Resource Allocation Policies for Smart Energy Efficiency in Data Centers

Marina Khludova
St-Petersburg State Polytechnic University, Russia
Institute of Applied Mathematics and Mechanics
Department of Telematics
28.08.2014
ruSMART_2014

Motivation

Data Centers in USA consumed about 61 Billions kW-hour and accounted for 1.5% of total US electricity consumption in 2006 [1]

It has been reported that energy consumption of Google datacenters is equivalent to the total energy consumption of a small city

---

[1] **US Environmental Protection Agency, Report to congress on server and data center energy efficiency, Technical Report, ENERGY STAR Program, August 2007.**

Motivation_2

Energy consumption measurement results of many computer components [2]

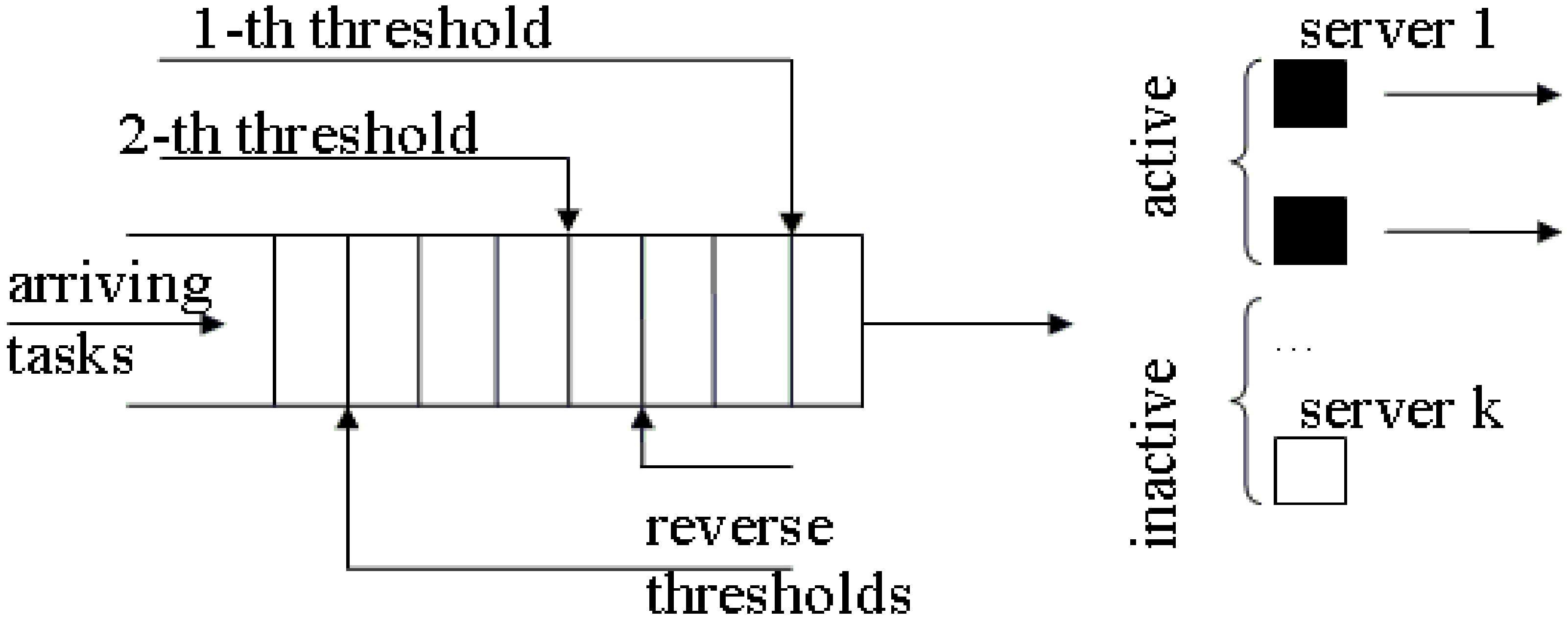
Component	Idle (W)	Max load (W)
CPU	8.49	38.66
RAM	6.06	6.23
HD	5.93	7.59

---

[2] T. Heath, B. Diniz, E.V. Carrera, W.M. Jr., R. Bianchini, Energy conservation in heterogeneous server clusters, in: Proc. of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming-PPoPP’05, 2005, pp.186–195.

System Model_M|M|k_t

Number of active servers is controlled and switching costs are incurred



Problem Formulation

The system state may be described by $\{n(t); k(t)\}$, where

$n(t)$ - the number of tasks in the system at time t ,

$k(t)$ - number of active servers at time t .

We want to construct a function

$$\Psi\ (n(t-); k(t-))=(n(t); k(t))\ ,$$

such that the average revenue earned per unit time is maximized

Average revenue earned per unit time

$$h_c\, n_{k^*} + e\,(K - k^*) + g\, r_{k^*} P_{success} - z_1\, r_{k^*}(1 - P_{success}) - z_2 ,$$

where

n_{k^*} -the mean number of tasks in the system,

k^* - the optimal number of servers that maximizes average revenue,

K – the total number of servers,

h_c -a negative ‘holding cost’, to keep a task in the system for one unit of time,

e - a profit to keep a server switched off for a unit of time,

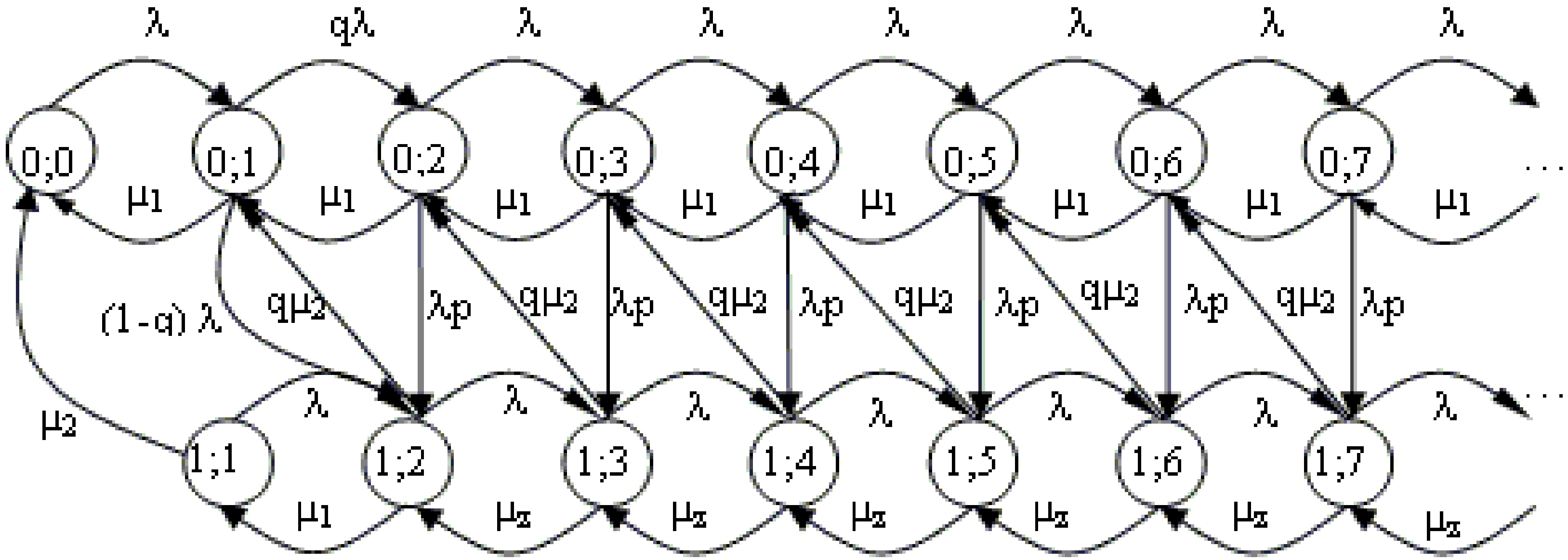
g - a profit to complete a task in accordance with its QoS need,

z_1, z_2 – penalties (QoS needs violation, server switching),

r_k – the mean throughput,

$P_{success}$ - the portion of tasks successfully completed in accordance with their QoS needs .

State transition diagram for policy with hysteresis behavior

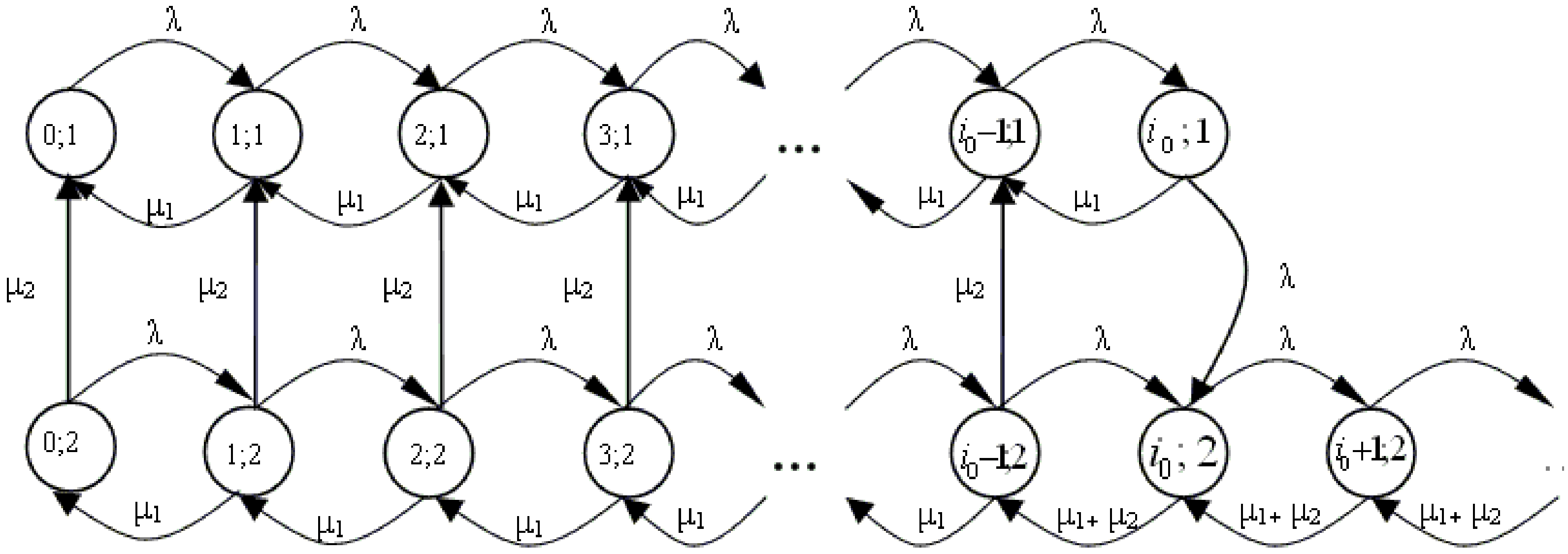


$$\mu_z = \mu_1 + (1-q)\mu_2$$

Continuous-time version of the Chapman-Kolmogorov equations

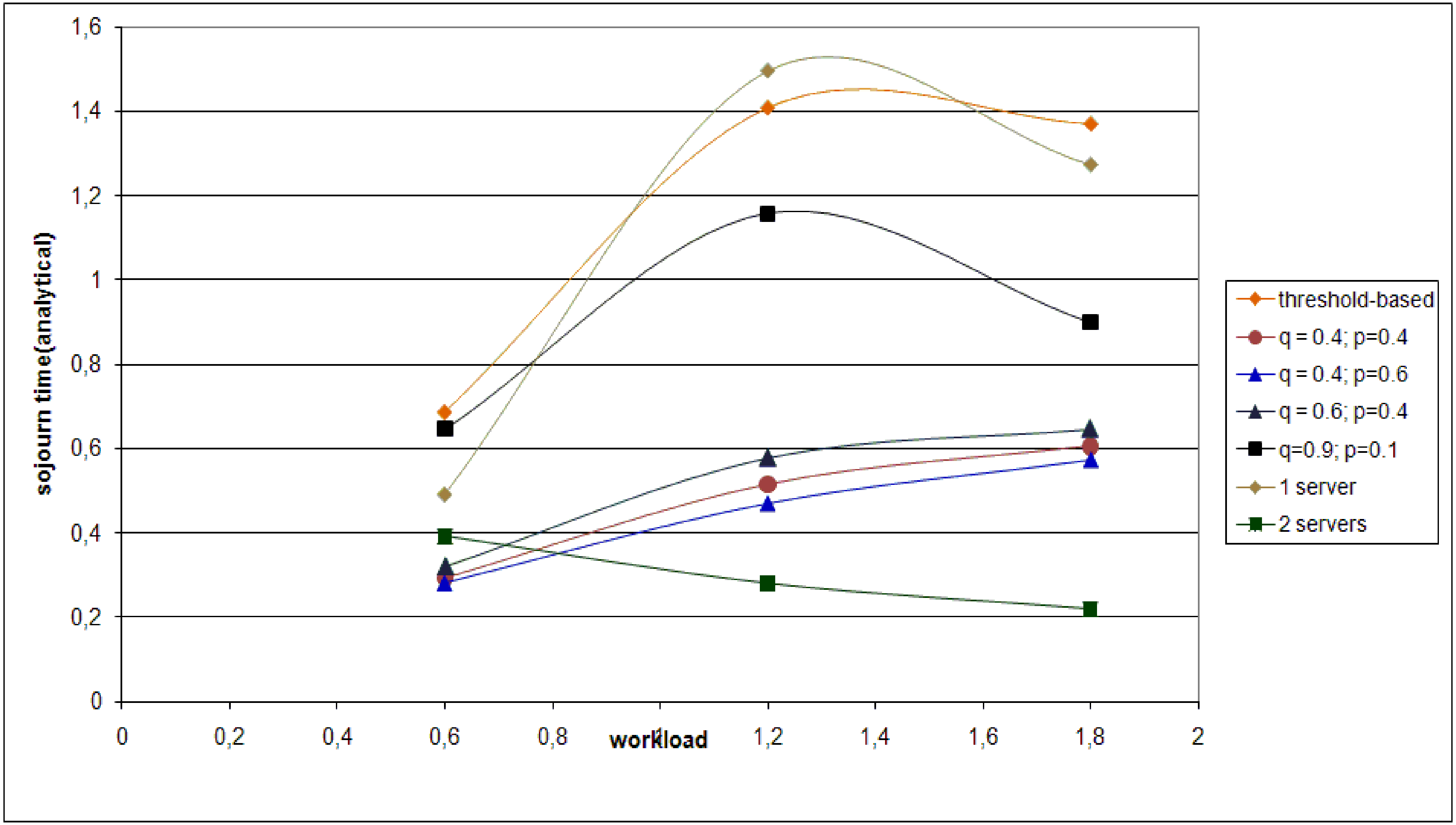
$$
\begin{cases}
-\lambda P_{00}(t)+\mu_1 P_{01}(t)+\mu_2 P_{11}(t)=\dfrac{dP_{00}(t)}{dt}\\
-(\lambda+\mu_1)P_{01}(t)+\lambda P_{00}(t)+\mu_1 P_{02}(t)+q\mu_2 P_{12}(t)=\dfrac{dP_{01}(t)}{dt}\\
-(\lambda+\mu_1+\lambda p)P_{02}(t)+q\lambda P_{01}(t)+\mu_1 P_{03}(t)+q\mu_2 P_{13}(t)=\dfrac{dP_{02}(t)}{dt}\\
-(\lambda+\mu_1+\lambda p)P_{0i}(t)+\lambda P_{0i-1}(t)+\mu_1 P_{0i+1}(t)+q\mu_2 P_{1i+1}(t)=\dfrac{dP_{0i}(t)}{dt},\forall i\in[3;9]\\
-(\mu_1+\lambda p)P_{010}(t)+\lambda P_{09}(t)=\dfrac{dP_{010}(t)}{dt}\\
-(\lambda+\mu_2)P_{11}(t)+\mu_1 P_{12}(t)=\dfrac{dP_{11}(t)}{dt}\\
-(\lambda+\mu_1+q\mu_2)P_{12}(t)+(1-q)\lambda P_{01}(t)+\lambda p P_{02}(t)+\lambda P_{11}(t)+(\mu_1+(1-q)\mu_2)P_{13}(t)=\dfrac{dP_{12}(t)}{dt}\\
-(\lambda+(\mu_1+(1-q)\mu_2)+q\mu_2)P_{1i}(t)+\lambda p P_{0i}(t)+\lambda P_{1i-1}(t)+(\mu_1+(1-q)\mu_2)P_{1i+1}(t)=\dfrac{dP_{1i}(t)}{dt},\forall i\in[3;9]\\
-((\mu_1+(1-q)\mu_2)+q\mu_2)P_{110}(t)+\lambda p P_{010}(t)+\lambda P_{19}(t)=\dfrac{dP_{110}(t)}{dt}
\end{cases}
$$

State transition diagram for threshold-based policy

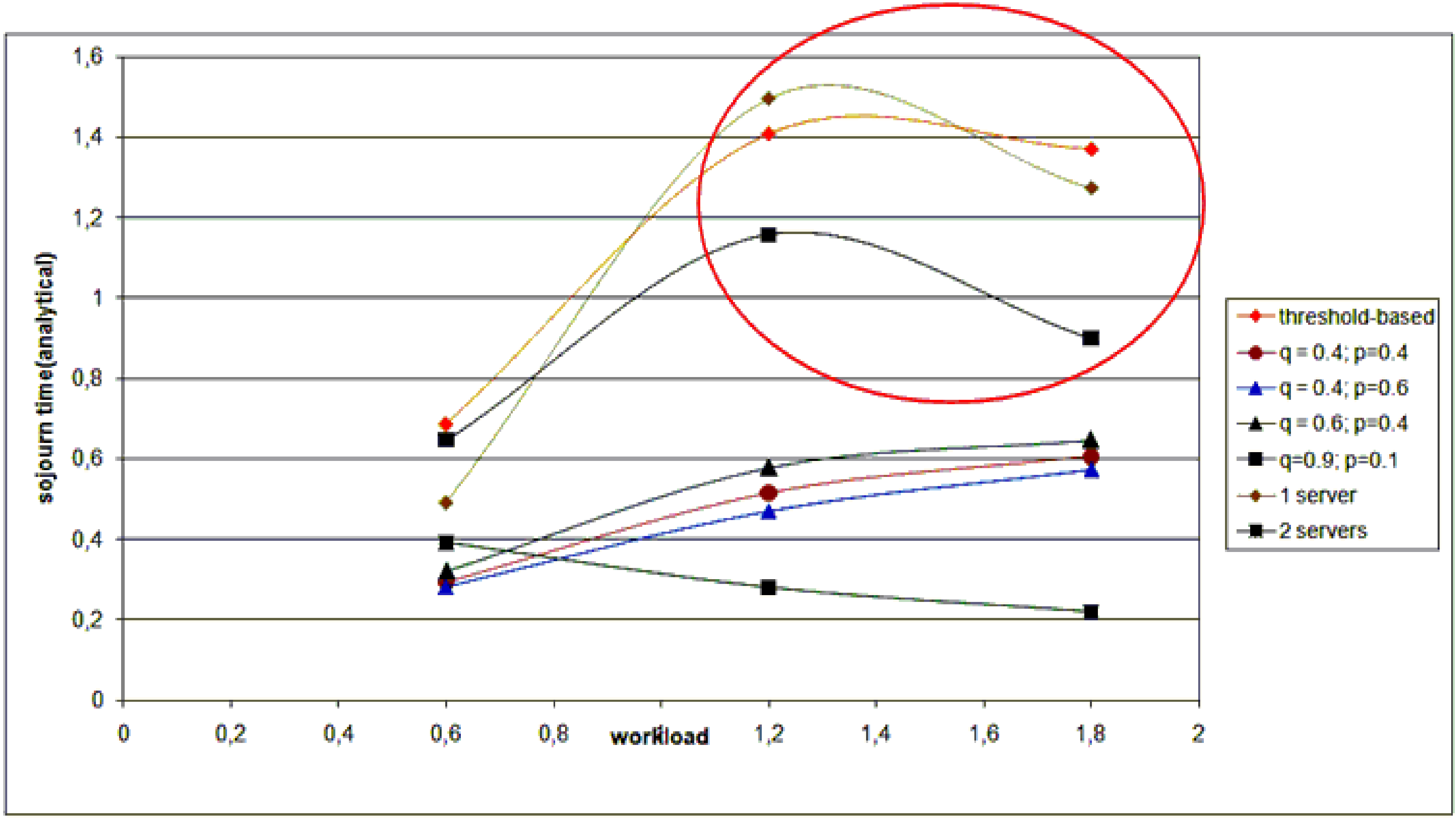


Performance Evaluation Result 1

Average sojourn time (in seconds) **derived from solving global balance equations**

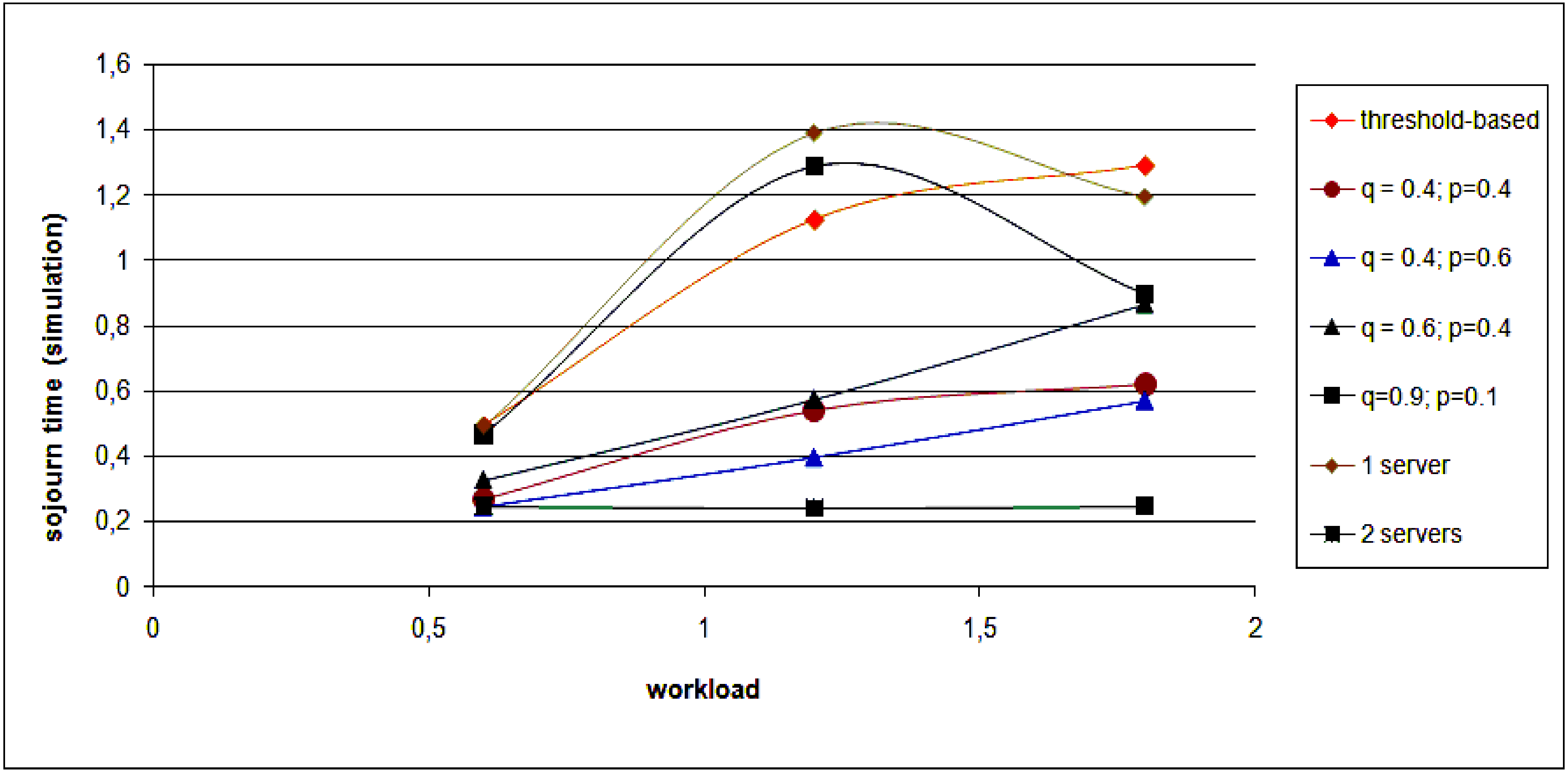


Performance Evaluation Result 1



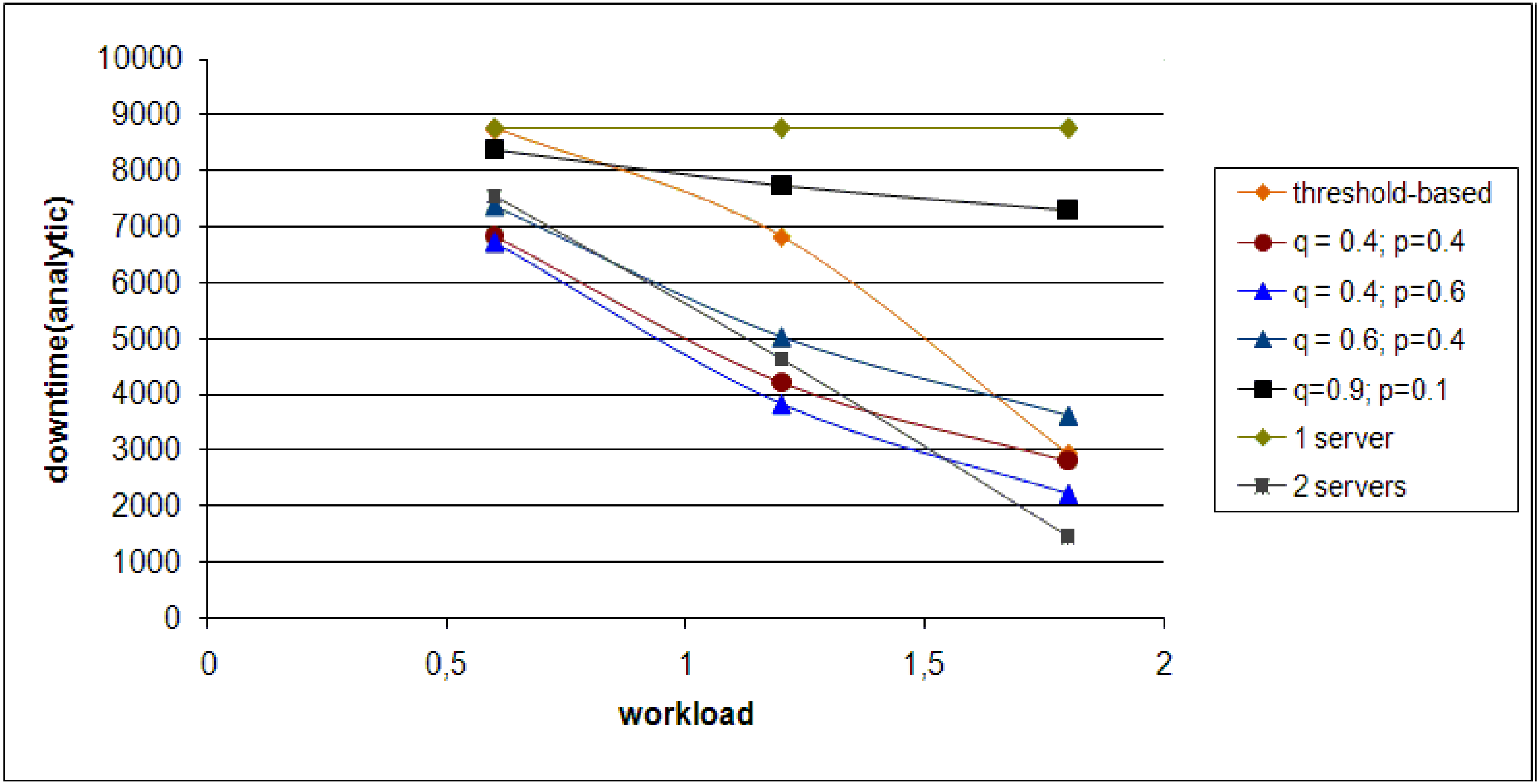
Performance Evaluation Result 2

Average sojourn time (in seconds) derived from simulation model (GPSS)



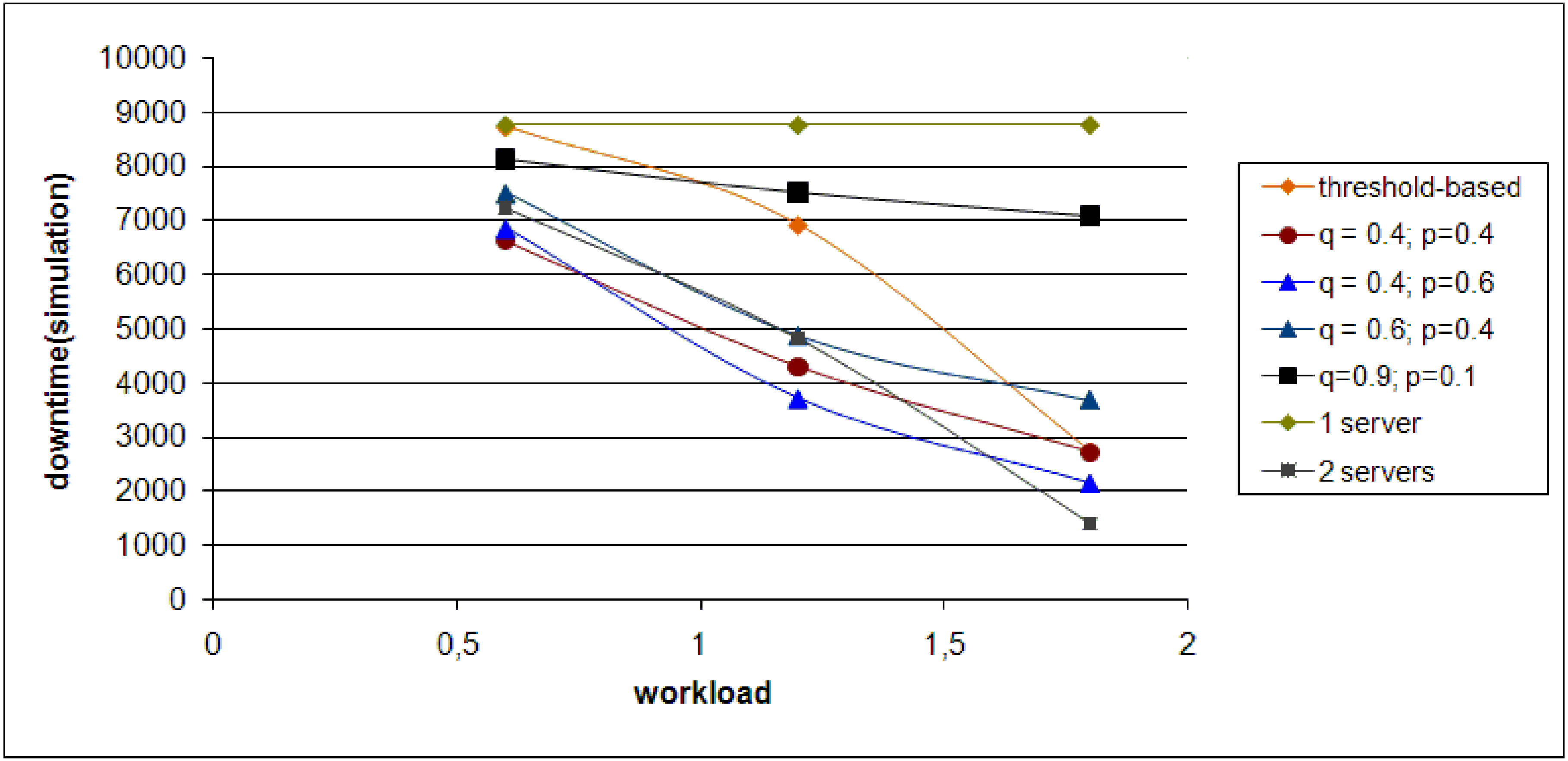
Performance Evaluation Result 3

Downtime (in hours) for reserve server per year for proposed and commonly used policies (analytical model)

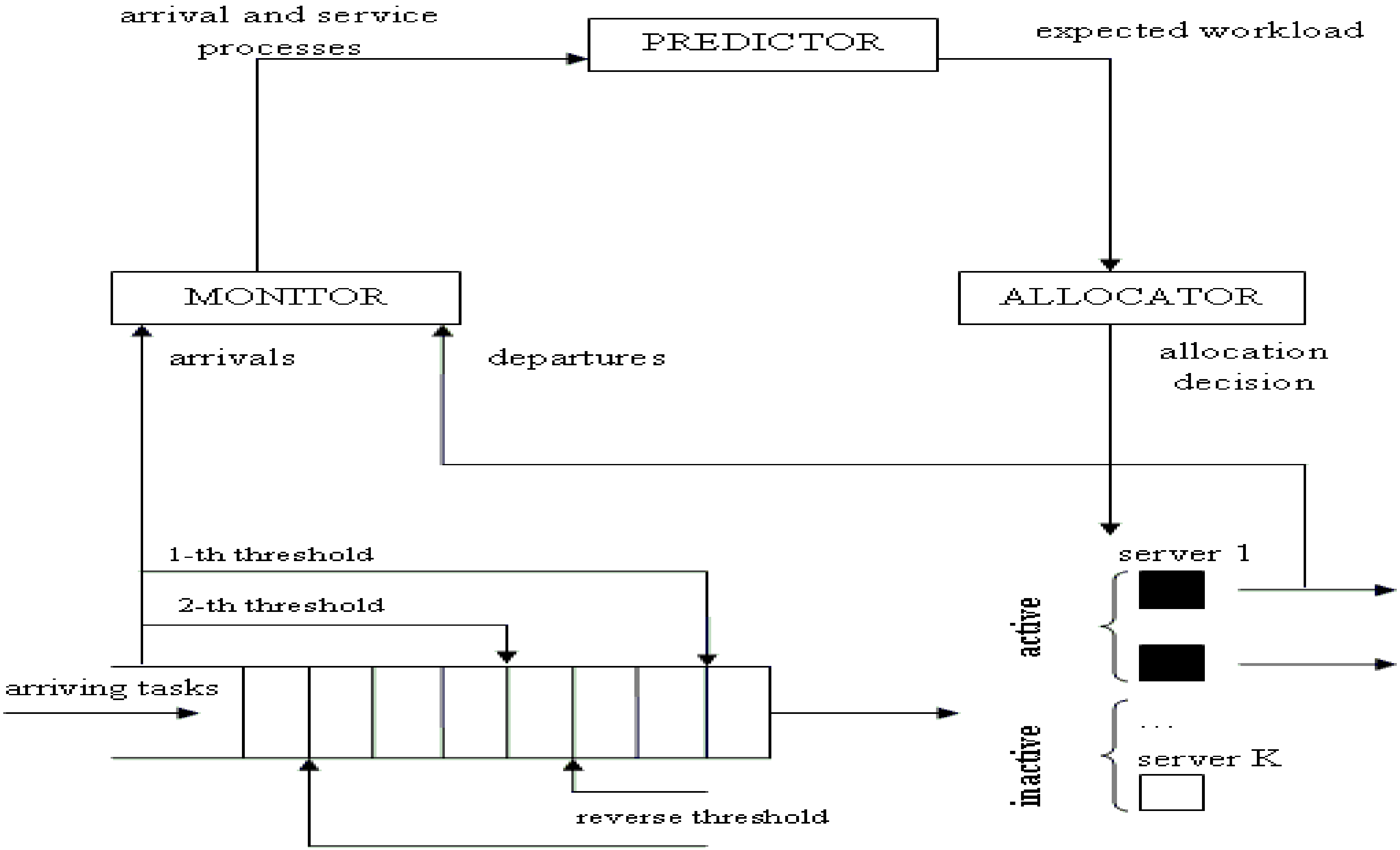


Performance Evaluation Result 4

Downtime (in hours) for reserve server per year for proposed and commonly used policies (simulation)



System architecture

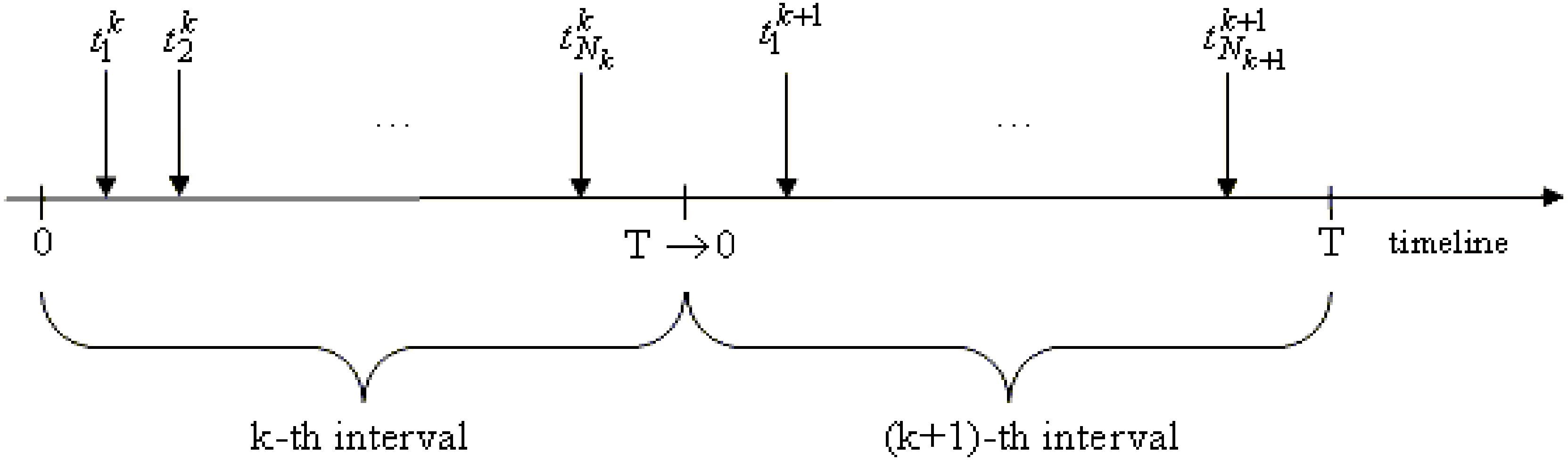


On-line Monitoring and Measurement 1

The tasks arrive at the system with the rate *λ(t).*
How can we estimate this rate ?

Let $\lambda(t) = c + d\frac{t}{T}, 0 \le t \le T$

$$v = \frac{N}{T}, \quad y = \frac{1}{T}\sum_{i=1}^{N}\frac{t_i}{T}$$



On-line Monitoring and Measurement 2

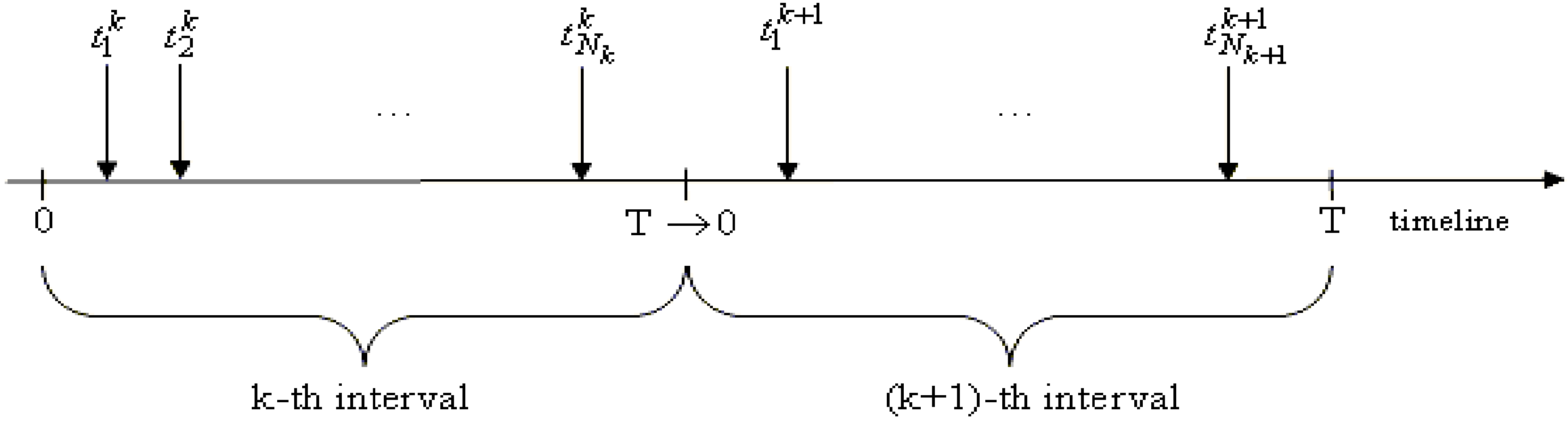
For k-th interval: $\hat{\lambda}_k(t) = \hat{c}_k + \hat{d}_k \left(\frac{t}{T} \right); 0 \le t \le T; k = 0,1,\dots$,

$$\hat{c} = 4\frac{N}{T} - 6\sum_{i=1}^{N} \frac{t_i}{T},$$

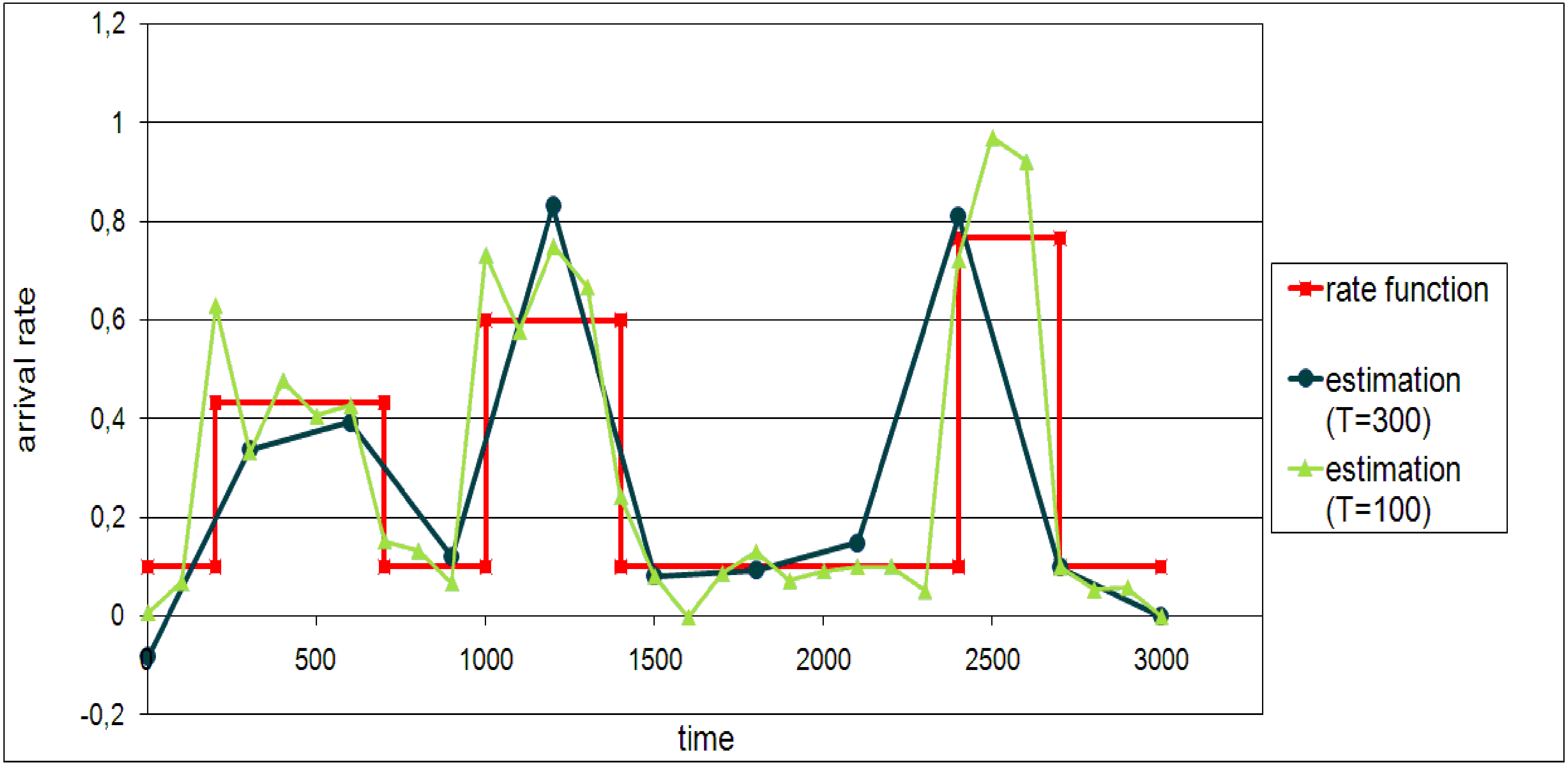
$$\hat{d} = -6\frac{N}{T} + \frac{12}{T}\sum_{i=1}^{N} \frac{t_i}{T}.$$

$$\hat{\lambda}_{k+1}(0) = \hat{\lambda}_k(T); k = 0,1,\dots$$

$$\hat{c}_{k+1} = \hat{c}_k + \hat{d}_k; k = 0,1,\dots$$



Rate function for non-stationary Poisson process and its estimation



Allocating Resources

The mean waiting time for *delayed* tasks for this **M/M/k**$_t$ queue:

$$E[\hat{T}_W] = \frac{E[\hat{b}]}{k(t) - \hat{\lambda}_k E[\hat{b}]} .$$

It can be used to define QoS needs for tasks and find the number *k** that maximizes average revenue earned per unit time.

System is stable, if $\hat{\lambda}_k E[\hat{b}] \leq k^*$

Conclusion and Further Work

- We considered techniques for dynamic resource allocation in DC. The main goal of our techniques is to react to transient system overloading and underloading in such way that the energy usage, reliability and QoS performance should be balanced.

- In order to comply with energy efficiency we use *threshold-based resource allocation policies* with *hysteresis* and present an efficient technique for solving the corresponding analytical models and computing various performance measures of interest.

- This model dynamically relates the total resource requirements of all applications to their workload characteristics. The advantage of this model is that it does not make steady-state assumptions about the system and adapts to changing application behavior.

The main directions for further investigation are:

As components degrade faster when they are switched on/off, therefore, it incurs the following *amortized* cost.

Wavelet analysis can be used to estimate the system model parameters and workload characteristics.

Thank you.

Questions?