Future Internet **2013**, *5*, 128-139; doi:10.3390/fi5020128

OPEN ACCESS

future internet

ISSN 1999-5903

www.mdpi.com/journal/futureinternet

Article

Energy–QoS Trade-Offs in Mobile Service Selection

Erol Gelenbe * and Ricardo Lent *

Intelligent Systems & Networks Group, Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2BT, UK

* Authors to whom correspondence should be addressed; E-Mail: e.gelenbe@imperial.ac.uk (E.G.); r.lent@imperial.ac.uk (R.L.); Tel.: +44-207-594-6173 (R.L.); Fax: +44-207-594-6274 (R.L.).

Received: 28 February 2013; in revised form: 5 March 2013 / Accepted: 12 April 2013 / Published: 19 April 2013

Abstract: An attractive advantage of mobile networks is that their users can gain easy access to different services. In some cases, equivalent services could be fulfilled by different providers, which brings the question of how to rationally select the best provider among all possibilities. In this paper, we investigate an answer to this question from both quality-of-service (QoS) and energy perspectives by formulating an optimisation problem. We illustrate the theoretical results with examples from experimental measurements of the resulting energy and performance.

Keywords: system load; measurements; QoS-energy trade-offs; optimum distribution

---

1. Energy and QoS

The power consumption of mobile services will depend on the load. Clearly quality of service (QoS) will also depend on load because a more heavily loaded computational or communication resource will quite naturally increase response times. However such issues are somewhat more complex, because the server clusters hosting the services may turn off some of the resources under lighter loads, so that when load is higher although power consumption will obviously increase, QoS can also improve.

A simple but quite realistic power consumption relation for current processing units is $\Pi = A + B\rho$, where A is the power consumption of the processing unit when it is idle, and B is the rate at which it increases as a function of the load factor ρ [1]. Thus, a very efficient processor might have a very small value of A, and B would correspond to the rate of increase in power consumption as more and more cores are turned on as the load increases. Unfortunately, for much of the current equipment A is

still a significant part of the total processor power consumption. This includes the fact that the memory system and the peripheral equipment and network connections need to be powered even when no jobs are being processed, and that the operating system can remain active (and hence contributes to the energy consumption) even when there are no external jobs that need to be processed. We can also obtain the expression for the energy consumption per job:

$$E_{job} = \frac{A}{\lambda} + BE[S] \tag{1}$$

where λ is the average arrival rate of jobs and $E[S]$ is the average job service time. The equation supports the principle of concentrating computation on a small number of processing units in order to minimise the power consumption per job.

However just power consumption on its own is not the only important fact: quality of service (QoS) is also primordial. In [1] we discuss how we can achieve optimum energy consumption to QoS trade-offs by adjusting system load in the context of a computing cloud. In this paper we discuss the much broader question: suppose that a mobile community could access services from both a local server within the operator provider (the "local server") and from remote service providers ("remote server"), then what fraction of their workload should they send remotely if they wish to optimise both QoS and energy consumption.

Of course, the decision to use a remote service will depend on a variety of considerations based on security, cost, data and software protection and resilience. Nevertheless, there will also be technical considerations based on QoS and energy consumption per job. Thus this paper only focuses on the technical choice between a local or remote cluster service, and shows that this choice can be formulated as an optimisation problem. In the sequel, we first review the literature, and then provide some experimental measurements regarding the energy consumption and performance of servers. Next we formulate the optimisation problem, describe its solution and present some numerical examples.

1.1. Optimising Energy and QoS

A simple analytic model that uses the combined energy–QoS cost function includes in its first part the well known Pollaczek–Khintchine formula [2,3] for the average response time, based on Poisson arrivals of jobs and general service time distributions, and in its second part the energy consumption per job:

$$C_{job} = aE[S][1 + \frac{\rho(1 + C_S^2)}{2(1 - \rho)}] + b\frac{A}{\lambda} + bBE[S] \tag{2}$$

where $E[S]$ is the average job service time as before; C_S^2 is the squared coefficient of variation of service time; λ is the job arrival rate; and the constants a and b describe the relative importance placed on QoS and energy consumption. This allows us to compute the value of the arrival rate that minimises C_{job}. The result shows that the optimum setting of the load $\rho^* = \lambda^* E[S]$ will depend on A (the idle power consumption) and on the ratio b/a:

$$\rho^* = \sqrt{\frac{2bA}{a(1 + C_S^2)}} \left(1 + \sqrt{\frac{2bA}{a(1 + C_S^2)}}\right)^{-1} \tag{3}$$

The expression (3) gives us a simple rule of thumb for selecting system load for optimum operation, depending on how we weigh the importance of energy consumption with respect to average response time or how fast we are getting the jobs done. We also see that ρ^* increases with the ratio $bA/a(1+C_S^2)$. This tells us that the optimum load should increase with the system's idle power consumption, the relative importance that we place on energy, and with the squared coefficient of variation of service time.

2. Mathematical Model of Energy and Quality of Service at the Local and Remote Cluster

The local cluster (LC) is assumed to incorporate a rack of L processors and related peripheral devices, with a power profile:

$$\Pi_L = A_L + L.B_L\rho_L \tag{4}$$

where:

- A_L is the local power consumption related to the internal networking and shared memory systems (main a secondary) plus their induced cooling and ventilation costs;
- B_L is the workload proportional power consumption, including cooling, per processor in the LC rack, and
- ρ_L is the individual utilisation (percentage of time it is busy) of each of the L processors in the local rack.

The local computational workload is represented by a flow of λ_L jobs per second, each of which on average takes S_L of processing time, and jobs are equally distributed to the L processors, so that:

$$\rho_L = \frac{\lambda_L S_L}{L} \tag{5}$$

As a result, the total expenditure of energy per job in the LC is the ratio of power consumption to total job arrival rate, or:

$$E_L = \frac{A_L}{\lambda_L} + B_L S_L \tag{6}$$

If the average response time $W(F,\lambda)$ is a function of job arrival rate λ and job service time distribution F, we will have:

$$W_L = W(F_L, \frac{\lambda_L}{L}) \tag{7}$$

2.1. The Remote Cluster Model

Similarly, the remote cluster (RC) is assumed to incorporate a rack of R processors and related peripheral devices, with a simplified power profile given by the expression:

$$\Pi_R = A_R + R.B_R\rho_R \tag{8}$$

where:

- A_R is the power consumption in the RC related to the internal networking and shared memory systems (main and secondary) plus the power consumption for cooling and ventilation;

- B_R is the workload proportional power consumption, including cooling, per processor in the RC rack; and
- ρ_R is the individual utilisation (percentage of time it is busy) of each of the R processors in the RC rack.

The computational workload in the RC is represented by a flow of λ_R jobs per second, each of which on average takes S_R of processing time, and jobs are equally distributed to the R processors, so that:

$$\rho_R = \frac{\lambda_R S_R}{R} \tag{9}$$

As a result, the total expenditure of energy per job in the RC is the ratio of power consumption to the total job arrival rate, or:

$$E_R = \frac{A_R}{\lambda_R} + B_R S_R \tag{10}$$

Assuming the same average response time formula $W(F, \lambda)$, function of job arrival rate and job service time distribution, we have:

$$W_R = W(F_R, \frac{\lambda_R}{R}) \tag{11}$$

3. Transferring a Fraction α of Jobs to the Remote Cluster

When the user decides to transfer some fraction α of its jobs to the remote cluster, and assuming that the RC has another load of jobs arriving at rate λ, we obtain that the net average response time perceived by the users who emanate from the LC is:

$$W_U = \alpha W_R(F_R, \lambda + \alpha\lambda_L) + (1-\alpha) W_L(F_L, (1-\alpha)\lambda_L) \tag{12}$$

where we have assumed that additional network delays between the users and the two clusters are equivalent since the users need not be "resident" in the facility that hosts the LC.

Under the assumption that each of the two clusters shares its load equally among its processors, that the RC processors are f times faster than the LC processors, and that all job arrival traffic is Poisson, we can use the well known Pollaczek–Khintchine formula [4] to estimate the average response time per job as a function of the load dispatching policy characterised by the fraction α of jobs that are sent to the RC. We have:

$$\begin{aligned} W_R &= \frac{E[S]}{f}[1 + \frac{\rho_R(1+C_S^2)}{2(1-\rho_R)}] \\ \rho_R &= \frac{[\lambda + \alpha\lambda_L]E[S]}{fR} \end{aligned} \tag{13}$$

and

$$\begin{aligned} W_L &= \frac{E[S]}{f}[1 + \frac{\rho_R(1+C_S^2)}{2(1-\rho_L)}] \\ \rho_L &= \frac{(1-\alpha)\lambda_L E[S]}{L} \end{aligned} \tag{14}$$

The composite cost function that includes both the energy consumption per job and the average response time then becomes:

$$C = b[\alpha E_R + (1 - \alpha)E_L] + aW_U \tag{15}$$

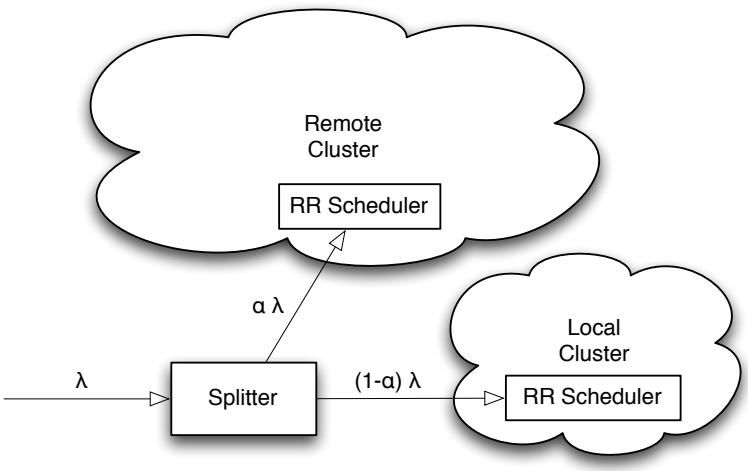
where a and b are the relative importance of response time versus energy.

4. Experimental Results

To validate the main findings of this work, we have conducted a series of experiments using a representative number of computing machines. In particular, we have used six computers ($R = 6$) in the "remote cluster" and three similar computers ($L = 3$) for our "local cluster". The computers were selected from a set of dual core Pentium 4 and quad-core Intel Xeon computers, all of them running Ubuntu Linux with CPU throttling enabled. Job requests were originated from an additional machine connected through a Fast Ethernet switch to both clusters.

A job consisted in calculating the number π, using Machin's formula, to a desired level of precision and in sending the resulting string back to the client over the network. Each job request indicated the desired number of digits to be used, which was randomly chosen in the range 10,000–50,000 by the client. In addition to generating requests periodically (exponentially spaced, at rate λ), the client also determined the cluster that handled the request as is illustrated in Figure 1. With probability α a request was sent to the remote cluster and with $(1 - \alpha)$ to the local cluster.

Figure 1. Experimental system.

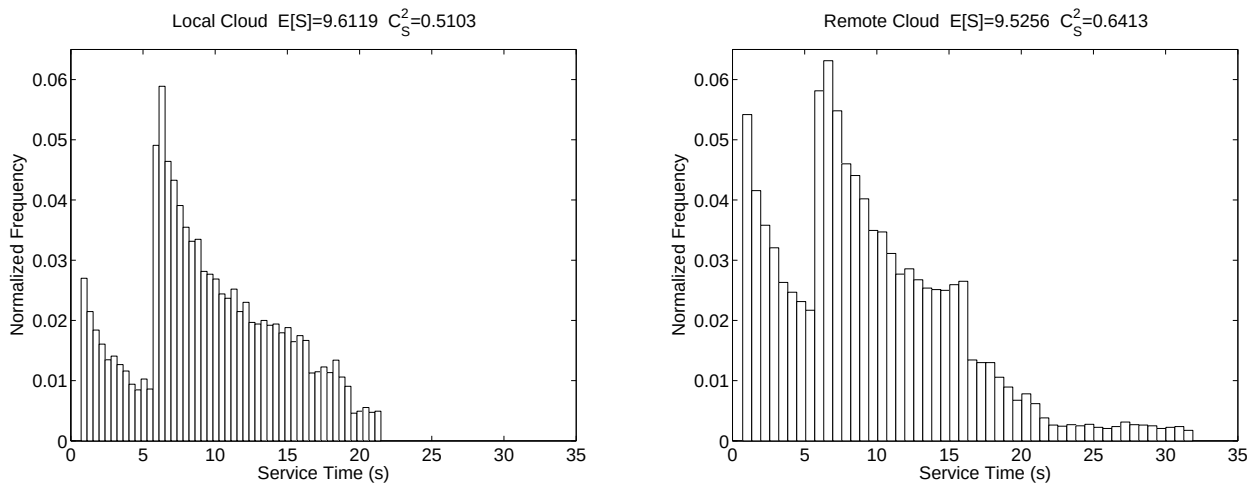


A round-robin scheduler ("RR" in Figure 1) was implemented in each of the clusters, so that each newly arriving job is assigned and placed in the input queue of the next machine in the list regardless of the machine's load. *This in effect results in an equal distribution of the incoming flow of jobs to each of the machines in the cluster, with a separate queue being created at each machine in the cluster, and is reflected in the way in which we construct the mathematical model in Section 2.*

Because of the differences in both number and kind of machines in the two clusters, the job service times varied as shown in Figures 2. The service time in the local cluster was on average 9.6119 and in

the remote cluster 9.5256, giving a speed-up factor of $f = 1.0091$ with corresponding coefficients of variation of 0.5103 and 0.6413.

Figure 2. Normalized histograms of service times. (**a**) Local cluster; (**b**) Remote cluster.

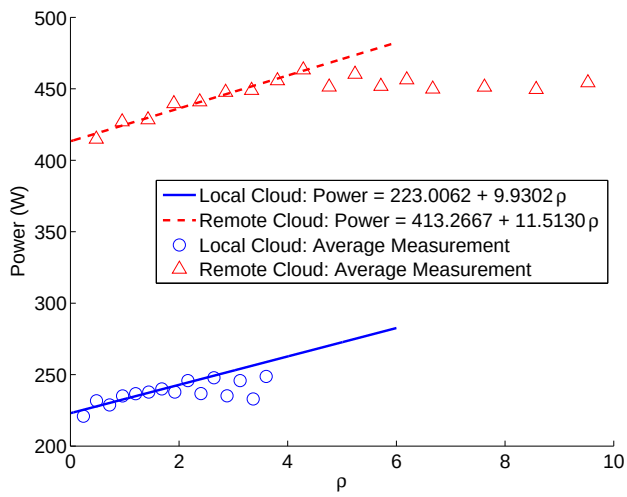


(**a**) (**b**)

The power consumption of both clusters is shown in Figure 3 [5]. From observations of power consumption and system utilization, it was possible to approximate parameters $A_L = 223.0062$, $B_L = 9.9302$, $A_R = 413.2667$ and $B_R = 11.5130$ using linear regression. Note that the linear regressions were applied to each model's operational region, which depends on the number of processors available: 6 and 3 processors respectively for the remote and local clusters. Power measurements beyond the model's validity regions are shown for illustration purposes only.

Clearly, the remote cluster has a higher power consumption because of the larger number of machines available to service jobs.

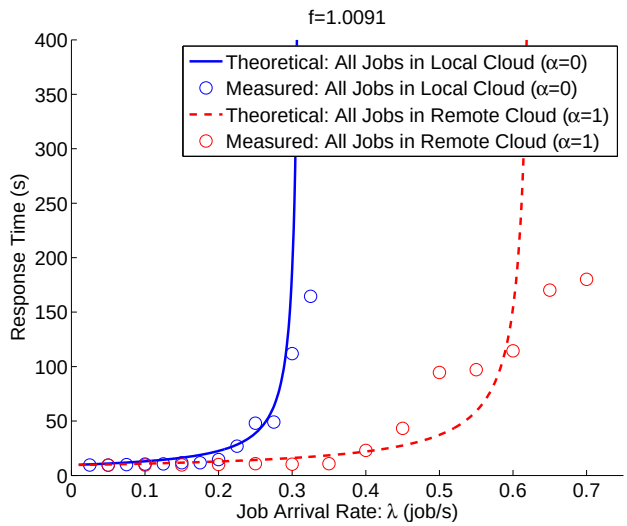
Figure 3. Power consumption of both the local and remote cluster.



By recording the start and completion times for each job at the client, we were able to measure the *average* job response time. We conducted the experiment by using only one of the clusters at the time

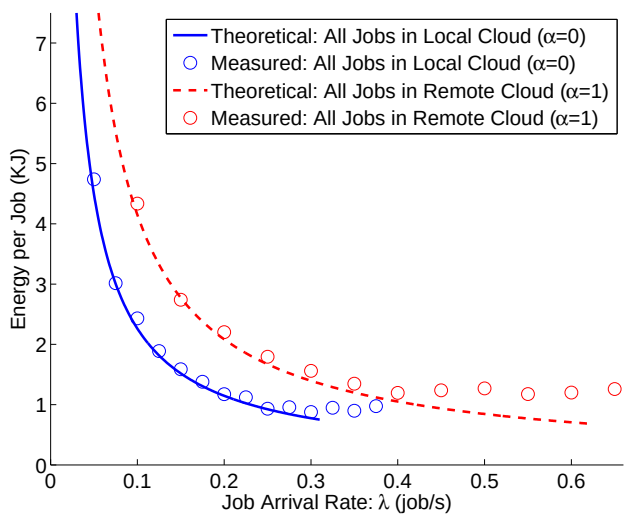
(*i.e.*, by fixing $\alpha = 0$, and later $\alpha = 1$). The results for each independent cluster are shown in Figure 4 alongside the theoretical values. The reported values were obtained by averaging the measurements corresponding to 1000 jobs. The saturation rate for the local cluster was at around $\lambda = 0.3$ and about twice as large for the remote cluster, which makes sense given the relative sizes of the clusters.

Figure 4. Job response time for the local ($\alpha = 0$) and remote cluster ($\alpha = 1$).



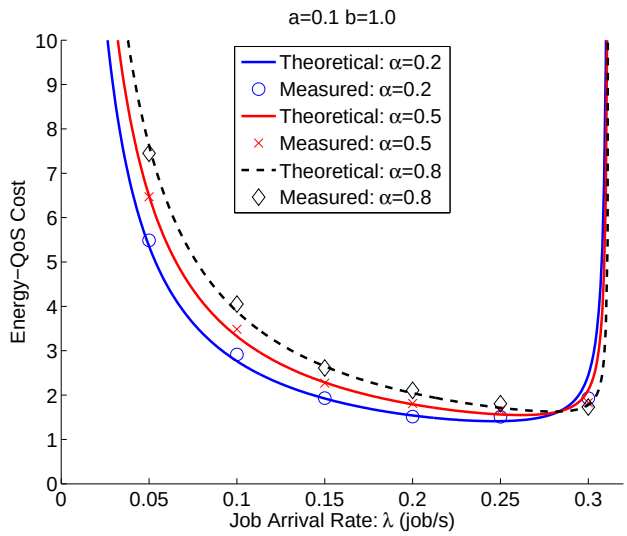
Similarly, we recorded the power consumption while executing the jobs along with the total execution time for each job set, which allowed us to estimate the energy consumption per job (see Figure 5).

Figure 5. Experimental and theoretical values for the energy per job.



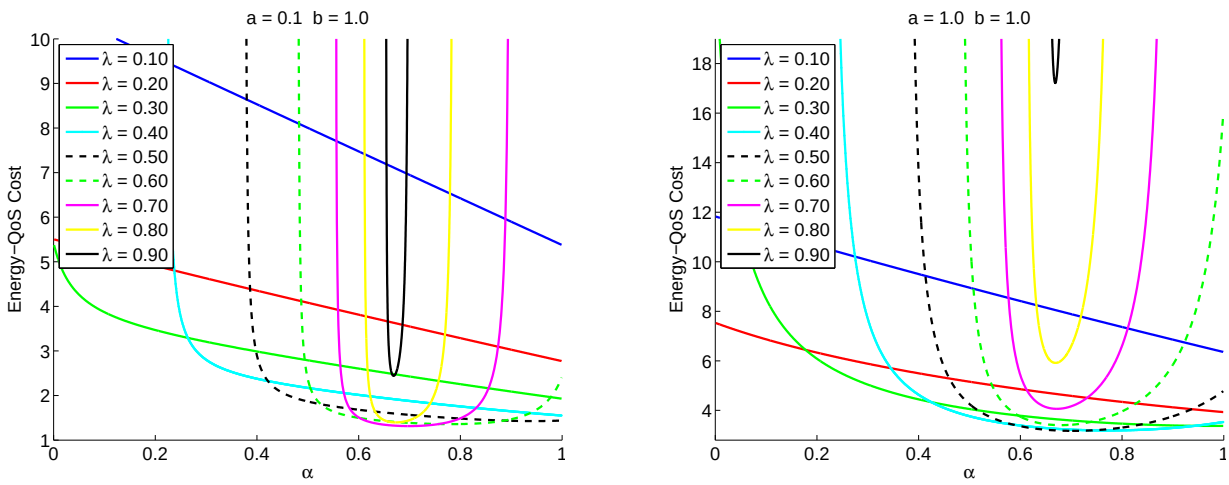
As shown in Figures 4 and 5, the theoretical model approximated well the measured values within the operational range (*i.e.*, when not exceeding the systems' capacity). These models allowed us to obtain energy–QoS costs (Equation 15) for different values of α as shown in Figure 6. The choice of parameters $a = 0.1$ and $b = 1.0$ was made to approximately normalize (to 10) the varying range of the response time and energy per job. The former gets around 100 s close to full system utilization, whereas the latter was around 10 KJ for low system utilization.

Figure 6. Energy–QoS cost *vs.* job arrival rate λ.



It is interesting to observe the values of α that minimize the overall system cost. These are graphically illustrated in Figure 7 for the cases $a = 0.1$, $b = 1.0$ (left), and $a = b = 1.0$ (right). The choice of a and b would in the end depend on the importance that we would like to give to both Energy and QoS. However, these two sets of values will serve us to illustrate the behaviour of the Energy–QoS cost. The horizontal axis depicts values of α in the range 0 to 1. As previously explained, when $\alpha = 0$, all the load is sent to the local cluster. At $\alpha = 1$ all load is sent to the remote cluster. The load is shared between the two clusters for values of α between these two extremes. The figures show that the choice of a and b can affect the cost of the load sharing, as well as the operating point (*i.e.*, the value of λ).

Figure 7. Energy–QoS cost *vs.* α. (**a**) $a = 0.1$ and $b = 1.0$; (**b**) $a = b = 1.0$.



(**a**) (**b**)

4.1. Related Work

Although we have not been able to find work that has discussed the issue that is at the centre of this paper, there has indeed been much work on power aspects of servers and clusters. Most works have

focused on power consumption models offering the advantage of simplicity, but also lack accuracy as suggested by Rivoire *et al.* [6] who examined five full-system representative power models in a recent study. The most common direction is the single-parameter black-box approach, finding relationships between a server's load (normally CPU utilization) and power consumption from measured data. For example, linear regression models have been used by Sasaki *et al.* [7] in web server clusters and Lewis [8] *et al.* in server blades. Fan *et al.* [9] obtained measurements of the power consumption of warehouse-sized computers (computer for large-scale Internet services), Li *et al.* [10] did a similar work on web servers also running on blade systems, and Economou *et al* proposed a modeling methodology for a full-system power consumption [11]. Similar works have been done by Chu *et al.* [12], Jaiantilal *et al.* [13], Yuan and Ahmad [14]. However, these models tend to be accurate only within certain operational regions as suggested by Lien *et al.* [15]. Bolla *et al* [16] investigated the impact in energy consumption and network performance of using low power idle and power scaling in network devices. The simultaneous minimization of a composite energy–QoS function in networks has previously been studied in [17], while other work has considered the dynamically flow of energy so as to support "On Demand" the energy needs of Cloud Computing [18,19].

A direct application of these models is in power saving techniques. A comprehensive survey of green networking research was compiled by Bianzino *et al.* [20]. Another survey on the power and energy management in servers was done by Bianchini and Rajamony [21]. Some relevant examples of power optimization techniques are the works of Sankar *et al.* [22] on metric composition energy-delay, Chase *et al.* [23], who proposed an economic approach to server resource management. Rodero *et al.* researched application-aware power management looking at individual components [24]. A popular approach to reduce power usage is by switching off unused equipment as done by Chen *et al.* [25] and Niyato *et al.* [26]. A control mechanism to adjust the peak power of a high density server was suggested by Lefurgy *et al.* [27] by means of a feedback controller.

5. Conclusions

In this paper, we have studied the optimum load sharing between a local and remote cluster service as a function of a compromise between perceived average response time and energy consumption per job accessed from a mobile. This requires that the average and variance of job service times, average job arrival rates, and the power consumption parameters of the servers involved are effectively measured. We have also provided experimental measurements of these quantities for a test case. Yet much still can be done in this broad area, and some interesting questions that we would like to address include:

- Considering an organisation of servers as a set of specialised service facilities, with multiple specialised units, what are the energy–QoS trade-offs and operating points in such a system?
- With multiple types and distinct steps within jobs themselves, what are the best job allocation [28] strategies for each job type?
- If jobs have synchronisation constraints as in distributed databases [29], how does this affect the energy–QoS trade-off?

- If we wish to simultaneously evaluate multiple QoS and energy criteria [30,31], such as peak power consumption, energy consumption, turn-around times, and throughput, how can we design task allocation and routing algorithms?
- When sub-systems can be turned on and off creating further start-up delays [32] and energy costs, how can we now address the optimum operating point of each sub-system in an interconnected network of servers?

References

1. Gelenbe, E.; Lent, R. Trade-Offs between Energy and Quality of Service. In Proceedings of the Second IFIP Conference on Sustainable Internet and ICT for Sustainability, Pisa, Italy, 4–5 October 2012.
2. Pollaczek, F. Über eine aufgabe der wahrscheinlichkeitstheorie [in German]. *Math. Z.* **1930**, *32*, 64–100.
3. Khintchine, A.Y. Mathematical theory of a stationary queue. *Mat. Sb.* **1932**, *39*, 73–84.
4. Gelenbe, E.; Muntz, R. Probabilistic models of computer systems—Part I. *Acta Inform.* **1976**, *7*, 35–60.
5. Lent, R. A Sensor Network to Profile the Electrical Power Consumption of Computer Networks. In Proceedings of the GLOBECOM Workshops (GC Wkshps), Miami, FL, USA, 6–10 December 2010.
6. Rivoire, S.; Ranganathan, P.; Kozyrakis, C. A Comparison of High-Level Full-System Power Models. In Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, San Diego, CA, USA, 8–10 December 2008.
7. Sasaki, H.; Oya, T.; Kondo, M.; Nakamura, H. Power-Performance Modeling of Heterogeneous Cluster-Based Web Servers. In Proceedings of the 10th IEEE/ACM International Conference on Grid Computing, Banff, Alberta, Canada, 13–15 October 2009.
8. Lewis, A.; Ghosh, S.; Tzeng, N.-F. Run-Time Energy Consumption Estimation Based on Workload in Server Systems. In Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, San Diego, CA, USA, 8–10 December 2008.
9. Fan, X.; Weber, W.-D.; Barroso, L.A. Power provisioning for a warehouse-sized computer. *SIGARCH Comput. Archit. News* **2007**, *35*, 13–23.
10. Li, L.; RuiXiong, T.; Bo, Y.; ZhiGuo, G. A Model of Web Server's Performance-Power Relationship. In Proceedings of the International Conference on Communication Software and Networks, 2009. ICCSN '09, Chengdu, China, 27–28 February 2009.
11. Economou, D.; Rivoire, S.; Kozyrakis, C. Full-System Power Analysis and Modeling for Server Environments. In Proceedings of the Workshop on Modeling Benchmarking and Simulation MOBS, Boston, MA, USA, 18 June 2006.
12. Chu, F.-S.; Chen, K.-C.; Cheng, C.-M. Toward Green Cloud Computing. In Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication, ICUIMC'11, Seoul, Korea, 21–23 February 2011.

13. Jaiantilal, A.; Jiang, Y.; Mishra, S. Modeling CPU Energy Consumption for Energy Efficient Scheduling. In Proceedings of the 1st Workshop on Green Computing, GCM '10, Bangalore, India, 29 Novemer–3 December 2010.
14. Yuan, H.; Kuo, C.-C.; Ahmad, I. Energy Efficiency in Data Centers and Cloud-Based Multimedia Services: An Overview and Future Directions. In Proceedings of the 2010 International Green Computing Conference, Chicago, IL, USA, 15–18 August 2010.
15. Lien, C.-H.; Bai, Y.-W.; Lin, M.-B.; Chang, C.-Y.; Tsai, M.-Y. Web Server Power Estimation, Modeling and Management. In Proceedings of the 14th IEEE International Conference on Networks, ICON '06, Singapore, 13–15 September 2006; Volume 2, pp. 1–6.
16. Bolla, R.; Bruschi, R.; Carrega, A.; Davoli, F. An Analytical Model for Designing and Controlling New-Generation Green Devices. In Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps), Miami, FL, USA, 6–10 December 2010.
17. Gelenbe, E.; Morfopoulou, C. A framework for energy-aware routing in packet networks. *Comput. J.* **2011**, *54*, 850–859.
18. Gelenbe, E. Energy Packet Networks: Adaptive Energy Management for the Cloud. In Proceedings of the 2nd International Workshop on Cloud Computing Platforms, CloudCP '12, Bern, Switzerland, 10 April 2012.
19. Berl, A.; Gelenbe, E.; di Girolamo, M.; Giuliani, G.; de Meer, H. Dang, M.-Q.; Pentikousis, K. Energy-efficient cloud computing. *Comput. J.* **2010**, *53*, 1045–1051.
20. Bianzino, A.P.; Chaudet, C.; Rossi, D.; Rougier, J.-L. A survey of green networking research. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 3–20.
21. Bianchini, R.; Rajamony, R. Power and energy management for server systems. *Computer* **2004**, *37*, 68–76.
22. Sankar, S.; Vaid, K.; Rogers, H. Energy-Delay Based Provisioning for Large Datacenters: An Energy-Efficient and Cost Optimal Approach. In Proceedings of the Second Joint WOSP/SIPEW International Conference on Performance Engineering, ICPE '11, Karlsruhe, Germany, 14–16 March 2011.
23. Chase, J.S.; Anderson, D.C.; Thakar, P.N.; Vahdat, A.M.; Doyle, R.P. Managing Energy and Server Resources in Hosting Centers. In Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, SOSP '01, Banff, Canada, 21–24 October 2001.
24. Rodero, I.; Chandra, S.; Parashar, M.; Muralidhar, R.; Seshadri, H.; Poole, S. Investigating the Potential of Application-Centric Aggressive Power Management for Hpc Workloads. In Proceedings of the 2010 International Conference on High Performance Computing (HiPC), Dona Paula, Goa, India, 19–22 December 2010.
25. Chen, G.; He, W.; Liu, J.; Nath, S.; Rigas, L.; Xiao, L.; Zhao, F. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, NSDI'08, San Francisco, CA, USA, 16–18 April 2008.
26. Niyato, D.; Chaisiri, S.; Sung, L.B. Optimal Power Management for Server Farm to Support Green Computing, Cluster Computing and the Grid. In Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, Shanghai, China, 18–21 May 2009.

27. Lefurgy, C.; Wang, X.; Ware, M. Server-Level Power Control. In Proceedings of the Fourth International Conference on Autonomic Computing ICAC '07, Jacksonville, FL, USA, 11–15 June 2007.
28. Aguilar, J.; Gelenbe, E. Task assignment and transaction clustering heuristics for distributed systems. *Inf. Sci. Inform. Comput. Sci.* **1997**, *97*, 199–221.
29. Gelenbe, E.; Sevcik, K. Analysis of update synchronisation algorithms for multiple copy databases. *IEEE Trans. Comput.* **1979**, *C-28*, 737–747.
30. Atalay, V.; Gelenbe, E. Parallel algorithm for colour texture generation using the random neural network model. *Int. J. Pattern Recognit. Artif. Intell.* **1992**, *6*, 437–446.
31. Gelenbe, E.; Fourneau, J.M. Random neural networks with multiple classes of signals. *Neural Comput.* **1999**, *11*, 953–963.
32. Gelenbe, E.; Iasnogorodski, R. A queue with server of walking type. *Ann. Inst. Henri Poincaré Sect. B*, **1980**, *16*, 63–73.