Power Management of Server Farms

Giuseppe Iazeolla and Alessandra Pieroni

“TorVergata” and “Guglielmo Marconi” Universities, Roma, Italy
{giuseppe.iazeolla, alessandra.pieroni}@uniroma2.it

Keywords: server farm, server farm modeling, server farm QoS, server farm power management, server farm performance management, server farm power and performance management.

Abstract

The power management of server farms (Sf) is becoming a relevant problem in economical terms. Server farms totalize millions of servers all over the world that need to be electrically powered. Research is thus expected to investigate into methods to reduce Sf power consumption. However, saving power may turn into waste of performance (high response times), in other words, into waste of Sf Quality of Service(QoS). By use of a Sf model, this paper investigates Sf power management strategies that look at compromises between power-saving and QoS. Various optimizing Sf power management policies are studied combined with the effects of job queueing disciplines. The (policy, discipline) pairs, or strategies, that optimize the Sf power consumption (minimum absorbed Watts), the Sf performance (minimum response time), and the Sf performance-per-Watt (minimum response_time-per-Watt) are identified. By use of the model, the work the server-manager has to do to direct his Sf is greatly simplified, since the universe of all possible (π, δ) strategies he needs to choose from is drastically reduced to a very small set of most significant strategies.

1. Introduction

Server farms have become common and essential to the functioning of business, communications, academic, and governmental systems. During the past years, increasing demand for Sf services has led to significant growth in the number of Sf, along with an estimated doubling in the energy used by the Sf and the power and cooling infrastructure that supports them. In the US, power absorbed by Sf is estimated in more than 100 Billion kW, for an expenditure of $8 Billions a year, that corresponds to the expenditure in electricity of about 17 Million homes [6]. This US-local Sf problem becomes a global problem when seeing at power consumption by web companies, say Google, Amazon, Yahoo etc. The number of Google servers, for example, will reach [7] an estimated 2,376,640 units by the end of 2013. Assuming a busy server absorbs around 240 W of power, Google will absorb around 600 MW of electrical power by the end of year 2013. Research is thus expected to investigate into methods to reduce the Sf absorbed power. To do that, one may decide to adopt policies to periodically switch-off servers when they are in an idle-state. Such policies, however, are to be sufficiently intelligent not to degrade the Sf Quality of Service (QoS). Indeed, returning an off-server to its on state requires spending a non-negligible amount of time (called setup-time) that makes the server slower to respond to customer requests. This may turn into Sf lowQoS, such as low response to web queries, unsatisfactory VoIP communications and limited streaming of data. Any research in Sf power management should thus look at compromises between power-saving and QoS.

This paper is structured as follows: Sect.2 introduces the modeling principles for Sf power and QoS management, Sect. 3 introduces the Sf power and QoS evaluation indices, and Sect. 4 identifies the optimal power and QoS management strategies.

2. Modeling the server farm power and QoS management

Let us denote by server management policy π the criterion according to which the server is switched between the on, the off and the idle state. Let us instead denote by queueing discipline δ the criterion according to which the servers pick jobs from the waiting queue when they become idle.
Most of the power absorbed by the servers of a farm is wasted. Indeed, due to over-provisioning, servers are found busy (i.e. making processing work) only 20% to 30% of the time, on average. So, energy saving requires the adoption of management policies to avoid powering the servers when they are not processing. In other words, policies to decide in which state (idle or off) to keep the servers when not busy.

Two families of server management policies are considered in literature [2, 3, 6, 9]: static and dynamic policies. Dynamic policies are policies that assume time-varying demand patterns (i.e.: job arrival rates $\lambda(t)$ changing over time). Such policies adapt themselves to the changing $\lambda(t)$. Static policies, instead, are defined for a given $\lambda$, i.e. not changing with $t$. In a previous paper [11], static and dynamic policies that optimize the $S_f$ power consumption and performance have been considered. In the paper, however, only the effects of various policies $\pi$ are studied, with no consideration of the effects of the queueing disciplines $\delta$.

In this paper the static case is considered, however the effect of both policies $\pi$ and disciplines $\delta$ is investigated, and the strategy or combination ($\pi, \delta$) that optimizes the following indices:

- farm power consumption (minimum absorbed Watts),
- farm performance (minimum response time),
- farm performance-per-Watt (minimum response_time-per-Watt)

is studied.

A busy-server in the on state absorbs around 240W, an idle-server about 160W and an off-server zero W. So why not to keep in the idle state or in the off state the servers when not busy? Just since switching a server from off to on consumes a time-overhead (the so-called setup-time). Thus, any power-saving policy may result in time-wasting problems. As a consequence, the server farm may lose performance (increased response time to the incoming jobs, low throughput of VoIP and streaming packets, etc.) and its QoS becomes unacceptable to customers.

Assume the farm consists of $n$ servers. Various static policies for an $n$ serve farm have been investigated in literature [2, 3, 6, 9]. Authors in [2], in particular, study three different static policies $\pi$ to manage server farms: the On/Idle (or NeverOff) policy, the On/Off (or InstantOff) policy and the On/Off/Sleep (or Sleep) policy.

Under the On/Idle policy, servers are never turned off. All servers are either on or idle, and remain in the idle state when there are no jobs to serve. If an arrival finds a server idle it starts serving on the idle server. An arriving job that finds all $n$ servers on (busy) joins a central queue from which the servers pick jobs when they become idle.

Under the On/Off policy, instead, servers are immediately turned off when not in use, thus yielding power-saving with respect to the On/Idle. As said above, however, there is a setup cost (in terms of time-delay and of additional power penalty) for turning-on an off server, and this yields an increase in response-time.

Fig. 1 compares the On/Off and the On/Idle policies [6] in an example case.

The On/Idle policy proves to be better in terms of response time (11 versus 39 sec), since the incoming jobs do not suffer by setup time delays, but involves a larger amount of power waste with respect to the On/Off policy (780 versus 320 W), since of the amount of power an idle server absorbs with respect to an off one.

The morale is that to reduce power consumption one has to pay a response time increase.

Fig. 1: Experimental results for policy comparison(n=4 servers, $\rho = 0.3$, $T_{\text{setup}}= 200$sec, $S = 7$sec) [6].
In the Fig.1 case, to reduce power consumption from 780W to 320W, we pay a response time increase from 11 to 39 sec.

Finally, the On/Off/Sleep policy is similar to the On/Off, except that whenever the server goes idle it goes into a sleep state, where it remains until there is no work to process and begins to wake up as soon as work arrives.

Fig. 2: The considered Sfqueueing model

Fig. 2 illustrates the Sfqueueing model, where $S$ denotes the average job size (or the job mean service time) and $\mu = 1/S$ denotes the server average service rate, while $T$ denotes the Sfmean response time.

Quantity $\rho = \lambda / n \mu$ denotes the server farm load. It is known [9] that, for system stability, the condition $0 \leq \rho < 1$ is to be satisfied, in other words the condition $0 \leq \lambda < n \mu$.

We shall use the following notation to define the Sf parameters: $T_{\text{setup}} = \text{server mean setup time}$, $P_{\text{on}} = \text{busy-server power absorption in the on state}$, $P_{\text{idle}} = \text{idle-server power absorption}$, $P_{\text{off}} = \text{off-server absorption}$, and $S = \text{average job service time}$.

3. Server farm power and QoS evaluation indices

As said above, we denote by policy $\pi$ the server management policy and by queueing discipline $\delta$ the criterion according to which the servers pick jobs from the waiting queue when they become idle. Policies, policies $\pi = \text{On/Off}$, $\pi = \text{On/Idle}$, $\pi = \text{On/Off/Sleep}$ and disciplines $\delta = \text{time-independent}$ and $\delta = \text{time-dependent}$ disciplines will be considered. Time-independent disciplines are conventional disciplines, such as FIFO (First In First Out), LIFO (Last In First Out), RAND (Random extraction). Such disciplines are also called abstract disciplines. Time-dependent disciplines, instead, are disciplines in which the servers pick jobs from the waiting according to job size, in other words to their service time $S$. An example of such a discipline is the SPTF (Shortest Processing Time First) [10] also called SJF (Shortest Job First) [9], in which servers pick jobs of shortest size first. As said above, to reduce the server farm power consumption, one has to pay a debt of increase in average response time.

It is known [9, 10] that, in queueing systems, the SPTF minimizes the system mean response time.

So, we conjecture that by using the SPTF discipline in server farm systems, the Sfdebtto pay inresponse time is smaller than with abstract disciplines.

Such a conjecture will be proved in the paper. Moreover, we shall see that, in some circumstances, the response-time benefit one may obtain by moving from the FIFO to the SPTF discipline is larger than the benefit one may obtain by moving from the On/Off to the On/Idle policy.

Disciplines $\delta = \text{FIFO}$ and $\delta = \text{SPTF}$ will be considered in the paper. For any given $(\pi, \delta)$ strategy, i.e. policy and discipline combination, we shall use the following notation for the Sf power and QoS indices:

- $P(\pi, \delta)$ the long-run average power absorbed by the Sf under the $(\pi, \delta)$ strategy;
- $T(\pi, \delta)$ the Sf mean response time under the $(\pi, \delta)$ strategy;
- $PT(\pi, \delta) = P(\pi, \delta) \cdot T(\pi, \delta)$ the mean power by response-time product under the $(\pi, \delta)$ strategy.
Minimizing the PT can be seen as maximizing the performance per Watt [2], with performance been defined as the inverse of the mean response time. Our goal is to seek rules to simplify the role of the Sf manager, by reducing the space of all possible \((\pi, \delta)\) strategies he needs to choose from. The \((\pi, \delta)\) strategies will be studied by also considering the effects of the Sf load \(\rho\) and of the Sf mean setup-time \(T_{\text{setup}}\). In particular:

1. To stress the effects of the queueing discipline \(\delta\), the low \(\rho\) (\(\rho \leq 0.5\)) and the high \(\rho\) (\(\rho \rightarrow 1\)) conditions will be considered. Indeed, when \(\rho\) is low the Sf queue remains empty and thus the response time benefits of the SPTF discipline with respect to the FIFO are negligible.

2. To stress the effects of the power management policy \(\pi\) the low setup-time (\(T_{\text{setup}} \leq 1s\)) and the high setup-time (\(T_{\text{setup}} \geq 100\) s) conditions will be considered. Indeed, when the \(T_{\text{setup}}\) is low, the jobs do not suffer setup delays and thus the response-time benefits of the On/Idle policy with respect to On/Off remain negligible.

Next section discusses results obtained for various \((\pi, \delta)\) Sf strategies, and various Sf loads \(\rho\) and Sf setup-times \(T_{\text{setup}}\).

4. Seeking the optimal \((\pi, \delta)\) strategy for the server farms

Under the assumption of Poisson arrivals, exponential service times and deterministic setup times, authors in [2] prove that the optimal, or nearly optimal, PT(\(\pi, \delta\)) holds for the combination \((\pi, \text{ FIFO})\) with \(\pi\) being one discipline from the set \{On/Off, On/Idle, On/Off/Sleep\}. In other words, according to their results, there is no need to consider other policies than the On/Off, the On/Idle and the On/Off/Sleep policy.

They however only study the effects of moving from one policy \(\pi\) to another, without paying attention to the effects of also moving from a \(\delta=\text{FIFO}\) discipline to another discipline. Under the FIFO assumption, however, they find that the On/Idle policy is typically superior to the remaining two in terms of PT(\(\pi, \delta\)), and that these two (i.e. the On/Off and On/Off/Sleep policies) show practically similar behaviors.

Our aim is to extend such results by studying the effects of the queueing discipline \(\delta\), both on the PT(\(\pi, \delta\)) index and on the P(\(\pi, \delta\)) and T(\(\pi, \delta\)) indices separately.

Since of the mentioned similar PT(\(\pi, \delta\)) behavior of the On/Off and On/Off/Sleep policies with respect to the On/Idle one, we shall limit our study to the comparison of the On/Off and On/Idle policies under different queueing disciplines \(\delta\). More precisely, the following four \((\pi, \delta)\) strategies are investigated in the paper:

1. (On/Idle, FIFO)
2. (On/Idle, SPTF)
3. (On/Off, FIFO)
4. (On/Off, SPTF)

and for each of such strategy the P(\(\pi, \delta\)) product is studied, besides the P(\(\pi, \delta\)) and T(\(\pi, \delta\)) indices. As said above, for each of those strategies, two largely different setup times (\(T_{\text{setup}} = 1s\) and \(T_{\text{setup}} = 100s\)) will be used to stress the effect of the setup time on the On/Idle and On/Off polices.

Similarly, two largely different server farm loads, low \(\rho\) (\(\rho \leq 0.5\)) and high \(\rho\) (\(\rho \rightarrow 1\)) will be used to stress the effect of the queueing disciplines.

The following Sf characteristics are assumed: server mean setup-time \(T_{\text{setup}} = 1\) sec (or 100 sec), server \(P_{\text{on}} = 240W\), server \(P_{\text{setup}} = 240W\), server \(P_{\text{idle}} = 150W\), server \(P_{\text{off}} = \text{zeroW}\), mean job service time \(S = 1\) sec, \(n = 30\) servers.
Tab. 1 shows simulation results [8] that compare the Sf power and QoS indices in the low setup case (T_{\text{setup}} = 1s):

- Seeing at the power consumption P, we note that there is no effect by the queueing discipline δ on the power consumption P, while there is an effect by the policy π for low ρ. Indeed, a drastic reduction can be seen (from 6000W to 4200W, for low ρ) when moving from On/Idle to On/Off, since when ρ is low, the waiting queue is almost empty and thus a large number of servers is in the off state. For high ρ instead, the power consumption P remains unchanged (P = 7100W) with the discipline δ since the queue is always full and thus the servers remains always in the on state.

- Seeing at the response-time T, we note that there is an effect both by the queueing discipline δ and by the policy π. The effects hold both for low ρ and for high ρ. In the On/Idle case, when ρ is low, there is no waiting in the Sf queue and thus the mean response time T is of about the mean job service time (S = 1sec) while it increases (T = 1.8sec for δ = FIFO and T = 1.3sec for δ = SPTF) for high ρ. In the On/Off case, when ρ is low, the mean response time is higher (T = 1.2sec with no effect by the discipline, the queue is empty), since almost every arrival finds servers in the off state, and thus every job incurs in the setup time. For high ρ, instead, the mean response time increases (T = 2sec for δ = FIFO and T = 1.35sec for δ = SPTF) due to large queueing. As predicted above, we can see that the benefit in response-time one may obtain moving from FIFO to SPTF is larger than the one obtainable moving from On/Off to On/Idle. Indeed (see high ρ) moving from the (On/Off, FIFO) strategy to the (On/Idle, FIFO) the response time T changes from 2 to 1.8 (a 10% reduction). Moving, instead, from the (On/Idle, FIFO) strategy to the (On/Idle, SPTF) the response time T changes from 1.8 to 1.3 (an almost 30% reduction).

- Seeing at the PT index, its values are a consequence of the P and the T ones. Tab.1 shows that the optimal PT is obtained for the (On/Off, FIFO) strategy and for the (On/Off, SPTF) strategy when ρ is low, while it is obtained for the (On/Idle, SPTF) strategy only when ρ is high.

Tab.2 shows simulation results [8] that compare the server farm the Sf power and QoS indices in the high setup case (T_{\text{setup}} = 100sec):

- Seeing at the power consumption P the table shows that there is no effect by the queueing discipline δ on the power consumption P, while there is an effect by the policy π for low ρ. Contrary from the low setup case (Tab.1), we see an increase of P (from 6000W to 6500W, for low ρ) when moving from On/Idle to On/Off, since, even though the queue is almost empty, the high setup time plays a preeminent role in the power consumption for low ρ. Indeed, as seen in the Sf parameters above, P_{\text{setup}} is higher than P_{\text{idle}}. For high ρ instead, the power consumption P remains unchanged (P = 7100W) with the discipline since the queue is always full and thus the servers remains always in the on state.
Tab. 2 Server farm results for high setup-time \( T_{\text{setup}} = 100s \)

<table>
<thead>
<tr>
<th>( \pi, \delta )</th>
<th>( P(\pi, \delta) ) ( (W) )</th>
<th>( T(\pi, \delta) ) ( (sec) )</th>
<th>( PT(\pi, \delta) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho = 0.5 )</td>
<td>( \rho \rightarrow 1 )</td>
<td>( \rho = 0.5 )</td>
<td>( \rho \rightarrow 1 )</td>
</tr>
<tr>
<td>On/Idle FIFO</td>
<td>6000</td>
<td>7100</td>
<td>1</td>
</tr>
<tr>
<td>On/Idle SPTF</td>
<td>6000</td>
<td>7100</td>
<td>1</td>
</tr>
<tr>
<td>On/Off FIFO</td>
<td>6500</td>
<td>7100</td>
<td>12</td>
</tr>
<tr>
<td>On/off SPTF</td>
<td>6500</td>
<td>7100</td>
<td>3.4</td>
</tr>
</tbody>
</table>

- Seeing at response-time (T) there is an effect both by the queueing discipline \( \delta \) and by the policy \( \pi \). The effect holds both for low \( \rho \) and for high \( \rho \). In the On/Idle case, when \( \rho \) is low, there is no waiting in the Sf queue and thus the mean response time T is of about the mean job service time (S = 1sec), while it increases (T = 1.8 sec for \( \delta = \) FIFO and T = 1.3sec for \( \delta = \) SPTF) for high \( \rho \).

  In the On/Off case, when \( \rho \) is low, the mean response time is higher (T = 12sec for FIFO, that reduces to 3.4 sec for SPTF due to the high setup time) since almost every arrival finds the servers in the off state, and thus every job incurs in the setup time. For high \( \rho \), instead, the mean response time increases (T = 84 sec for \( \delta = \) FIFO and T = 16.5 sec for \( \delta = \) SPTF) due to large Sf queueing.

  In this high setup case, the effect of the policy \( \pi \) shows to be dominant with respect to the effect of the discipline \( \delta \). Indeed, moving from the (On/Off, FIFO) strategy to the (On/Idle, FIFO) the response time T changes from 84 sec to 1.8 sec (an almost 97% reduction), while moving from the (On/Off FIFO) strategy to the (On/Idle, SPTF) the response time T changes from 84 sec to 16.5 sec (an almost 80% reduction).

- Seeing at the PT index, its values are a consequence of the P and the T ones. Tab.2 shows that the optimal PT is obtained for the (On/Idle, FIFO) and the (On/Idle, SPTF) strategies when \( \rho \) is low, and it is obtained for the (On/Idle, SPTF) strategy only when \( \rho \) is high.

In summary, making predictions of the Sf management strategy, that optimizes

- the Sf power consumption (minimum absorbed Watts), or
- the Sf performance (minimum response time), or
- the Sf performance-per-Watt (minimum response_time-per-Watt)

is a non trivial task. The most significant policies \( \pi \) are first to be drawn from the universe of all possible policies. Then, for each such a policy, the effects of time-dependent and time-independent queueing disciplines are to be studied. On the other hand, once the modeling work has been done, the work the server-farm manager has to perform to direct his Sf is greatly simplified, since the universe of all possible (\( \pi, \delta \)) strategies he needs to choose from is drastically reduced to very a small set of most significant strategies.

Summary

The concept of server farm (Sf) management strategy \( (\pi, \delta) \) has been introduced, with a queueing model of the Sf to study the combined effect of the Sf power management policy \( \pi \) and of the Sf queueing discipline \( \delta \), in order to evaluate various Sf power and QoS indices: 1) the optimal Sf power consumption, or minimum absorbed Watts, 2) the optimal Sf performance, or minimum response time, and 3) the optimal Sf performance-per-Watt, or minimum response_time-per-Watt.

The set of the most significant policies \( \pi \) has been drawn from the universe of all possible policies, and for each such policy, the Sf model has been used to predict the effect of time-dependent and time-independent queueing disciplines. Model simulation results give the best management strategies the server-farm manager may use to minimize the farm power consumption while maintaining its QoS to its best.
Acknowledgements
This work was partially supported by the PhD research program of the University of Roma TorVergata, funded by Telecom Italia, and by the University Guglielmo Marconi in Roma.

References