A Framework for Modelling and Analysis of Software Systems Scalability

Leticia Duboc
Dept. of Computer Science
University College London
London WC1E 6BT
United Kingdom
l.duboc@cs.ucl.ac.uk

Prof. David S. Rosenblum
Dept. of Computer Science
University College London
London WC1E 6BT
United Kingdom
d.rosenblum@cs.ucl.ac.uk

Dr. Tony Wicks
Searchspace Ltd
80-110 New Oxford Street
London WC1A 1HB
United Kingdom
t.wicks@searchspace.com

ABSTRACT

Scalability is a widely-used term in scientific papers, technical magazines and software descriptions. Its use in the most varied contexts contribute to a general confusion about what the term really means. This lack of consensus is a potential source of problems, as assumptions are made in the face of a scalability claim. A clearer and widely-accepted understanding of scalability is required to restore the usefulness of the term. This research investigates commonly found definitions of scalability and attempts to capture its essence in a systematic framework. Its expected contribution is in assisting software developers to reason, characterize, communicate and adjust the scalability of software systems.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems—*design studies, measurement techniques, modeling techniques, performance attributes*; D.2.8 [**Software Engineering**]: Metrics—*product metrics*; D.2.11 [**Software Engineering**]: Software Architectures

General Terms

Design, Economics, Measurement, Performance

Keywords

design, microeconomics, requirements, scalability

1. WHY SCALABILITY?

Developers' attitudes towards scalability have often been to wait for the next generation of machines to appear and speed up their applications for "free". Nevertheless, the gains that can be achieved with current technology advances are reaching their limits. Speed increases on conventional processor cores, for example, are being limited by practical issues related instruction parallelism, pipelining and cooling technologies [13]. The move to chip multiprocessors (CMP) demonstrates this trend. This technology will automatically benefit throughput-oriented workload systems. However, applications measured in terms of the execution latency of individual tasks will require efforts on the part of the programmer. Such systems should take advantage of CMPs while coping with high latency of distributed computational resources. Development will therefore have to enter a new era requiring a proactive attitude towards scalability. This paper describes initial research in the development of a framework for proactive characterization, analysis and understanding of software system scalability.

Building scalable systems is not trivial. The first obstacle is the lack of a common conceptual understanding of scalability. This problem was highlighted in 1990 by Mark D. Hill in "What is Scalability?", which concluded with the following: *"I then question whether scalability is useful and conclude by challenging the technical community to either rigorously define scalability or stop using it to describe systems."* [6]. The article was targeted at multiprocessors systems, but its claims could be made for computing in general. Some researchers took the challenge and, for software in particular, attempted better definitions [5, 11, 2, 3]. Although definitions in the literature could be argued, most of them cannot be classified as entirely incorrect. They correspond to an intuitive notion that scalability is related to a system's ability to accommodate the "scaling" of some dimension.

A *dimension* in the context of our research represents some aspect of the application domain (as defined by Jackson [8]) whose scaling affects system behavior. We therefore argue that, in a subjective field like scalability, a universal definition should be avoided. Instead, one should concentrate on its essence: In order to support scaling of dimensions, it is necessary to (1) clearly identify its causes and effects, and (2) understand their relationship to guarantee desired behavior when aspects related to the system vary.

> **The Problem:** *Stakeholders need a systematic way to understand causes, effects and their relationships in a scaling system, to then judge the extent of the system's scalability.*

2. RELATED WORK

Scalability has been studied in many contexts, such as video imaging, mobile computing, simulation, data mining, distributed systems, software process, among others. One of the areas in which scalability has received more emphasis is parallel systems, where a small number of well-defined metrics were established. Nevertheless, because of fundamental differences between parallel computing and other classes of systems, such metrics cannot be broadly applied.

ICSE'06, May 20–28, 2006, Shanghai, China.
ACM 1-59593-085-X/06/0005.

Most uses of the term "scalability" in scientific papers imply a desired goal or completed achievement, whose precise nature is never defined but rather left to the readers' imagination. Like ourselves, others have questioned the meaning of scalability and propose definitions [6, 14, 2]. However, we believe that definitions found in the literature either represent an intuitive ideal or are restricted to narrowly delimited problems. Previous works have recognized this limitation, stating that scalability has dimensions and should be seen in the context of system requirements [5, 9, 3]. Nevertheless, accommodating all dimensions in clearly defined categories is very challenging, if not impossible. In addition, measuring scalability purely against compliance with system requirements can result in arguable classifications, such as exponential use of machine resources, not to mention that requirements may—and are likely to—change.

Related work can also be found in the characterization and predictions of other software qualities, such as performance and reliability. These works are normally based on trend analysis of selected metrics as the system configuration changes or alternative solutions are compared. Especially among the performance community, the use of well-established models such as layered queuing networks, Petri nets and stochastic process algebra are very popular. Many solutions propose extensions to standards such as UML, ADL, OWL-S and WSOL to incorporate performance attributes [16, 7]. Others instead rely on monitoring capabilities to model and predict performance [1].

Scalability prediction has to consider a dynamic view of the world. More specifically, it requires the understanding of how the system will respond to scaling aspects of this world. Frameworks have been proposed especially for scalability. We, however, see them as tackling only subsets of the problem. Some solutions, for example, limit the study to performance metrics [9, 12]. Others target specific classes of problems or technologies [14, 10]. To the best of our knowledge, there is not a general approach that can accommodate all the different nuances of scalability.

3. REASONING ABOUT SCALABILITY

Stakeholders are often concerned about a system's ability to handle a varied workload, its response to an increase of resources and/or the complexity of models/algorithms when the problem size changes. Whatever the concern, the scaling of a dimension is always present—as an intrinsic aspect of a scalable system. Therefore, our first observation is that *the need to support the scaling of dimensions differentiates systems that are required to be scalable from others.*

Scalability is frequently associated with performance and, at times, the terms are erroneously used interchangeably. Performance, in our opinion, can be an indicator of scalability only when the stakeholder's interests are performance indexes, such as throughput and execution time. Otherwise, performance is simply another requirement to be met by the system, like message reliability, memory usage and other metrics associated with software qualities. For this reason, we believe that *the scalability of a system should be seen in the context of its requirements.*

It is an intuitive notion that scalability is related to the system's ability to accommodate the scaling of some dimension. This scaling is characterized by the system's response to changes in the application domain. We therefore state that *scalability is implicit on the relationship between cause and effect.* However, its classification as good or bad is ultimately *a matter of stakeholder's interest.*

> **Research Claim:** *Scalability is a quality of software systems that is characterized by the relationship between cause and effect, namely the impact that world and machine characteristics have on measured system qualities. This quality can be measured and analyzed in order to provide the stakeholder the support required to judge the scalability of a software system or compare the scalability of alternative designs.*

4. EXPRESSING SCALABILITY

This research attempts to capture the essence of scalability by identifying, for the problem in hand, the causes, effects and relationships that characterize scalability.

Proposed Solution *A systematic framework to support the process of reasoning, characterizing, adjusting and predicting the scalability of software systems.*

The framework is defined in terms of:

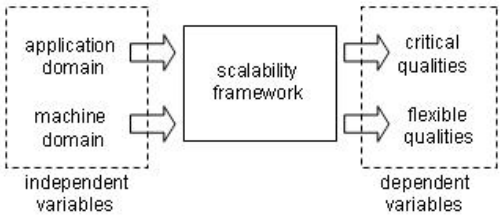

Figure 1: Scalability Framework

Independent and dependent variables relate causes and effects. *Independent variables* represent the subset of the machine and application domains that affects the system behavior. The subset belonging to the machine domain correspond to properties of the system. The ones related to the application domain express aspects of the world. Examples within the machine domain are cache size, number of threads and parallel processes. In the application domain, variables can represent properties like nature and distribution of input data, workload and number of concurrent users.

Dependent variables are aspects of the system behavior (or qualities) that are affected by changes in the application and machine domains. In the framework, they correspond to metrics related to performance, cost, reliability and security, among others. Examples are throughput, storage consumption and memory footprint.

The framework recognizes that not all qualities are equally important. The relative significance of them is a matter of stakeholder's interest. The stakeholder may expect a subset of these qualities to be at their optimal level, while others can simply comply with minimum requirements. Therefore, the framework recognizes a division between critical and flexible qualities. The former represent the aspects of the systems that should be optimized, while the latter can be implemented more flexibly.

We plan to use multi-criteria optimization techniques to analyze the trade-off between alternative machine configurations across scaling dimensions. The appeal of multi-criteria

optimization is that it allows conflicting quantitative objectives to be simultaneously optimized. The stakeholder's interest is formalized as a utility function, which expresses the relative importance of dependent variables. A number of utility functions can be used to compare alternative machine configurations. The chosen one will reflect the specific requirements of the application being analyzed.

5. THE SEARCH FOR SCALABILITY

To the best of our knowledge no generic framework for analyzing the scalability of software systems has been developed. Proposed frameworks are invariably aimed at narrowly defined problems and cannot accommodate all the nuances of scalability [9, 12, 14].

The relevance of the research increases as machines are reaching physical limitations in hardware. Development will have to adopt a more proactive attitude towards scalability, requiring a clear and widely-accepted understanding of the subject. Our research intends to set the grounds for this new era. Ultimately this research aims to provide the following:

> **Research Contribution:** *(1) a precise definition that capture the essence of scalability across a wide-range of software systems domains; and (2) a sound method and prototype implementation framework to reason about, communicate, characterize and predict the scalability of software systems.*

It is not our intent to provide an out-of-the-box solution, with predefined formulas and rules of thumb. Although a common ground may be possible, the specifics of the framework instantiation will vary from system to system according to their requirements. One could argue that software systems fit into categories, and that our work should attempt to provide a catalog of framework instantiations. We are, however, reluctant to adopt the idea, as it is often easy to imagine exceptions to such rules. It also would be very challenging in the life-span of a PhD to investigate, formalize and evaluate solutions to the whole range of software domains. Nevertheless, others could use the proposed framework as a starting point to develop specific solutions. This common ground would enable a shared language to express scalability concerns to people outside the specific problem domain.

A flexible solution imposes risks. Misleading results could be derived if an inadequate framework instantiation was chosen. This is, however, a problem faced by many other model-based solutions, such as queuing networks. Adoption could be jeopardized, especially among first-time adopters, because the lack of hard rules may be intimidating. Nevertheless, the main challenge to be faced by this research will be in determining the degree of utility and accuracy of its analysis results. Approaches will have to be investigated to overcome this problem.

In terms of assumptions, this work presumes that the stakeholder has enough knowledge about the problem domain to choose a suitable set of variables to characterize the application in hand. The stakeholder is also assumed to be able to articulate requirements in terms of utility functions.

6. THE WAY TO SCALABILITY

As the research enters its eighth month, we can summarize current results and the plan the research method for the remaining time of the PhD:

Research Method.

- *Critical literature review*
- *Identification and scope of the problem*
- *Formalization of a model for reasoning about, communicating, characterizing and predicting scalability*
- *Model refinement and evaluation through case studies*

Research Progress. A critical literature review took place in the first months of research. The objective was an understanding of the different contexts in which scalability has been studied. In a second stage, proposed solutions to analyze and predict software system scalability were reviewed. An overview of the state-of-the-art can be found in section 2. Readings also targeted the area of performance prediction. The two subjects are considered closely related in literature. Although we believe there is a clear division between them, many of the issues surrounding both topics are the same, and proposed solutions for performance prediction can provide valuable insights for a scalability study. The literature review highlighted the many misunderstandings and concerns on the subject. It also demonstrated that the field lacked a clear, common, widely-accepted definition of scalability.

The following stage was devoted to the initial experiments with a real-world system. A scalability problem was identified, and interactions with stakeholders started to shape the analysis objectives. The comparison of the case study with examples found in the literature highlighted what we now believe to be the essence of scalability: its causes, effects and their relationships. The development of a conceptual model naturally followed from this observation. A framework was formalized and applied to the case study. Metrics were collected for distinct machines configurations and different values in the scaling dimension—in this case, the number of distinct business entities.

The research is currently investigating ways to analyze the data collected in the case study. Attention is being paid to the multi-criteria optimization methods. The trade-off between alternative machine configurations is being represented by a family of Pareto frontiers. Utility and social welfare utilitarian functions are under investigation to formalize the stakeholders' interest and transform a multi-criteria optimization problem into a scalar one. Such models are commonly used in microeconomics to support the decision process [15]. However, to the best of our knowledge, no study has looked at understanding the scaling of Pareto frontiers against varying dimensions.

Research Plan. The conceptual model developed is still not complete. Once the problem has been expressed in terms of independent and dependent variables, data has to be gathered so that the relationship between scalability causes and effects can be analyzed. Data may come from two sources: measurement or inference. The first can be used to analyze the scalability of existing software systems. The second should enable the analysis of applications yet to be developed. A possible approach is to derive data from early life cycle development models, such as MDA and UML diagrams [12].

In both cases, data also have to be extrapolated to account for future demands. One should take care with extrapolation

because of uncertainty, which reflects unexpected changes in the world and in the system's behavior. We plan to use the computational complexity of the different observable characteristics of the system to estimate possible future values of the scaling dimensions. The expected welfare of the system can then be calculated by taking into consideration estimated bahaviour of the system and the probability that the scaling dimensions will reach certain levels.

Another area to be investigated further is the selection of variables for the framework. The correct identification of a system's attributes is crucial for a useful and reliable scalability analysis. Attributes may be correlated, measuring the same or similar system properties. Statistical data analysis techniques such as principal component analysis can assist the selection of a representative set of variables [4].

Plan for Evaluation. Evaluation of the research claim will represent a major challenge for this work. Ultimately, this work aims to establish the usefulness of the term "scalability". Usefulness is, however, a concept hard to prove. Nevertheless, we can evaluate the accuracy, expressiveness and adequacy of the proposed model by applying it to software systems in distinct domains. More specifically we plan to:

- Verify the accuracy of the prediction by applying the conceptual model to source-controlled versions of software systems.
- Analyze the system's ability to deal with uncertainty by comparing the results of consecutive releases of systems and analysing whether changes in the application domain would have invalidated the analysis results.
- Apply the framework to a growing number of scaling dimensions and compare the estimated scalability with data empirically collected.
- Investigate the challenges and consequences of defining and applying distinct utility functions to scalability analysis.
- Study the impact of instantiating the framework with different sets of variables.
- Collect stakeholders' feedback to evaluate adequacy and expressiveness of the model.

7. CONCLUSION

A more proactive approach to scalability is required as current technology advances are reaching their physical limits. To set the ground to this new era, a clear and widely-accepted understanding of scalability is needed. In our opinion, current solutions represent either an intuitive ideal or are restricted to narrowly-defined problems. This research claims that scalability is a software quality that can be measured and analyzed to support decision making in software development. We attempt to capture its essence in a systematic framework to reason about, measure, analyze and predict the scalability of software systems.

8. ACKNOWLEDGEMENTS

Leticia Duboc is funded under a studentship from UCL. David Rosenblum holds a Wolfson Research Merit Award from the Royal Society. The authors thank Wolfgang Emmerich, Anthony Finkelstein, Damon Wischik and Rami Bahsoon for their valuable contributions to this work.

9. REFERENCES

[1] A. Avritzer, J. Kondek, D. Liu, and E. J. Weyuker. Software performance testing based on workload characterization. In *Proc. Third Int'l Workshop on Software and Performance*, pages 17–24. ACM Press, 2002.

[2] A. B. Bondi. Characteristics of scalability and their impact on performance. In *Proc. Second Int'l Workshop on Software and Performance*, pages 195–203. ACM Press, 2000.

[3] G. Brataas and P. Hughes. Exploring architectural scalability. In *Proc. Fourth Int'l Workshop on Software and Performance*, pages 125–129. ACM Press, 2004.

[4] L. Eeckhout, H. Vandierendonck, and K. De Bosschere. Quantifying the impact of input data sets on program behavior and its applications. *J. Instruction-Level Parallelism*, 5, 2003.

[5] D. B. Gustavson. The many dimensions of scalability. In *COMPCON*, pages 60–63, 1994.

[6] M. D. Hill. What is scalability? *ACM SIGARCH Computer Architecture News*, 18(4):18–21, 1990.

[7] R. P. Hopkins, M. J. Smith, and P. J. B. King. Two approaches to integrate UML and performance models. In *Proc. Third Int'l Workshop on Software and Performance*, pages 91–92. ACM Press, 2002.

[8] M. Jackson. *Software Requirements & Specifications: A lexicon of practice, principles and prejudices.* Addison-Wesley, 1995.

[9] P. Jogalekar and M. Woodside. Evaluating the scalability of distributed systems. *IEEE Trans. Parallel and Distributed Systems*, 11(6):589–603, 2000.

[10] Y. Liu and I. Gorton. Accuracy of performance prediction for EJB applications: A statistical analysis. In *Proc. 2004 Workshop on Software Engineering and Middleware*, pages 185–198, 2004.

[11] E. A. Luke. Defining and measuring scalability. In *Proc. Scalable Parallel Libraries Conference*, pages 183–186. IEEE Press, October 1993.

[12] S. Masticola, A. B. Bondi, and M. Hettish. Model-based scalability estimation in inception-phase software architecture. In *Proc. ACM/IEEE 8th Int'l Conference on Model Driven Engineering Languages and Systems*, pages 355–366, 2005.

[13] K. Olukotun and L. Hammond. The future of microprocessors. *ACM Queue*, 3(7):26–29, 2005.

[14] M. van Steen, S. van der Zijden, and H. J. Sips. Software engineering for scalable distributed applications. In *Proc. 22nd Int'l Computer Software and Applications Conference*, pages 285–293, 1998.

[15] H. R. Varian. *Intermediate Microeconomics: A Modern Approach.* W. W. Norton, 6th edition, 2003.

[16] M. Woodside, D. C. Petriu, D. B. Petriu, H. Shen, T. Israr, and J. Merseguer. Performance by unified model analysis (PUMA). In *Proc. Fifth Int'l Workshop on Software and Performance*, pages 1–12. ACM Press, 2005.