Computer Systems Performance Engineering in Trondheim: Origins and Development, 1970–1995

Peter Hughes

Computer Systems Performance Engineering in Trondheim: Origins and Development, 1970–1995

Peter H. Hughes

Department of Computer and Information Science
NTNU, Trondheim, Norway
peterh@idi.ntnu.no

Abstract. Developments covered by this review include early experimental work on Univac mainframes, a contribution to early debate regarding the application of queue-network analysis, the development of bespoke benchmarking techniques, one of the first interactive load generators, and modeling tools for system sizing and for simulation. We show how addressing problems that arose from practical needs has benefited both university teaching, and industrial practice. The period covers the transition of performance evaluation as a set of ad hoc modeling and measurement techniques to performance engineering, which aspires to be an integral part of systems design and development. Cross-fertilization and collaboration with developments internationally form an important part of the activities reported.

Keywords: Benchmarking, model-support environments, performance engineering, queue-network analysis, simulation, system sizing

1 Introduction

The computer center at SINTEF in Trondheim, later known as RUNIT, was a pioneer in the practice and teaching of computer systems performance evaluation. Today, NTNU is one of a very few teaching establishments in the world with computer systems performance engineering as an established part of the undergraduate curriculum [1]. In this paper, we trace some of the activities and developments, which contributed to this long journey.

In scope, the paper deals with those applications of performance evaluation related to computer systems capacity management and information systems development. Other application domains, such as computational software optimization, hardware design, and the design of telecommunications systems are not considered.

Over the period of twenty-five years, some eight researchers were involved, augmented by a number of master-level students. At any one time, the core research group numbered between two and five. The treatment is broadly chronological, with some parallel threads for clarity of exposition. In the final section, we consider the larger context in which these activities took place and make some general observations.

2 The Mainframe Era: Measurement, Modeling and Experimentation

The work began with a practical need, mainly to understand capacity aspects of the expensive UNIVAC 1100 series mainframes introduced to Scandinavia in the late 1960s and early 1970s. The performance of mainframe computers and their operating systems was at the time something of a mystery. They were extremely complex engines with opaque scheduling policies and a range of adjustable parameters, the tuning of which was a dark art. Even senior technical personnel at Univac had difficulty with the more esoteric areas of the EXEC8 operating system.

A performance evaluation group came in existence at Trondheim in 1971. The purpose was to guide decisions on tuning, upgrade and future investment. The approach was apparently simple: select a representative set of programs and data and use this "benchmark" to measure the capacity of alternative system configurations. The predictive accuracy of this approach depended on the benchmark being sufficiently representative of the user workload and on the measurements obtained being sufficiently comparable. Neither requirement was easy to fulfill. This early experimental work turned out to have a wider impact in the following three directions.

(i) The Art of Benchmarking

Firstly, we developed techniques whereby benchmark tests became the basis of reproducible experiments, in which both workload and system state were carefully controlled. This was a complex challenge, involving an understanding of a large, disparate set of user programs and data, analysis of system log-files, and special instrumentation of the operating system [2]. We developed the idea of a benchmark beyond its origin as an arbitrary reference point, to become a model of an actual user workload. The reward for this rather laborious investment was that we were able to offer a customized benchmarking service to other computer centers with procurement projects. Internationally, they invited us to contribute to a state-of-the-art report for the industry on computer measurement [3].

(ii) Models of Multiprogramming

The second contribution was less direct, but at least as significant. Certain results due to Geir Moe [4] gained international recognition as a rare experimental validation of the new theory of queue-network analysis.

A purely measurement-based approach to performance prediction is limited by the range of configurations, which are available for testing and by the cost of such an exercise. It was natural to consider a modeling approach to reduce the number of measurements required. Modeling of computer systems was however in its infancy.

Moe had developed a simulation model for multiprogramming based on simple probabilistic assumptions. This model displayed some intriguing asymptotic behavior. About this time, at a conference of the British Computer Society, Conway Berners-Lee, from the UK computer firm ICL, provided a queuing network interpretation of multiprogramming systems [5]. Armed with Berners-Lee's analysis, Moe was able to explain the behavior of his simulation model and show that it was consistent with a queuing network model. Moreover, results from the model were consistent with results obtained in the benchmark experiments [6].

It was both difficult and expensive to do controlled experiments with large mainframes. It turned out that our painstaking benchmark tests constituted one of the very few measurement experiments worldwide that provided evidence for the queue-theoretic interpretation. At the time there was much controversy regarding the validity of queue-network models. Real systems do not have exponential service times and other mathematically convenient Markov properties, and it was not clear until years later, how far queue-theoretic predictions could be relied upon.

(iii) Synthetic Programs
It quickly became apparent that modeling and measurement together could provide powerful insights. Performance measurement, in combination with queuing theory, simulation and statistics, was helping to demystify mainframe computers and to untangle the complexity of computer system behavior. We investigated some key issues surrounding the role of so-called synthetic programs in benchmarks. Was it legitimate to model computer workloads with artificial programs and data? Which properties should we preserve? [7, 8] How might a set of programs be statistically representative of a workload? [9]

3 The Birth of an Academic Subject

At RUNIT in 1972 a teaching unit was formed which became the new *Institutt for Databehandling* (IDB) at *Norges Tekniske Høgskole* (NTH), led by professor Arne Sølvberg. After various reorganizations, these two entities were subsumed in larger ones, leading eventually to the present structure: the *Institutt for Datateknikk og Informasjonsvitenskap* (IDI) at NTNU.

Following an initiative by Sølvberg, we packaged and distilled some of our practical experience for teaching purposes [10]. The new subject came to be entitled *Systemering III*, a sequel to the systems design courses *Systemering I* and *II*. This association anticipated by more than a decade the link between performance evaluation and design now implicit in terms such as software performance engineering [11]. The course content gradually extended to both practice and theory by contact with industry and the wider academic community.

Globally, research in the mathematical analysis of queuing networks proceeded in parallel with the practical side of performance engineering, as is still the case today. Although mathematical textbooks appeared, there was little to support the more holistic engineering approach we wished to develop. In 1978, the situation had transformed by the appearance of Domenico Ferrari's comprehensive and scholarly work, *Computer Systems Performance Evaluation* [12]. This text became the foundation of the subject for years to come.

In the same year, the author received an invitation to present our work on benchmarking techniques as part of a summer school in Urbino, Italy, led by Ferrari. The gathering attracted many leading researchers in the field. The widening of horizons provided by this experience was of long-term benefit to the course in Trondheim.

In the late 1970s, the theory of operational analysis developed [13]. Based on directly measurable quantities, this provided a simpler approach for analyzing multi-

programming systems. Our earlier simulation results also supported an operational interpretation. Indeed, some of the invariance rules derived in [6] are equivalent to operational laws. Alternative theories continued in passionate debate for some years, until the protagonists of the several mathematical camps learned to accommodate one another [14].

In 1984, a second landmark textbook appeared in which the new queue-network solution technique known as "Mean Value Analysis," expressed in operational form [15]. This relieved our syllabus of an overload of queuing theory.

4 Interactive Computing: Mini-computers, Hardware Monitors and the RUNIT Interactor

By the late 1970s, the importance of interactive computing was increasing and the minicomputer revolution was undermining the position of mainframe computing centers. In a changing economic climate, the performance group at RUNIT became a cost center, which had to finance itself from industrial consultancy and research grants. The group made a successful research proposal in collaboration with the Norwegian computer manufacturer Norsk Data A/S to develop one of the world's earliest interactive load-generators: the RUNIT Interactor [16]. This device could monitor and learn the keystrokes of a human operator, scale up and emulate a controlled load, and measure the resultant response times.

They used the Interactor in numerous benchmarking studies in the state sector under Norway's *Statens rationaliseringsdirektorat*. It subsequently became one of Norsk Data's principal tools for remote stress testing over telephone links (e.g., from Oslo to Stockholm). The Interactor attracted commercial interest from Denmark and the UK. In today's climate, it would undoubtedly have been a prime candidate for commercial exploitation.

The Interactor stimulated the development of other experimental techniques. They frequently used it together with a performance monitor, which could be software- or hardware-based depending on the target system. This required intimate acquaintance with either the workings of the target operating system, or the circuitry of the hardware "back-plane" to which measurement probes were to be attached.

A second important technique was the development of executable workload models. These consisted of two parts: scripts, which emulated user-to-computer interactions, and target programs, where scripts invoked execution. The target programs could consist of real programs and data. However, this approach limited our range of investigation. We therefore developed a prototyping system known as PILOT to study the performance of CODASYL database systems before they became operational. PILOT generated synthetic workloads with appropriately randomized keys and artificial data for the target database schema [17].

5 Simulation Modeling and Model-driven Design

At IDI, they introduced a course in discrete-event simulation based on the Simula language. The introduction of an elegant Simula class known as DEMOS (Discrete Event Modeling on Simula) transformed this course [18]. Developed at the University of Bradford in the UK, Graham Birtwistle (who had worked on the Simula project in Oslo) created DEMOS. It was the outcome of practical experience with applying simulation to industrial problems. The original SIMULATION context of the Simula compiler was low-level and complicated to use. DEMOS exploited the powerful extendibility features of Simula by providing a small set of high-level synchronization constructs. These constructs lend themselves readily to a graphical representation known as "activity diagrams." In Trondheim, we enhanced the DEMOS activity diagrams to develop an accessible undergraduate teaching method. This enabled us to focus more on the core techniques of discrete-event simulation, and less on the details of the simulation language.

The Process Interaction Tool (PIT) developed in the UK later adopted the enhanced activity diagrams [19]. PIT was a unique tool, which supported the construction of simulation models via a graphical interface [20]. Industry used it for models ranging from hardware design, to telecommunications, to real-time financial settlement. The appeal of the DEMOS constructs is demonstrated in their adoption by simulation tools in other object-oriented languages such as C++ and Java. A modern example is DESMO-J from the University of Hamburg.

6 Sizing, Configuration and Deployment

Computer systems have a coarse-grained modularity arising from separately developed software and hardware units configured together for particular applications. The choices made regarding dimensioning and deployment of such modules directly affect performance. The industrial experience with this problem in the 1970s in the UK and Norway caused us to develop a semi-formal quantitative framework known as "Structure and performance Specification" (Sp). The first description of Sp appeared in 1983 [21]. Collaborative projects established with Norsk Data and subsequently with ICL in the UK [22] led to a succession of prototype Sp tools.

Sp has exhibited a long staying power. It proved capable of modeling successive generations of computer and software architecture such as enterprise systems and mobile platforms; these conceptions did not exist at its inception. Recent work [23] suggests that Sp offers a more powerful alternative to the "deployment diagram" currently offered by the Unified Modeling Language (UML). Moreover, we have found that Sp provides a vital part of the conceptual foundation needed for the teaching of Performance Engineering [1].

7 Graphical Workstations and the Integrated Modeling Support Environment

The 1980s saw the emergence of powerful graphical workstations with potential to construct and solve detailed models. This gave an enormous stimulus to the development of new performance tools based on various modeling paradigms. In parallel with this development, the new generation of object-oriented databases was emerging.

At IDI, a separate line of research was investigating model-driven design in the context of information systems. Most work in this area was conceptual and functional rather than quantitative. It proved feasible in combination with Sp to address some of the quantitative issues [24]. Several doctoral dissertations investigated related topics [25–27].

In 1989–1992, the various strands of research and development described above came together in a European Research project under the Esprit 2 program. This was IMSE, an Integrated Modeling Support Environment, which involved nine industrial and academic partners from five European countries [28]. Trondheim was an active partner [29]. The conceptual basis of IMSE was developed via a UK predecessor project known as SIMMER [22]. Sp and PIT tools integrated with queue-network, petri-net, simulation, and workload analysis tools in an object-based system having a common meta-model and a common graphic support system. They built an experimenter tool that exploited the common framework provided by the meta-model. IMSE was a pioneering environment that had a strong influence on the development of performance engineering across Europe, still traceable today.

8 Conclusions

This condensed case history provides a basis for the following observations.

A. It contains concrete demonstrations of some familiar principles such as
   i. practical needs leading theoretical advance,
   ii. application of the classical scientific method,
   iii. synergy of research and teaching,
   iv. vital role of the international community.

B. It demonstrates the importance of taking a long-term view. No one could have predicted in 1970 how the field of performance engineering would develop or the challenges it faces today. Many thought that the subject would not survive in the face of Moore's Law. Yet Moore's Law is now becoming obsolete and the field is well established.

C. Mobility of individuals between industry and academia was extremely valuable, perhaps even essential, to the development of this field in Norway. Cross-fertilization with the United Kingdom, with its larger industrial base, had mutual benefits. The SINTEF model of industry-related research seems to this observer to have been particularly advantageous.

D. With a great deal of effort and some delay, the evolution of the teaching material reflected step-changes in the state of the art. Course content improved markedly

with the arrival of landmark texts in 1978, 1979, and 1984. Our thoughts could make bigger strides and we created more room for applications. The need to keep up with constantly changing technology balanced these gains. The syllabi we developed were not contained in any one text. Although methodology and mathematics changed infrequently, the applications that gave meaning for successive generations of students changed much faster.

E. The context in which the early work took place was quite different from today. At the beginning of the period, computing itself received scarce recognition in Norway as an academic subject. Although we had benefited from RUNIT's good economy, obtaining funds from research committees was extremely difficult and uncertain. We were indebted to the good offices of a few far-sighted individuals and to the personal enthusiasm and commitment of many team members and students.
F. We could have done better at exploiting of our research and development results. This would have required more mentoring early on and more financial investment at crucial times. We can now better appreciate such needs. Nonetheless, we had the privilege to work in a very supportive environment.

This review over a quarter-century ends in 1995. We conclude that performance engineering has not only been a valuable practical subject but also a strong stimulus to research. The dynamic interplay of measurement and modeling continues to throw up challenging questions about the systems we create as computer systems engineers.

Feedback from alumni and industrial contacts indicates that inclusion of performance engineering in the education of computing engineers in Trondheim is greatly valued. Nonetheless, projects and systems everywhere often fall short in performance, sometimes with disastrous consequences. As system complexity rises, high-level software engineering becomes increasingly remote from its physical effects. Clearly, we need to do more education and more research.

References

1. Hughes, P.H.: Lecture Notes in Performance Engineering IDI, NTNU (1995, 2010)
2. Hughes, P.H.: Developing a reliable benchmark for performance evaluation. Proceedings, NordDATA 72 Conference, Helsinki, vol. 2, pp. 1259–1284. Finska Dataförbundet rf. (1972)
3. Hughes, P.H.: Towards Precise Benchmarks. Infotech International Limited State of the Art Report 18 Computer Systems Measurement ISBN 8553-9170-7 (1974)
4. Moe, G.: Computer Performance Analysis using Simple Queuing Models. Lic.techn. thesis, NTH (1973)
5. Berners-Lee, C.M.: Three analytical models of batch processing systems. Proceedings British Computer Society Conference 1972, pp. 43–52 (1972)
6. Hughes, P.H., Moe, G.: A structural approach to computer performance analysis. AFIPS Joint Computer Conferences, Proceedings of the National Computer Conference, New York 1973, pp. 109–120 (1973)
7. Barber, E.O., Asphjell, A., Dispen, A.: Benchmark construction. ACM SIGMETRICS Performance Evaluation Review vol. 4, (4). ISSN:0163-5999 (1975)

8. Hughes, P.H.: Benchmarks, Workloads and System Dynamics. Invited paper, Infotech International Limited, Conference on Performance Modelling and Prediction, ISBN 8553-9410-2 London (1977)
9. Barber, E.O.: A question of balance. D.Ing dissertation, Division of Computer Science, University of Trondheim (1981)
10. Asphjell, A.: Ytelsesvurdering av datamaskinsystemer. Teaching Notes, RUNIT/ NTH (1976)
11. Smith, C.U.: Performance Engineering of Software Systems. SEI Series in Software Engineering, Addison-Wesley, ISBN 0-201-53769-9 (1990)
12. Ferrari, D.: Computer Systems Performance Evaluation. Prentice-Hall, ISBN 0-13-165126-9 (1978)
13. Denning, P.J., Buzen, J.P.: The operational analysis of queuing network models. ACM Computing Surveys, 10(3). (1978)
14. Denning, P.J.: A tale of two islands (a fable about operational analysis). ACM SIGMETRICS Performance Evaluation Review 9, (4). (1980)
15. Lazowska, E.D., Zahorjan J., Scott Graham G., Sevcik K.C.: Quantitative System Performance. Prentice Hall, ISBN 0-13-746975-6 (1984)
16. Hughes, P.H., Dispen, A., et al.: Applications of the RUNIT Interactor. Proceedings, NordDATA 81 Conference, Copenhagen, vol. 3, pp. 286–291. Dansk Databehandlingsforening (1981)
17. Hughes, P.H.: PILOT – A Synthetic Prototype Generator for Database Applications. In: Potier, D. (ed.) Proc. International Conference on Modelling Techniques and Tools for Performance Analysis, Paris, France. INRIA, North Holland (1984)
18. Birtwistle, G.M.: Discrete Event Modelling on Simula. MacMillan, ISBN 0-333-23881-8 (1979)
19. Pooley, R.J., Hughes, P.H.: Towards a standard for hierarchical process oriented discrete event simulation diagrams part II: the suggested approach to flat models. Transactions, Society for Computer Simulation, vol. 8 (1) pp. 21–31 (1991)
20. Barber, E.O., Hughes, P.H.: Evolution of the Process Interaction Tool. In: Proceedings, Association of SIMULA Users Conference, Pilsner, Czechoslovakia (1990)
21. Hughes, P.H.: A structural analysis of information processing systems (with application to the sizing problem). Inst. for Databehandling NTH. (1983)
22. Hughes, P.H.: Design of a Performance Modelling Environment. In Proceedings, Association of SIMULA Users conference, Stockholm (1986)
23. Hughes, P.H., Løvstad, J.S.: A generic model for quantifiable software deployment. In Proceedings International Conference on Software Engineering Advances ICSEA 2007, IEEE. ISBN 0-7695-2937-2 doi>10.1109/ICSEA.2007.4 (2007)
24. Opdahl, A.L.: Sensitivity Analysis of Combined Software and Hardware Performance Models: Open Queuing Networks. Performance Evaluation Journal, vol. 22(1), Elsevier (1995)
25. Opdahl, A.L.: Performance Engineering during Information System Development. Dr.Ing dissertation, IDT, NTH Trondheim (1992)
26. Vetland, V.: Measurement-based Composite Computational Work Modelling of Software. Dr.Ing dissertation. IDT, NTH Trondheim (1993)
27. Brataas, G.: Performance engineering method for workflow systems: an integrated view of human and computerised work processes. Dr.Ing. dissertation, IDI, NTNU (1996)
28. Pooley, R.J.: The Integrated Modelling Support Environment. In: Balbo and Serazzi (eds.) Proc. 5th Int. Conf. On Modelling Techniques and Tools for Computer Performance Evaluation, (Turin), pp. 1–16, North Holland, ISBN 0 444 88989 2 (1991)
29. Brataas, G., Opdahl, A.L., Vetland, V., Sølvberg, A.: Final Evaluation of the IMSE. Technical Report, IMSE project deliverable D6.6-2 SINTEF/University of Trondheim (1991)