Noname manuscript No.
(will be inserted by the editor)

Separating internal and external fluctuation in distributed Web-based services

Sara Casolari · Marco Villani · Michele Colajanni · Roberto Serra

Abstract The observable behavior of a complex system reflects the mechanisms governing the internal interactions between the system's components and the effect of external perturbations. We investigate the behavior of a distributed system providing Web-based services and the effects of the impact of external request arrivals on the internal system resources; the results of our study are of primary importance for taking several runtime decisions on load and resource management. Here we show that by capturing the simultaneous activities of several performance indexes of the Web-based system nodes we can separate the internal dynamics from the external fluctuations. For every internal performance index, we are able to determine the origin of fluctuations, finding that while all the considered performance indexes of the application server have robust internal dynamics, the CPU utilization and the network throughput of the Web and database servers are mainly driven by external demand.

1 Introduction

Almost all interesting processes in nature are highly cross linked. In many systems, however, the basic entities interact nonlinearly to form emergent structures and functions (a process often referred to as "hierarchical self-organization"). Examples of these systems are gene networks [15,22,24], metabolic networks [12], food webs [17], neural networks [25], Internet [10], Web [16], ecological networks [13], social networks [11,20,21].

Also technological systems display emergent self-organization processes, which have been investigated mostly in large-scale systems like the Internet [10], the WWW [16], power distribution networks [1] and others (note however that the structure of these systems emerges from the interaction of millions of human beings and is therefore a function of social interactions and not only the outcome of some explicit technical design).

Sara Casolari, Michele Colajanni
Department of Information Engineering, University of Modena and Reggio Emilia
E-mail: {sara.casolari,michele.colajanni}@unimore.it

Marco Villani, Roberto Serra
Deparment of Social, Cognitive and Quantitative Sciences, University of Modena and Reggio Emilia E-mail: {marco.villani,roberto.serra}@unimore.it

At the same time, there are smaller socio-technical systems whose components and architectures are explicitly designed and that nevertheless show not predictable complex behaviors. In the literature, to the best of our knowledge, there are no studies which make use of methods from complex systems science to analyze systems composed by relatively few computers, probably because the presence of few nodes suggests that the classical engineering techniques can adequately describe and control their behavior.

On the contrary, there are technological systems consisting of a number of interconnected servers, clearly affected by the pattern of human usage and at the same time highly influenced by their internal courses of action, the two classes of processes being highly and inextricably interrelated. In this work we investigate the behavior of a distributed system providing Web-based services and the effects of the impact of external request arrivals on the internal system resources, a study crucial for taking several runtime decisions on load and resource management [2].

The problem is hard, because of the presence of a very high variance and non stationary characteristics of the involved entities. In addition, the problem is even worsened by the frequent and unpredictable interactions between the artificial system and the human beings that desire to add and/or extract information from it.

The introduction of the methods of complex systems science could lead to a better comprehension of the system processes, of the entities really involved in the interaction and of the feedbacks active at all the subsystems levels. In effect, complex systems have many remarkable features which escape from the usual engineering viewpoint. The final goal of this process still to be achieved could be the identification of new levels of description able to host new rules, aiming to carry new kinds of control parameters. The present study aims at providing useful indications in this direction.

In this paper we are interested in separating the load caused by internal and external processes, a not obvious study that could have significant applications on the runtime management and control of these systems (see Section 2). The approach we use in order to interpret the data, which is described in Section 3, was initially developed by de Menezes and Barabasi [19] for systems which differ from our because of the presence of a conserved quantity. Here we generalize the use of a measure proposed in [19] and we show that it is useful to identify internally vs. externally generated load also in our system. Section 4 contains the description of the testbed and Section 5 shows the achieved results. Section 6 reports the main indications coming from our studies.

2 Problem definition and relevant questions

The advent of large infrastructures providing any kind of service through Web related technologies has changed the traditional processing paradigm. One interesting aspect of the new applications is that they are not executed in isolation from the external world, and multiple heterogeneous activities are required even to achieve a single task, usually in a coordinated fashion. Unlike traditional computing, the modern infrastructures must accommodate varying demands for different types of processing within certain time constraints. The majority of critical Internet-based services run on distributed infrastructures that have to satisfy scalability and availability requirements and have to avoid performance degradation and system overload. In these contexts, runtime management and control are becoming extremely complex, because they are a function not only of individual applications, but also of their interactions as they contend for processing and I/O resources, both internal and external. This paper focuses on the problem of investigating the impact of external request arrivals on the internal system resources: this study is crucial for taking several runtime decisions on load and resource management.

Among the different goals, this study is fundamental to investigate new strategies for controlling the system and guaranteeing more stable behavior through several mechanisms: load balancing and load sharing [3], overload and admission control [18], job dispatching and redirection even at a geographical scale [8]. The advent of self-adaptive systems and autonomic computing [26] will further increase the necessity for runtime management models that are able to evaluate the system behavior and take important actions on the basis of internal and external factors.

There are many issues which make this problem interesting, difficult and not deeply investigated. In fact, there are several studies and models representing the external factors. They show that the traffic reaching Internet-based services is characterized by heavy-tailed distributions [4], bursty arrivals [14] and hot spots [5] that are difficult to predict and to manage because the server side has no or little control on the client side.

Models on internal system factors have received less attention. We can observe that in the considered Web-based context any user click causes a large number of arrivals at a open system. In their turn, these arrivals activate a possibly huge number of requests to the internal hardware and software resources with unpredictable mutual interactions and interdependency. As a consequence, it is becoming impossible to understand, for example, where the system load really is and where the system is going from monitored raw data, such as snapshots of the CPU utilization, request arrivals, average number of queued processes, memory utilization. All opinions agree to indicate that the complexity of the middleware and applications for the support of Web-based systems is continuously increasing, but there is only some preliminary characterization of the internal characteristics of these complex systems [2].

We claim that for taking adequate runtime decisions, we should be able to understand the impact of the external and the internal factors on the system behavior that is observed through the time series coming from the resource measures of a Web-based system. Different nodes in this system are influenced to a different degree by these internal and external factors, thus making it impossible for a runtime decision system that has access only to a single node to separate the internal dynamics from the externally imposed arrivals. Here, we propose a method to separate for each node the external from the internal contributions, in order to pave the way of taking runtime management decisions for the sake of the system stability.

3 Internal and external fluctuations in network dynamics

Nowadays the most important critical services are supported by locally distributed architectures characterized by multiple layers of servers. Figure 1 shows an example of a typical Web cluster infrastructure where the main functions of request management, service generation and data repository are carried out by a Web interface layer, an application layer and an information repository layer, respectively.

The Web technologies nodes represent the system interface that accepts HTTP connection requests from the clients, serves static content from the file system and, when necessary, activates the application nodes. The application layer is represented by a multitude of servers running different processes. They handle all the business logic

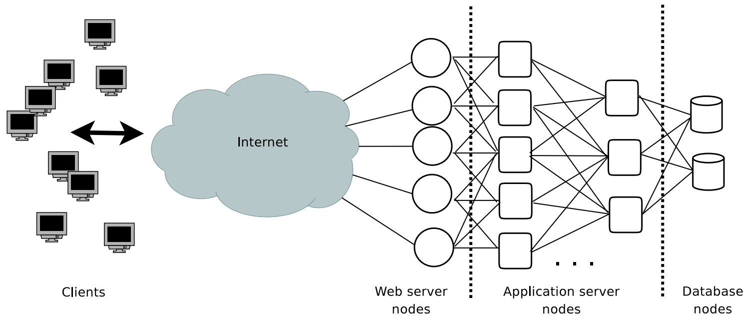


Fig. 1: A typical architecture supporting Web-based services

of the Web-based services and, when necessary, retrieve the information by interacting with the database nodes. These nodes at the back-end layer typically consist of database servers and information storages.

A similar system can be intrinsically represented through a dynamic network. To this purpose we adopt the methodology which was proposed by de Menezes and Barabasi for the analysis of dynamical networks oriented to decoupling the systems internal collective dynamics and the changes in the external environment [19]. They were able to evaluate relevant scaling exponents by applying the same analytical methods to different kinds of systems, such as the flow of water through an US river basin, the Colorado highway system of Colorado, and the Internet traffic. By following the movements of the users through a net representation of these systems, the authors note that there is some conserved quantity for any kind of users (water, cars, or packets).

Not all the assumptions in [19] are valid in the considered Web-based system because the external requests generate a different computational load at each node, and each node may be the source of new information exchanges while in [19] the internal interactions are preserved. Moreover, the nodes in Figure 1 are not homogeneous as in [7, 23]. Nevertheless, our context shares fundamental similarities with the systems analyzed in [19], and it is the main goal of this paper to verify whether that model can be applied even to a more general context. In all the considered models there is an exchange of flow through a system composed by nodes and links; an external environment interacts with the system and forces a variable external load; at the same time, the internal processes may influence and in our case generate internal tasks.

By taking into account the results in [19], we model the system in Figure 1 as a network of nodes where N external client requests (the so called "walkers") have to perform a given number of hops through the nodes of the system, starting from the Web server nodes. We represent the behavior of each node by monitoring the load of a system internal resource that is representative for that node (for example, the CPU utilization, the disk and network throughput). These measures denote time series where the dynamics are determined by the interactions of the N walkers that pass from one node to another through the interconnection network. As in [19], we assume that the dynamics of each node of the model is determined by two factors:

– the component interactions are governed by some internal dynamical rules that distribute the activities among the various parts of the system;

– there are external variations in the overall activity of the system. They are induced, for example, by variations in the number of Web users and in the request type.

As each node is influenced by internal and external factors and by the interactions with other nodes, in order to separate the internal dynamics from the external fluctuations it is mandatory to have a global view of the system. From these considerations, we assume that each time series that is related to some measures referring to the node i is composed by the sum of internal and external contributions:

$$f_i(t) = f_i^{int}(t) + f_i^{ext}(t) \tag{1}$$

where the external contribution can be estimated by:

$$f_i^{ext}(t) = A_i \sum_{i=1}^{N} f_i(t) \tag{2}$$

$$A_i = \frac{\sum_{t=1}^{\tau} f_i(t)}{\sum_{t=1}^{\tau} \sum_{i=1}^{N} f_i(t)} \tag{3}$$

Hence, we can represent the internal contribution as:

$$f_i^{int}(t) = f_i(t) - f_i^{ext}(t) \tag{4}$$

The respective standard deviations are given by:

$$\sigma_i^{int} = \sqrt{\langle f_i^{int}(t)^2 \rangle - \langle f_i^{int}(t) \rangle^2} \tag{5}$$

$$\sigma_i^{ext} = \sqrt{\langle f_i^{ext}(t)^2 \rangle - \langle f_i^{ext}(t) \rangle^2} \tag{6}$$

and their ratio is denoted by:

$$\eta_i = \frac{\sigma_i^{ext}}{\sigma_i^{int}} \tag{7}$$

When $\eta_i > 1$, the external fluctuations dominate the dynamics of the node i. When $\eta_i < 1$, the systems internal dynamics dominate over the external changes.

4 Testbed system

In our paper, we consider the nodes of a multi-tier Web-based system (Figure 2) and the traffic flowing through their nodes. The system is adapted from the architecture in [6]: the first node executes the HTTP server and it is connected to the application server node, that is deployed through the Tomcat servlet container; the back-end node runs the MySQL database server. We collected the activities of all internal components of every node, although in this paper the metrics to describe the internal activities of the nodes refer to the CPU utilization, the disk and network throughput.

The Web-based system activities are driven by the TPC-W workload model that is becoming the *de facto* standard for the performance evaluation of systems providing dynamically generated Web contents (e.g., [6, 9]). The TPC-W benchmark is a transactional Web benchmark. The workload is exercised in a controlled e-commerce environment that emulates the activities of a book seller site. Client requests are generated through a set of *emulated browsers*, where each browser is implemented as a

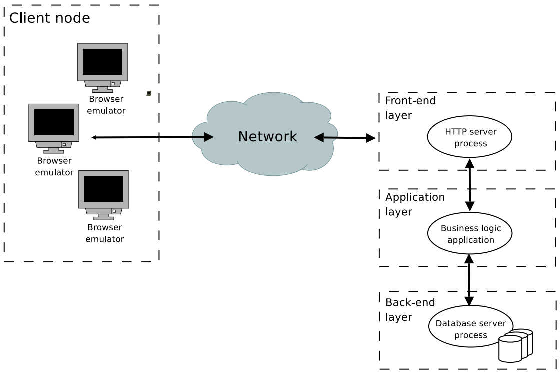


Fig. 2: Architecture of the considered multi-tier Web-based system

Java thread reproducing an entire User Session within the Web site. The Web interactions represent the “walkers” of the model described in Section 3. We consider three scenarios where the number of emulated browsers issuing requests to the Web-based system does not change during the experiment lasting for 30 minutes. We emulate a *light*, an *average* and a *heavy* scenario, where the population refers to 50, 100 and 150 emulated browsers, respectively.

5 Results

In this section we present the results obtained by applying the model described in Section 3 to the Web-based system outlined in Section 4. In Section 5.1, we aim to separate the external from the internal contributions to load fluctuations. In Section 5.2, we evaluate the impact of the perturbation by using the η ratio. In Section 5.3, we demonstrate the robustness of the η ratio evaluation for different conditions in the number and type of Web interactions.

5.1 Separation of external and internal fluctuations

We apply the model described in Section 3 for separating the external and internal components on the CPU utilization, disk and network throughput of the Web, application and database servers. As a representative example, we report the results referring to the network throughput in the case of a heavy user scenario. For each server, we split the time series of the network throughput (figures (a)) into its external contributions (figures (b)) and internal contributions (figures (c)). It is worth to observe that from the Equations 2 and 4 we have that the external factors are characterized by positive values, while the internal factors may be represented by positive and negative values.

Figure 3(a) shows that the network throughput on the Web server is stable with some out of scale values (represented in gray color). By evaluating the variances of the external and internal fluctuations shown in Figures 4(b) and (c), we have that $sigma\ ext = 1292$ and $sigma\ int = 1397$. As they have same magnitude, we can conclude that the internal and external contributions are similar, and there is no component prevailing on the other one.

In Figure 4(a) we report the network throughput of the application server that is characterized by a low average load and many out of scale values. The fluctuations caused by the internal dynamics of the system, shown in Figure 4(c), are characterized by the same behavior as that of the overall measure in Figure 4(a). On the other hand, the external fluctuations in Figure 4(b) are less variable and do not present out of scale values: the variance of the external factor $\sigma^{ext} = 93$ is much lower than that of the internal factor that is equal to $\sigma^{int} = 224$. We can conclude that the external factors do not represent the source of the spikes and that the network throughput fluctuations of the application server depend mainly on the internal system dynamics.

The database node exhibits an interesting alternative to the previous two cases. Its network throughput is characterized by a non stationary and highly variable behavior (Figure 5 5(a)). By separating the internal and external fluctuations, we can appreciate that these components are quite different in terms of intensity and variability ($\sigma^{ext} = 12533$ and $\sigma^{int} = 1501$). The external factors exhibit the same statistical properties as the overall resource measures ($\sigma = 13890$), while the internal factors are quite stable and characterized by low intensity. This behavior induces us to think that the network throughput of the database server is strongly dependent on external client requests.

In the following section, we can confirm some of the previous qualitative and quantitative conclusions by evaluating the η ratio.

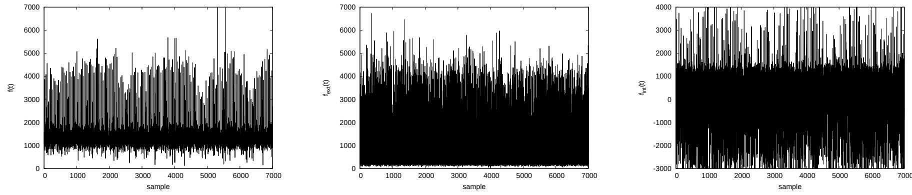
5.2 Classification of the node behavior

The behavior of the internal and external variances of the different nodes influences the correspondent η values. We computed the η ratios of the CPU utilization, disk and network throughput for each node of the Web-based system that are subject to three scenarios: *light*, *average* and *heavy*. Table 1 reporting the different η ratios confirms that the Web, application and database nodes belong to different classes of behavior in terms of [19]. We anticipate an interesting result: independently of the scenario and of the resource measure considered for characterizing a node, the conclusions about classification are the same for all nodes but for the database server. In this last case, the choice of the measure seems to influence the classification.

The typical η ratios for the Web server are ~ 1 for any measure and scenario. This result confirms that the Web server behavior is similarly influenced by external and internal factors. Indeed, this could be expected by considering that all the user requests and the responses have to pass through this node.

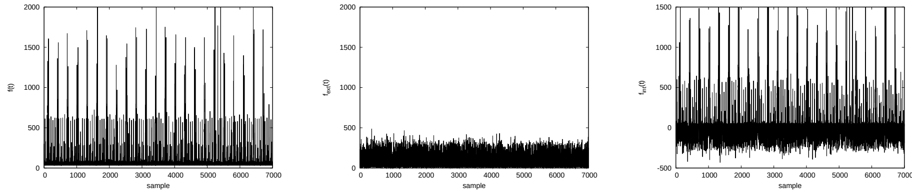
The η values of the application server are always lower than 1. This result denotes a clear prevalence of the internal factors. We motivate this initially unexpected result by considering that in the considered system the application server activities are performed by software based on Java Virtual Machines characterized by strong internal dynamics.

The database server behavior seems anomalous with respect to the previous two nodes. If we consider the CPU utilization and the network throughput as representative measures, then we can conclude that the external factors prevail because the η values are always greater than 1. On the other hand, the same conclusion does not hold true if we consider the disk throughput as a metrics. In such a case, the η values are always clearly below 1, thus indicating a prevalence of the internal factors. We motivate the dependence of the disk throughput on internal load by considering that



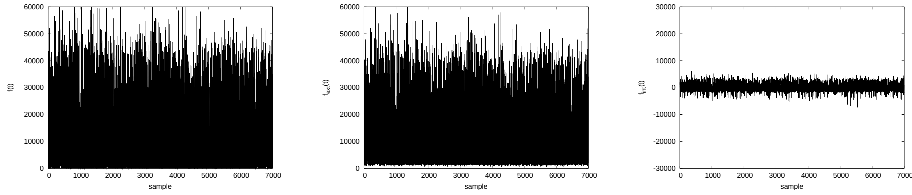
(a) Time series (global view) (b) External factors (c) Internal factors

Fig. 3: Network throughput of the Web server



(a) Time series (global view) (b) External factors (c) Internal factors

Fig. 4: Network throughput of the application server



(a) Time series (global view) (b) External factors (c) Internal factors

Fig. 5: Network throughput of the database server

	Light user scenario		
	HTTP Server	**Application Server**	**DB Server**
CPU utilization	1.27	0.38	1.59
Disk throughput	1.13	0.82	0.69
Network throughput	0.95	0.66	4.26
	Medium user scenario		
	HTTP Server	**Application Server**	**DB Server**
CPU utilization	0.96	0.29	1.30
Disk throughput	1.04	0.73	0.61
Network throughput	0.93	0.58	6.25
	Heavy user scenario		
	HTTP Server	**Application Server**	**DB Server**
CPU utilization	0.92	0.24	1.24
Disk throughput	0.99	0.68	0.67
Network throughput	0.92	0.41	8.35

Table 1: Evaluation of η ratio

in the considered context most requests to the database node are carried out by means of the disk and database cache. Hence, it is not necessary that each external request has to access the physical disk. Although further studies related to different workloads are necessary to confirm this conclusion, our experience in different contexts [2] leads us to conclude that the CPU utilization is the most representative measure of the database node behavior.

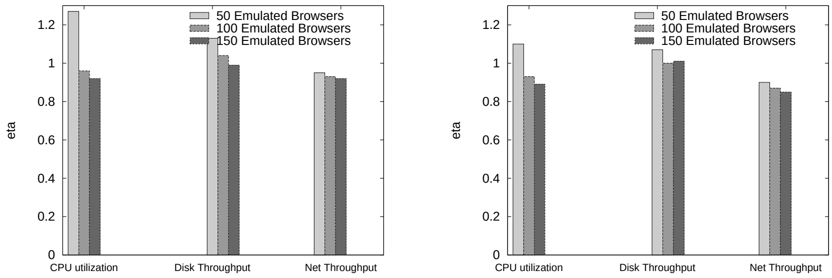
5.3 Robustness of the analysis

In order to evaluate the robustness of the proposed approach and conclusions, we extend the analysis to different workload scenarios. We consider the already applied *light*, *average* and *heavy* scenarios in the context of the browsing and ordering models of the TPC-W benchmark [9]. These workloads are interesting because they have a different impact on the system nodes: the ordering model is characterized by a high percentage of Web interactions with the database server, while the requests of the browsing model affect mainly the Web and the application server.

The histograms of Figures 6, 7 and 8 report the η ratios for CPU utilization, disk and network throughput as a function of the number of emulated browsers (50, 100, 150) for the browsing and ordering workload model. It is quite interesting to observe that all the previous results about node classification and impact of the internal/external factors are confirmed even when we change the workload intensity (number of emulated browsers) and workload model (browsing vs. ordering). Hence, we can conclude the following.

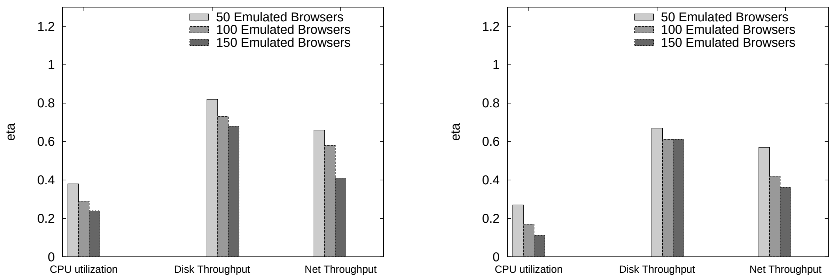
- The nodes, such as the Web server, that are conditioned by the external client behavior and by the interactions with the internal nodes, are always characterized by $\eta \sim 1$ independently of the considered resource measure, as shown in Figure 6(a) and 6(b).
- For the nodes, as the application server shown in Figure 7(a) and 7(b), where the predominating activities are internal, then the η ratios are always lower than 1 for each representative measure, as it was expected.
- The majority of the resource measures referring to the database server (Figure 8(a) and 8(b)) seem to be mainly conditioned by the external factors. Indeed, the CPU utilization and the network throughput are characterized by $\eta > 1$ and $\eta >> 1$, respectively. The exception is represented by the disk throughput that in all instances confirms a value $\eta < 1$.

We conclude by observing that although the evaluations about the internal/external factors are confirmed for any node and scenario, the workload model has a different impact on the value of η. The Web and application servers subject to the ordering workload are characterized by lower values of η than those measured when the browsing workload is considered. The opposite is true when we consider the database server. Our results seem promising and robust, but there is no doubt that the back-end node (characterized by DBMS, operating system, indexes, disk cache, physical storage devices) represents a complex sub-system by itself and requires further investigations.



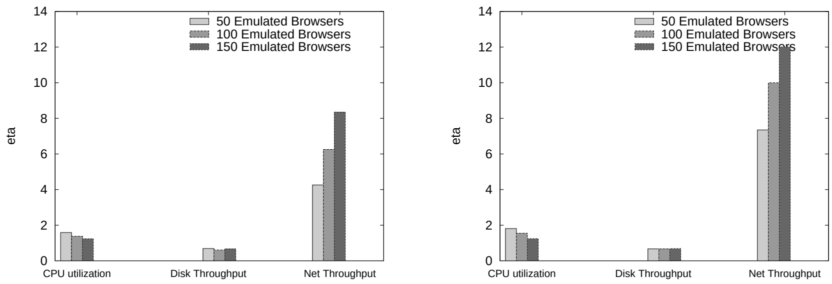
(a) Browsing workload **(b) Ordering workload**

Fig. 6: Web server



(a) Browsing workload **(b) Ordering workload**

Fig. 7: Application server



(a) Browsing workload **(b) Ordering workload**

Fig. 8: Database server

6 Conclusion

Numerous interesting processes in nature and in technological structures are highly cross linked: the basic entities interact nonlinearly to form emergent structures and recurrent patterns of interaction. Scientists and engineers are studying many of these systems (typically large-sized systems hugely interconnected with living entities) by means of the tools coming from the complex systems science. Nevertheless, there are smaller socio-technical structures whose components and architectures that show not predictable complex behaviors although they are explicitly designed by engineers.

There seems to be no studies directly concerned with the properties of these intermediate sized structures which make use of methods from complex systems science, probably because the presence of few nodes suggests that the classical engineering techniques can adequately describe and control their behavior. However, there are technological systems consisting of a number of interconnected servers, clearly affected by the pattern of human usage and at the same time highly influenced by their internal courses of action, the two classes of processes being highly and inextricably interrelated.

In this paper we investigate the behavior of a distributed system providing Web-based services and the effects of the impact of external request arrivals on the internal system resources, a study crucial for taking several runtime decisions on load and resource management. We claim that the introduction of the methods of complex systems science could lead to a better comprehension of the system processes, of the entities really involved in the interaction and of the feedbacks active at all the subsystems levels.

The ability of separating and comparing the variances of the internal and external fluctuations allows new ways to classify heterogeneous nodes, thus revealing the impact of external requests on the system and permitting a first measure of the response of each single node to these solicitations. Another interesting result is the presence of a relationship between the kind (and not the size) of external solicitations and the ratios among the internal and external variances of the system's nodes.

The applied methods include the use different indicators, which allow a more adequate characterization of the processes running on the different nodes and of the recurrent pattern of interaction they are developing. Forthcoming researches could include the extension of this analysis to the study of the influence of different topologies on the system capabilities of maintaining internal stability and a high level of efficiency in delivering external services. The final goal of this process still to be achieved could be the identification of new levels of description able to host new rules, with the aim of developing new kinds of control parameters. The present study aims at providing an initial useful step towards this direction.

References

1. Amaral, L.A.N., Scala, A., Barthelemy, M., , Stanley, H.E.: Classes of small-world networks. In: Proc. Natl. Acad. Sci. USA (2000)
2. Andreolini, M., Casolari, S., Colajanni, M.: Models and framework for supporting runtime decisions in web-based systems. ACM Transactions on the Web **2**(3) (2008)
3. Andreolini, M., Colajanni, M., Nuccio, M.: Scalability of content-aware server switches for cluster-based Web information systems. In: Proc. of WWW. Budapest, HU (2003)
4. Arlitt, M., Krishnamurthy, D., Rolia, J.: Characterizing the scalability of a large web-based shopping system. IEEE Trans. Internet Technology **1**(1), 44–69 (2001)
5. Baryshnikov, Y., Coffman, E., Pierre, G., Rubenstein, D., Squillante, M., Yimwadsana, T.: Predictability of Web server traffic congestion. In: Proc. of the 10th International Workshop on Web Content Caching and Distribution (WCW2005). Sophia Antipolis, FR (2005)
6. Cain, H.W., Rajwar, R., Marden, M., Lipasti, M.H.: An architectural evaluation of Java TPC-W. In: Proc. of the 7th Symposium on High Performance Computer Architecture (HPCA2001). Monterrey, ME (2001)
7. Cancho, R.F., Janssen, C., Sole, R.: Phys. Rev. (2001)
8. Cardellini, V., Colajanni, M., Yu, P.: Request redirection algorithms for distributed Web systems. IEEE Trans. Parallel and Distributed Systems **14**(5), 355–368 (2003)
9. Dodge, R.C., Menascé, D.A., Barbará, D.: Testing e-commerce site scalability with TPC-W . In: Proc. of the 27th Computer Measurement Group Conference (CMG2001). Anaheim, CA (2001)

10. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology (1999)
11. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. In: Proc. Natl. Acad. Sci. USA (2002)
12. Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., , Barabasi, A.L.: The large-scale organization of metabolic networks. Nature **407** (2000)
13. Jordano, P., Bascompte, J., Olesen, J.M.: Invariant properties in coevolutionary networks of plant-animal interactions. Ecology Letters (2003)
14. Jung, J., Krishnamurthy, B., Rabinovich, M.: Flash crowds and denial of service attacks: characterization and implications for CDNs and Web sites. In: Proc. of the 11th International World Wide Web Conference (WWW2002). Honolulu, HW (2002)
15. Kauffman, S.A.: The origins of order. Oxford University Press (1993)
16. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., Upfal, E.: Stochastic models for the web graph. In: In Proc. of the 41st Annual Symposium on Foundations of Computer Science, pp. 57–65. Redondo Beach, CA, USA (2000)
17. Martinez, N.D.: Constant connectance in community food webs. American Naturalist **139** (1992)
18. Menascé, D., Kephart, J.: Autonomic computing. IEEE Internet Computing **11**(1), 18–21 (2007)
19. Argollo de Menezes, M., Barabasi, A.: Separating internal and external dynamics of complex systems. Physical review letters **93**(6) (2004)
20. Potterat, J.J., Phillips-Plummer, L., Muth, S.Q., Rothenberg, R.B., Woodhouse, D.E., Maldonado-Long, T.S., Zimmerman, H.P., Muth, J.B.: Risk network structure in the early epidemic phase of hiv transmission in colorado springs. Sexually Transmitted Infections (2002)
21. Scott, J.: Social network analysis: A handbook. Sage Publications (2000)
22. Serra, R., Villani, M., Semeria, A.: Genetic network models and statistical properties of gene expression data in knock-out experiments. Journal of Theoretical Biology **227** (2004)
23. Vasquez, R.P.S., Vespignani, A.: Phys. Rev. (2002)
24. Villani, M., Serra, R., Graudenzi, A., Kauffman, S.: Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. Journal of Theoretical Biology **249** (2007)
25. White, J.G., Southgate, E., Thompson, J.N., Brenner, S.: The structure of the nervous system of the nematode c. elegans. Philosophical Transactions of the Royal Society of London (1986)
26. Wildstrom, J., Stone, P., Witchel, E., Mooney, R., Dahlin, M.: Towards self-configuring hardware for distributed computer systems. In: Proc. of the Second International Conference on Autonomic Computing (ICAC2005). Seattle, WA (2005)