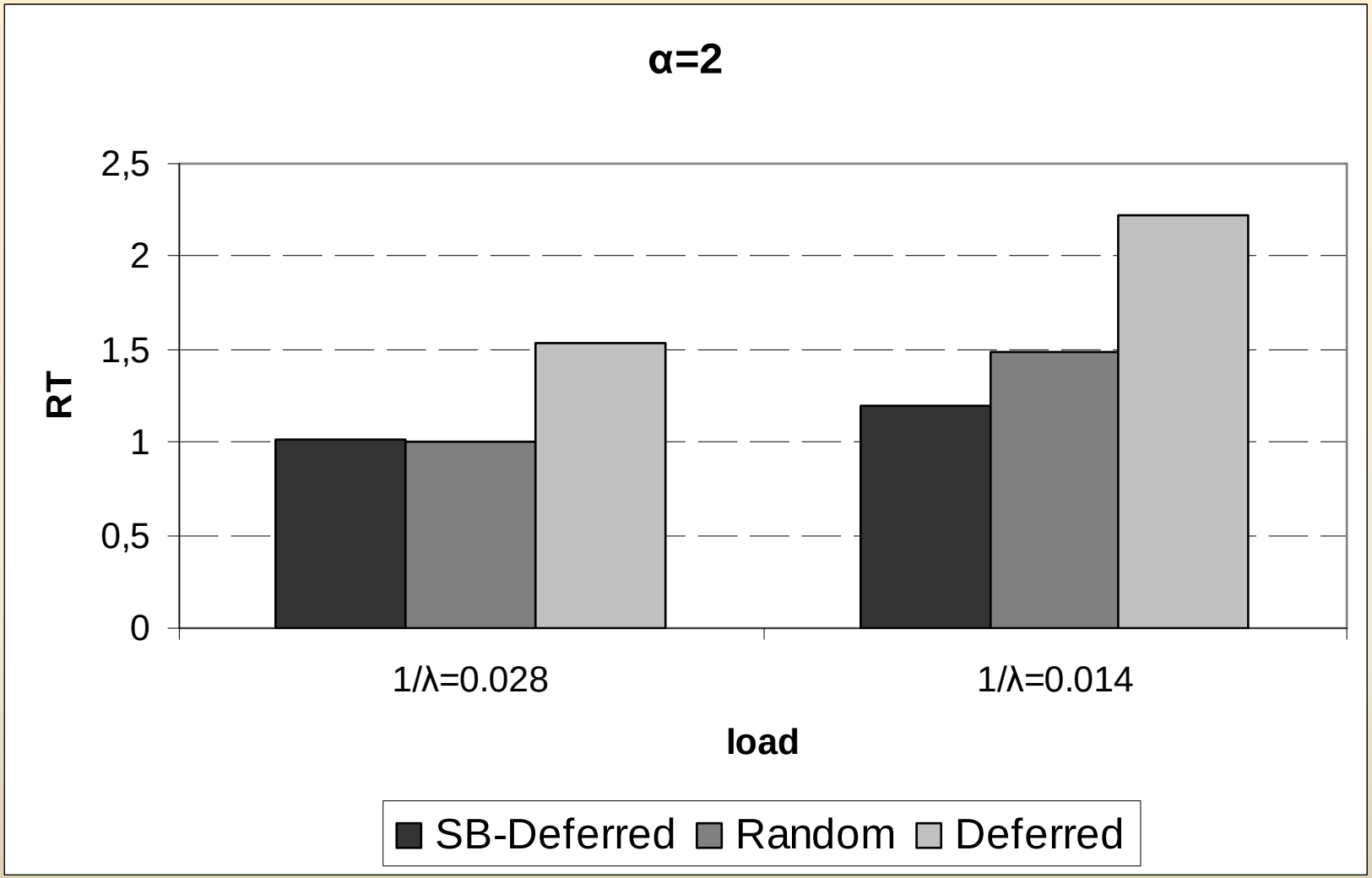

Figure 7. Comparison of the policies in terms of RT when α=2

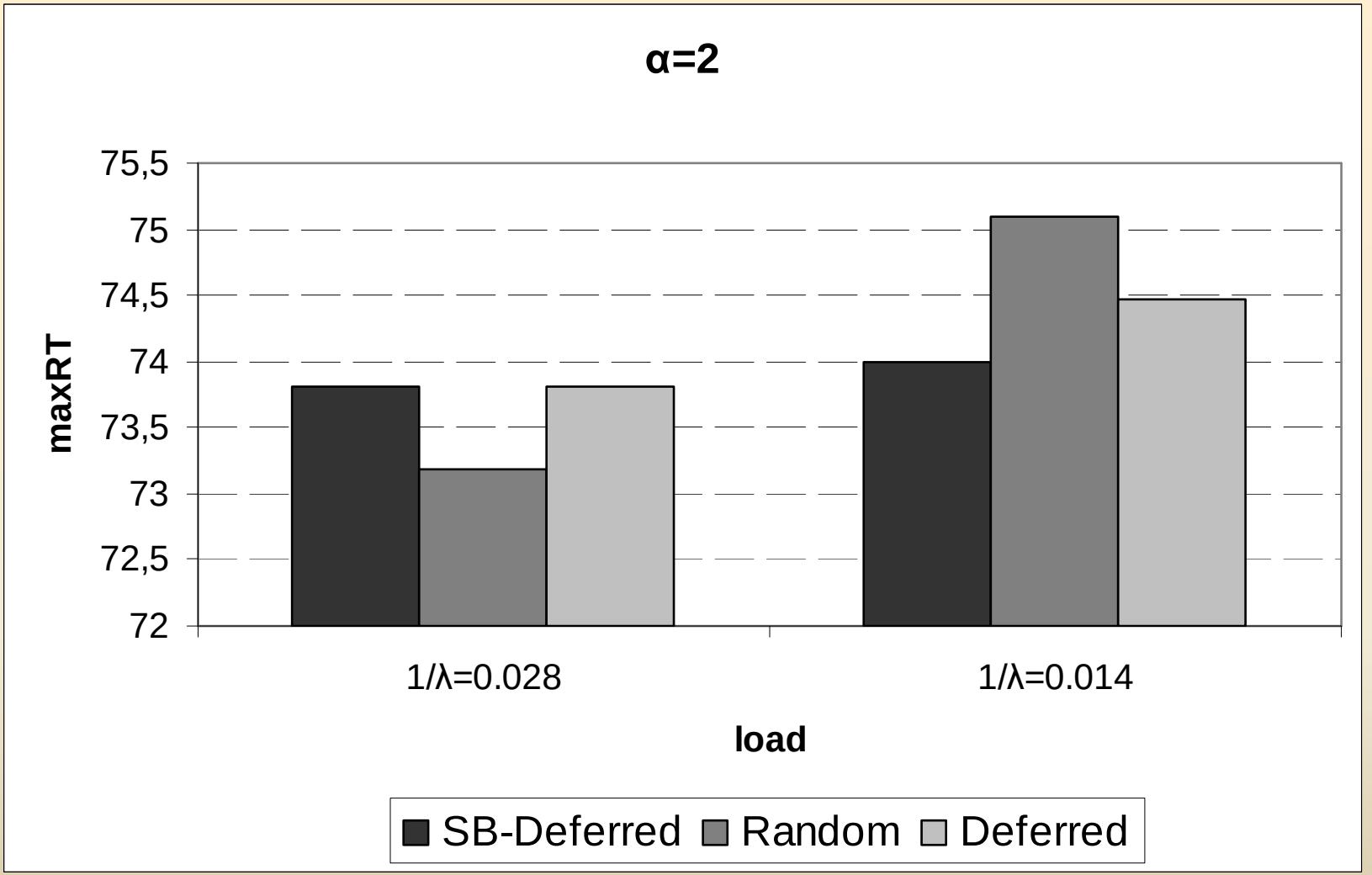

Figure 8. Comparison of the policies in terms of maxRT when α=2

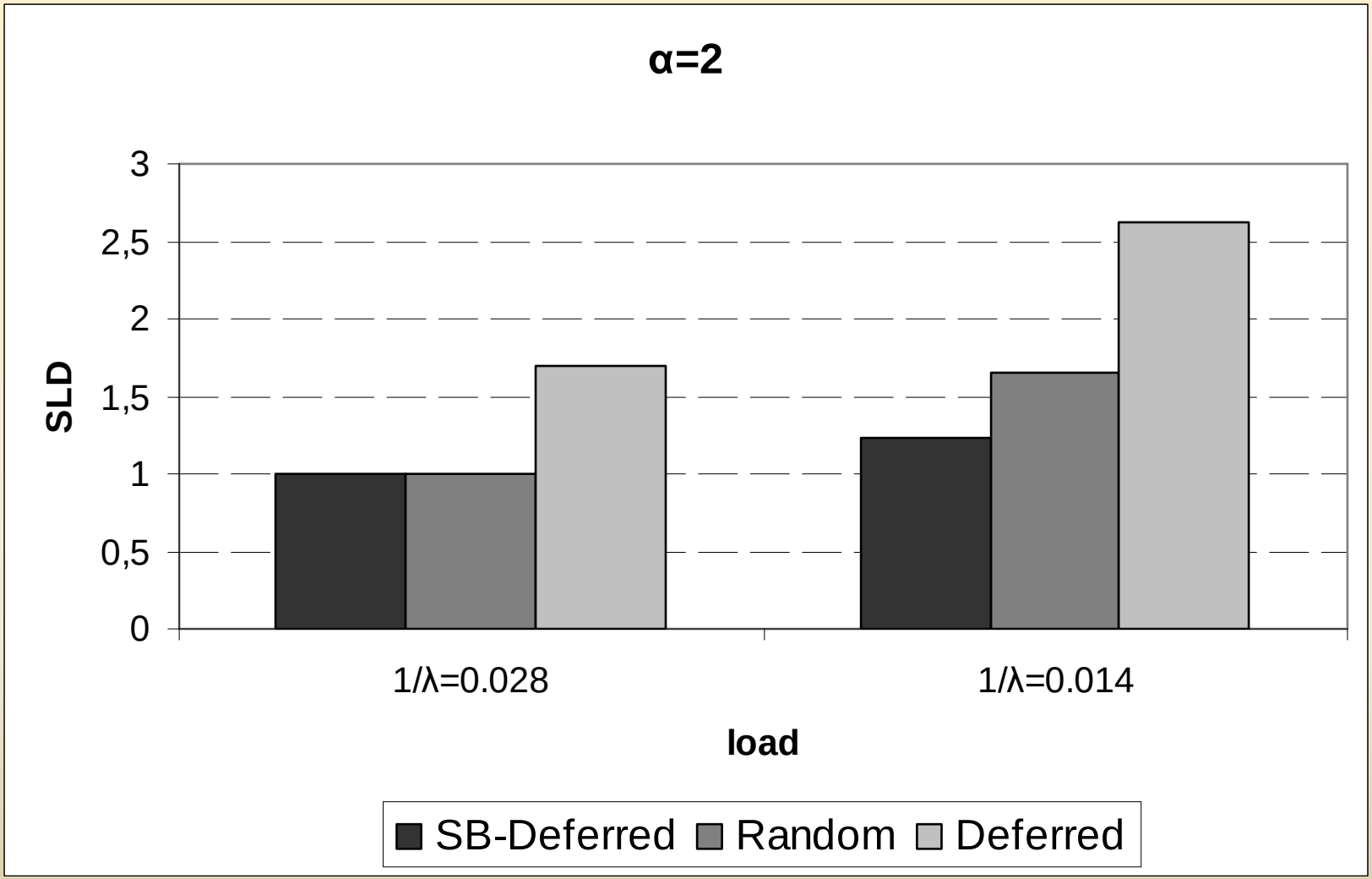

Figure 9. Comparison of the policies in terms of SLD when α=2

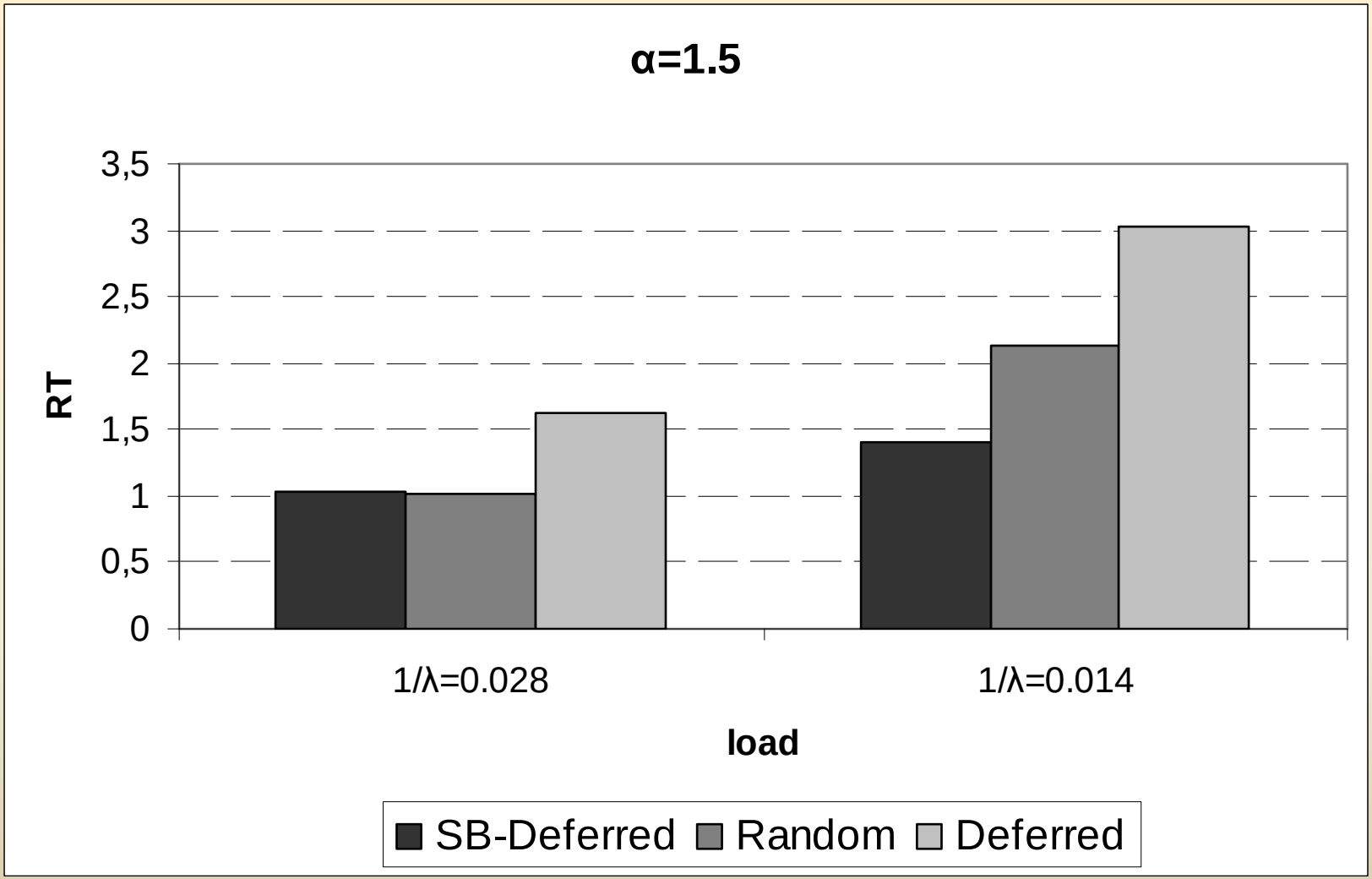

Figure 10. Comparison of the policies in terms of RT when α=1.5

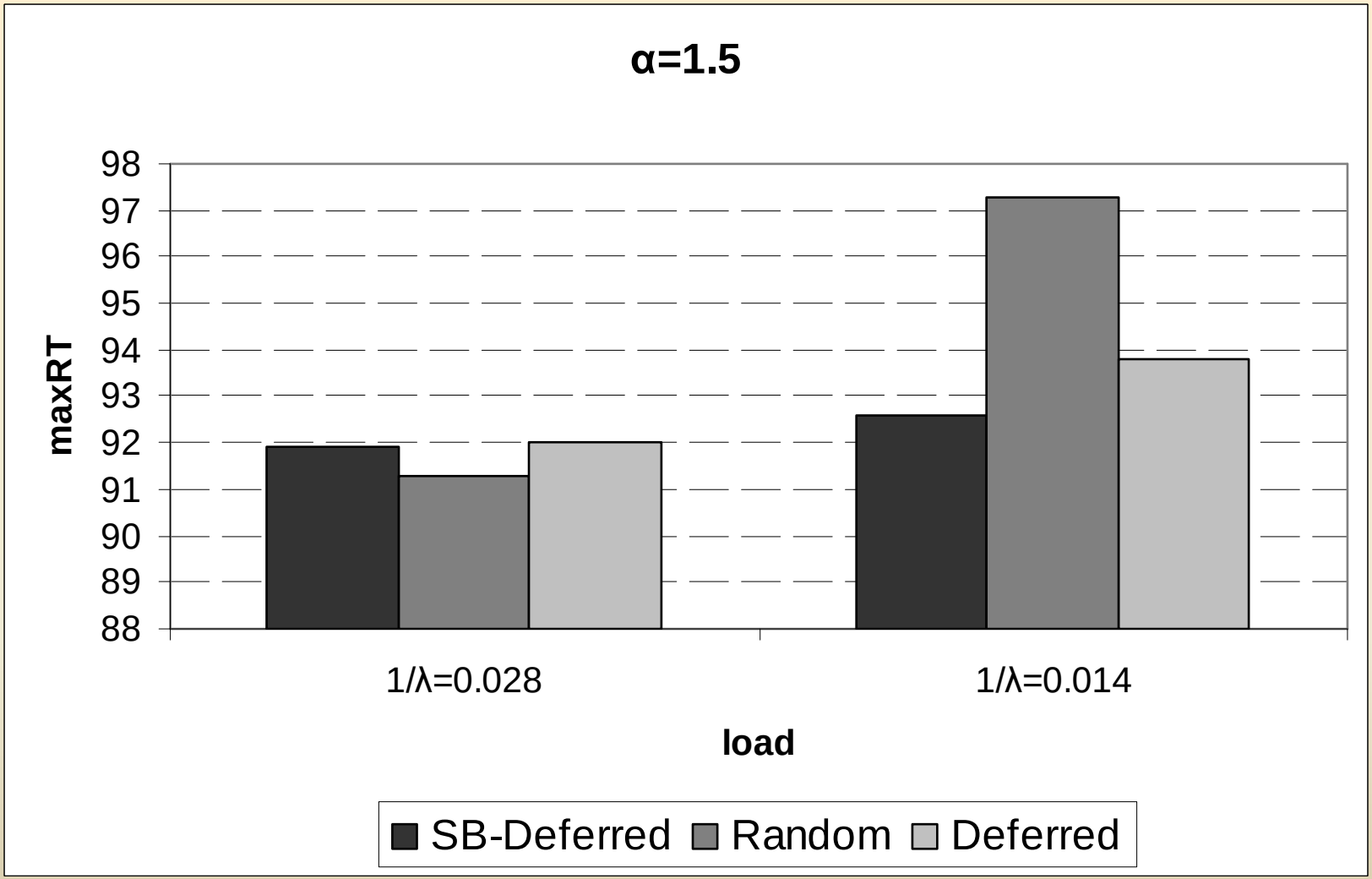

Figure 11. Comparison of the policies in terms of maxRT when α=1.5

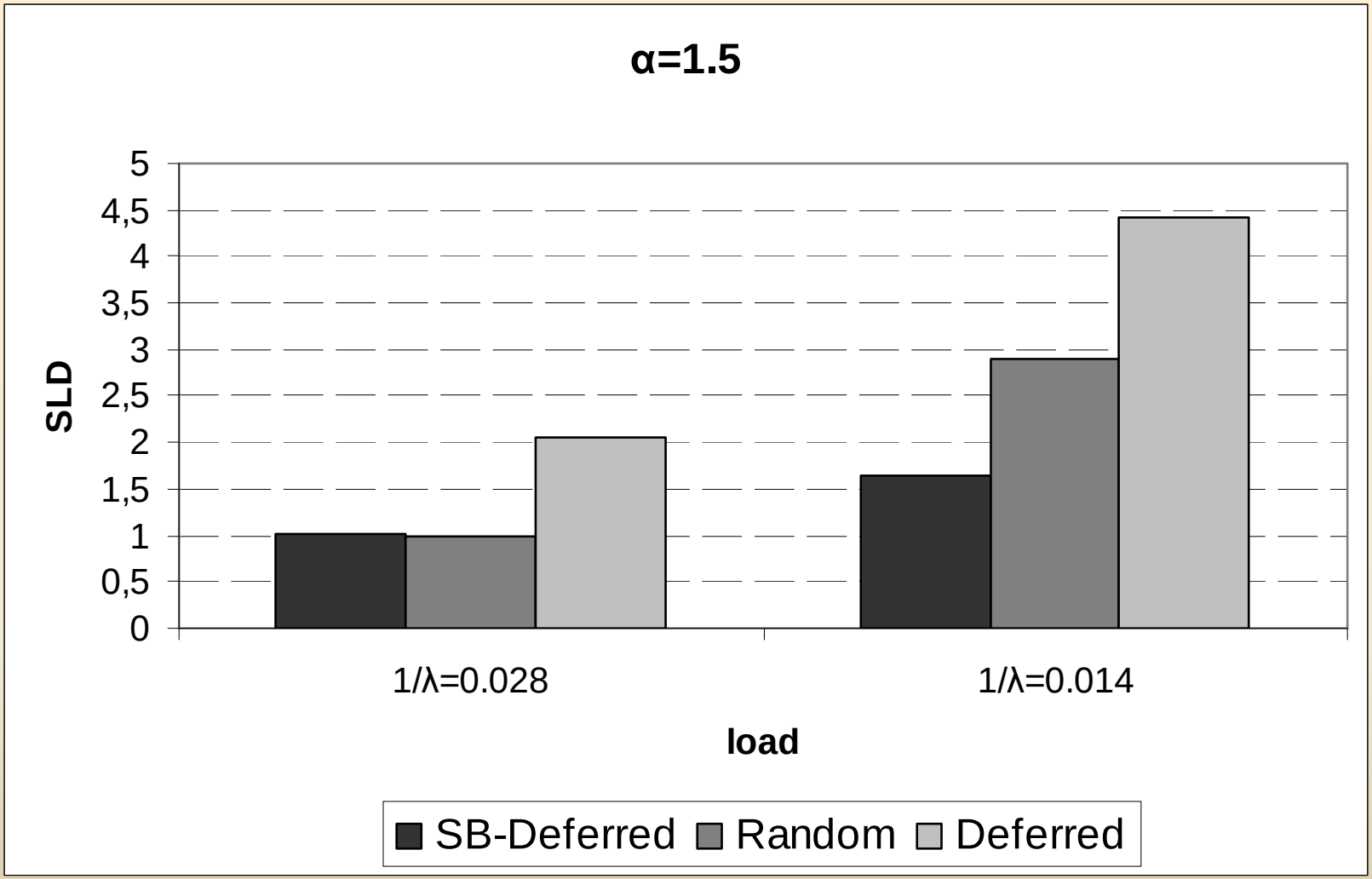

Figure 12. Comparison of the policies in terms of SLD when α=1.5

Conclusions

- In the present paper we evaluated the performance of three site allocation policies (Random, Deferred, and SB-Deferred) in a 2-level computational grid.
- The proposed SB-Deferred policy, which combines Random and Deferred, outperformed both Random and Deferred when they are applied separately, even at high service demand variability.
- We also showed that the performance degradation due to load increase is minor when SB-Deferred is employed instead of the two other policies.

Future directions

- As future work, we plan to model the estimation of service demands of jobs by the schedulers, in order to examine the behaviour of the policies.

- Furthermore, it would be interesting to conduct simulation experiments in the case where additional metrics for site load information are used, such as the number of idle processors.

THANK YOU!