Dagstuhl Seminar 00271

Stochastic and Dynamic Real-Time Systems

02. 07. - 07. 07. 2000

G. Hommel (Berlin), L.R. Welch (Athens, Ohio)

Introduction

Günter Hommel and Lonnie R. Welch

In many existing real-time computing models, the execution time of a "job" is used to characterize its workload. Typically, it is assumed that an *integer worst-case execution time* (WCET) is known *a priori.* This is not without justification, since static engineering approaches based on non-stochastic models have utility in many application domains [Sha91]. Furthermore, the pre-deployment guarantee afforded by such approaches is highly desirable. However, there are numerous applications which must operate in dynamic environments, thereby precluding *accurate* characterization of the applications' properties by static models which are non-stochastic. Some real-time systems operate in environments which can be characterized *a priori* by a statistical distribution. Other control systems operate in environments which can not be modeled accurately with a time-invariant distribution; their time-variant stochastic characterizations must be repeatedly derived *a posteriori.*

A growing number of researchers in the field of real-time systems are aware of those problems. On the other side there are researchers in the field of stochastic modeling who are interested in modeling and analyzing non-Markovian stochastic systems including their partially deterministic behavior. The goal of this Dagstuhl-Seminar is now to bring together researchers of both fields in order to consider engineering approaches for real-time systems which cannot be characterized accurately by non-stochastic *a priori* models.

In typical real-time computing models (e.g., see [Liu73, Ram89, Xu90, Sha91, Bak91]), execution time is assumed to be an a priori integer "worst-case" execution time (WCET). While [Sha91] establishes the utility of a priori WCET-based approaches by listing some domains of successful application, others [Leh96, Jah95, Hab90, Kuo97, Sun96, Ram89, Tia95, Str97, Ste97, Liu91, Abe98, Atl98, Bra98] cite the drawbacks, and in some cases the inapplicability, of the approaches in certain domains. [Ram89, Tia95, Leh96, Hab90, Abe98] indicate that characterizing workloads of real-time systems using a priori worst-case execution times can lead to poor resource utilization, particularly when the difference between WCET and normal execution time is large. It is stated in [Ste97, Abe98] that accurately measuring WCET is often difficult and sometimes impossible. In response to such difficulties, techniques for detection and handling of deadline violations have been developed [Jah95, Str97, Ste97].

Recently, paradigms which generalize the execution time model have emerged. Execution time is modeled as a set of discrete values in [Kuo97], as an interval in [Sun96], and as a time-invariant probability distribution in [Leh96, Str97, Tia95, Atl98]. These approaches assume that the execution characteristics (set, interval or distribution) are known a priori.

Others have taken a hybrid approach; for example, in [Hab90] a priori worst case execution times are used to perform scheduling, and a hardware monitor is used to measure a posteriori task execution times for improving hardware utilization via dynamic adaptation. [Liu91, Str97] view jobs as consisting of mandatory and optional portions, with one of these having characteristics that can not be known a priori. In [Liu91] the mandatory portion has an a priori known execution time, while the optional portion has an unknown execution time. In [Str97], the optional portion is used for handling timing violations of the mandatory portion and thus has an a priori known execution time. In [Bra98, Wel98] resource requirements are observed a posteriori, allowing applications which have not been characterized a priori to be accommodated. Also, for those applications with a priori characterizations, the observations are used to refine the a priori estimates. These characterizations are then used to drive dynamic resource allocation algorithms.

Engineering approaches for stochastic and dynamic real-time systems have the potential to extend the applicability of real-time computing research into new domains of use. Thus, we propose to focus on advancing the modeling and analysis techniques for such systems.

[Abe98] L. Abeni and G. Buttazzo, "Integrating multimedia applications in hard real-time systems," in Proceedings of the 19th IEEE Real-Time Systems Symposium, 3-13, IEEE Computer Society Press, 1998.

[Atl98] A. Atlas and A. Bestavros, "Statistical rate monotonic scheduling," in Proceedings of the 19th IEEE Real-Time Systems Symposium, 123-132, IEEE Computer Society Press, 1998.

[Bak91] T.P. Baker, "Stack-based scheduling of realtime processes," Journal of Real-time Systems, 3(1), March 1991, 67-99.

[Bra98] S. Brandt, G. Nutt, T. Berk and J. Mankovich, "A dynamic quality of service middleware agent for mediating application resource usage," in Proceedings of the 19th IEEE Real-Time Sys. Symposium, 307-317, IEEE Computer Society Press, 1998.

[Hab90] D. Haban and K.G. Shin, "Applications of real-time monitoring for scheduling tasks with random execution times ," IEEE Transactions on Software Engineering, 16(12), December 1990, 1374-1389.

[Jah95] F. Jahanian, "Run-time monitoring of real-time systems," in Advances in Real-time Systems, Prentice-Hall, 1995, 435-460, edited by S.H. Son.

[Kuo97]T.E. Kuo and A. K. Mok, "Incremental reconfiguration and load adjustment in adaptive real-time systems," IEEE Transactions on Computers, 46(12), December 1997, 1313-1324.

[Leh96] J. Lehoczky, "Real-time queuing theory," in Proceedings of the 17th IEEE Real-Time Systems Symposium, 186-195, IEEE Computer Society Press, 1996.

[Liu73] C.L. Liu and J.W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," Journal of the ACM, 20, 1973, 46-61.

[Liu91] J.W.S. Liu, K.J. Lin, W.K. Shih, A.C. Yu, J.Y. Chung and W. Zhao, "Algorithms for scheduling imprecise computations," IEEE Computer, 24(5), May 1991, 129-139.

[Ram89] K. Ramamritham, J.A. Stankovic and W. Zhao, "Distributed scheduling of tasks with deadlines and resource requirements," IEEE Transactions on Computers, 38(8), August 1989, 110-123.

[Sha91] L. Sha, M. H. Klein, and J.B. Goodenough, "Rate monotonic analysis for real-time systems," in Scheduling and Resource Management, Kluwer, 1991, 129-156, edited by A. M. van Tilborg and G. M. Koob.

[Ste97] D.B. Stewart and P.K. Khosla, "Mechanisms for detecting and handling timing errors," Communications of the ACM, 40(1), January 1997,87-93.

[Str97] H. Streich and M. Gergeleit, "On the design of a dynamic distributed real-time environment," in Proceedings of the 5th International Workshop on Parallel and Distributed Real-Time Systems, 251-256, IEEE Computer Society Press, 1997.

[Sun96] J. Sun and J.W.S. Liu, "Bounding completion times of jobs with arbitrary release times and variable execution times," in Proceedings of the 17th IEEE Real-Time Systems Symposium, 2-11, IEEE Computer Society Press, 1996.

[Tia95] T.S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.C. Wu and J.W.S. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times," in Proceedings of the 1st IEEE Real-Time Technology and Applications Symposium, 164-173, IEEE Computer Society Press, 1995.

[Wel98] L. R. Welch, B. Ravindran, B. Shirazi and C. Bruggeman, "Specification and analysis of dynamic, distributed real-time systems," in Proceedings of the 19th IEEE Real-Time Systems Symposium, 72-81, IEEE Computer Society Press, 1998.

[Xu90] J. Xu and D.L. Parnas, "Scheduling processes with release times, deadlines, precedence and exclusion relations," IEEE Transactions on Software Engineering, 16(3), March 1990, 360-369.

Workshop Program

Monday, July 03, 2000

9:00 – 10:30
Lonnie R. Welch, Ohio University
Toward a Taxonomy of Real-Time Mission-Critical Systems

11:00– 12:00
Günter Hommel, TU Berlin
Structuring of the Seminar

14:00 – 15:30
Hermann Härtig, TU Dresden
Dresden Real-Time Operating System

16:00 – 17:00
Günter Hommel, TU Berlin
Stochastic Petri Nets for Modeling and Evaluation of Real-Time Systems - Motivation and Open Problems

17:00 –18:00
Armin Zimmermann, TU Berlin
Stochastic Petri Nets for Modeling and Evaluation of Real-Time Systems - Appropriate Net Classes For Real-Time Systems

Tuesday, July 4, 2000

9:00 – 10:30
Barbara B. Pfarr, NASA - Greenbelt
Designing a Cost Effective Communications Satellite for Mars

11:00 – 12:00
Jörn Freiheit, TU Berlin
Stochastic Petri Nets for Modeling and Evaluation of Real-Time Systems - Solutions to the State Space Explosion Problem

14:00 – 15:30
Gunter Bolch, Universität Erlangen
Performance Modelling of Time Dependent Priority Systems for Diffentiated Services in the Internet

16:00 – 17:00
Jeffery Hansen, CMU - Pittsburgh
Resource Allocating Policies for Dynamic Optimization of QoS

17:00 – 18:00

E. Douglas Jensen, MITRE - Bedford
A General Scalable Technology for Software Execution Timeliness as a Quality of Service

Wednesday, July 5, 2000

9:00 – 10:30
E. Douglas Jensen, MITRE - Bedford
A General Scalable Technology for Software Execution Timeliness as a Quality of Service (continued)

11:00 – 12:00
Scott A. Brandt, Univ. California - Santa Cruz
Soft Real-Time Processing with Dynamic QoS Level Resource Management

14:00 – 19:00
Excursion to Trier

Thursday, July 6, 2000

9:00 – 9:30
Lonnie R. Welch, Ohio University
DeSiDeRaTa: QoS and Resource Management Tools

9:30 – 10:30
David L. Andrews, University of Arkansas - Fayetteville
Virtual Foundation for Real-time Systems

11:00 – 12:00
Brett Tjaden, Ohio University
Secure-RM: Security and Resource Management for Dynamic Real-Time Systems

14:00 – 15:30
Kenji Toda, ETL -Tsukuba - Japan
Some Issues on Making Parallel / Distributed Real-time Systems

16:00 – 17:00
Claude-Joachim Hamann, TU Dresden
A Probabilistic Model to Schedule Tasks in DROPS

16:00 – 17:00
Peter K. Ibach, HU Berlin
A Scheduling Taxonomy

Friday, July 7, 2000

9:00 – 10:30
Lonnie R. Welch, Ohio University
Problem Solving Session

11:00 – 12:00
Günter Hommel, TU Berlin
Future Work

Toward a Taxonomy of Real-Time Mission-Critical Systems

Lonnie R. Welch
School of Electrical Engineering and Computer Science
Ohio University
Athens, OH 45701
welch@ohio.edu

Abstract

This speaker presented a taxonomy that characterizes common design patterns within the real-time mission-critical systems domain. Three primary design patterns have emerged from the author's study and characterization of existing real-time systems: (1) a periodic entity that uses sensor information to perform situation assessment; (2) a transient unit that performs initiation of actions in response to conditions detected by situation assessment; and (3) a transient-periodic pattern that guides actions to successful completion.

The major characteristics of the design patterns define the categories of the taxonomy, which include properties of a real-time system *and* properties of the environment. *The taxonomy enables the capture of the following features of a real-time system:*

- *Timing requirement (granularity, strictness, complexity, and abstraction level)*
- *Behavior*
- *Task relations*
- *Forms of adaptation*

The environment is a very important, but often overlooked, aspect of most real-time systems. The, the major features of environments can also be characterized within the taxonomy:

- *Characteristics*
- *Dynamics*
- *Workload (data stream size, event arrival rate, stream elements, and period)*

The taxonomy is useful for classifying technology and for characterizing applications. Such classification and characterization can be helpful for guiding real-time system developers in the selection of appropriate technology solutions. Additionally, the taxonomy can be used by researchers to identify gaps in technology, thereby contributing to the articulation of open research problems.

Dresden Real-Time Operating System

Hermann Härtig
University of Technology
Dresden, Germany
hermann.haertig@inf.tu-dresden.de

Abstract

DROPS is designed to support systems where non real-time and dynamic real-time components share resources and are present within single applications. By "dynamic" we mean systems where new real-time components come and completed or obsolate ones go and where the "worst case / normal case" ratio is much too bad to base resource assignment on worst case execution times. DROPS supports this model on various system levels, ranging from cache partitioning at the lowest level to an adaptive file system on high level. It is based on a statistical variant of the imprecise computation model.

Stochastic Petri Nets for Modeling and Evaluation of Real-Time Systems - Motivation and Open Problems

Günter Hommel, Armin Zimmermann, and Jörn Freiheit
TU Berlin
hommel@cs.tu-berlin.de azi@cs.tu-berlin.de freiheit@cs.tu-berlin.de

Abstract

One of the challenges in real-time systems today is the increasing complexity of upcoming applications and the associated goals for their design. Standard methods are often not capable of handling these new problems anymore. Appropriate methods for the system specification and analysis of quantitative and qualitative requirements are therefore needed.

One way to advance the analysis of complex real-time systems is the development of mathematical modeling and analysis frameworks for this application domain. It is necessary to prove logical and temporal properties based on this model. Example measures could be the liveness and performance of the system, or the timeliness of task execution [12]. All these measures should take into account the effect of possible failures and repairs of the system.

The authors believe that Petri nets [15] and their stochastic timed extensions can be successfully applied for real-time systems. They are generally used for the modeling and analysis of discrete event systems because of their ability to describe them in a concise and appropriate way. On the other hand, there are a lot of different analysis and simulation techniques as well as software tools available. Petri nets have already been proposed in the context of real-time systems, see e.g. [11, 17, 3, 11, 14].

In many existing real-time applications the timing of tasks can only be described with a certain accuracy using both deterministic and stochastic times. This cannot be done using Markovian stochastic Petri nets, like generalized stochastic Petri nets (GSPNs [4]). GSPNs only allow transition ,ring times to be either immediate or exponentially distributed. In contrast to this, non-Markovian SPNs offer deterministic or even more general distributions in addition to that [10]. This makes them suitable for real-time systems from the point of view of necessary ,ring time distributions. One example is the model class called deterministic and stochastic Petri nets (DSPNs [1]).

It is well known that the numerical analysis of non-Markovian SPNs (today) is only efficiently possible if in every system state there is at most one transition enabled with non-exponentially distributed ,ring time [5]. Different approaches aim at relaxing this property, like

- *approximation of deterministic transitions by substituting them with Erlang or generalized Cox distributions [8], or*

- *allowing some special cases under the name of cascaded deterministic and stochastic Petri nets [9].*

The reason for this problem is that for every non-exponential transition the remaining firing time has to be somehow memorized during the state space analysis.

One approach to escape it is to interpret the Petri net model as having a discrete underlying time scale (as opposed to continuous time as it is being done normally). This class is named discrete time DSPNs [6, 18]. Exponential distributions then naturally map to geometrical ones, because this is the memoryless distribution in the discrete case. Because in every time step the firing probabilities are given by the geometrical distribution, and the remaining firing times (having only some discrete values) can be stored together with the marking, the performance analysis does not pose mathematical problems. Moreover, as immediate and deterministic transitions are special cases of the geometric distribution, there is no problem in having any number of them enabled together in a marking. This overcomes the main principal problem of analyzing real-time systems with stochastic Petri nets.

The analysis is then of course based on a mapping of the (reduced) reachability graph onto a discrete time Markov chain. Unfortunately the reachability graph is bigger with respect to a continuously timed model, because for all enabled transitions the associated remaining firing times are part of the unique state description. Care has to be taken with transitions trying to fire at the same instant of time. Proper association of priorities avoids the problem of confusions in this case.

While this class of Petri nets offers suitable modeling and analysis means for simple real-time systems, it seems not to be adequate for more complex ones. The appearing problems fall into two categories:

- *Models of complex systems are not very well understandable.*
- *The state space gets very large and prevents analytical tractability.*

Colored Petri nets [13] offer tokens with attributes and a hierarchical composition of the model, thus overcoming the first mentioned problem. The *wide applicability is, however, paid for with a rather large amount of textual definitions. It should however be possible to develop a dedicated class of colored Petri nets for real-time systems that avoids this problem and makes the modeling easier. At the level of the reachability graph colored Petri nets can be treated like uncolored ones, making all techniques developed for the latter available as well. The authors propose to use dedicated colored Petri nets incorporating non-Markovian firing times. An underlying discrete timing for this net class should be investigated to combine the advantages of both techniques. The same idea has already been successfully applied to manufacturing systems [22, 19]*

The second problem is known under the term state space explosion. For all well-known analysis methods the computation of the whole state space is necessary. But for many systems of real-life size the state space is too large to be handled. This necessitates advanced techniques to overcome this limitation. For an overview of the wealth of techniques to overcome these limitations see e.g. [2].

The idea of one of it, namely decomposition methods, is to divide the whole system into smaller subsystems, so that the computation of the full state space can be avoided. To compute the interaction among the subsystems the authors present an iterative algorithm [7, 20] in which a net called basic skeleton contains a simplification of the whole net. The main advantage of decomposition methods is that the analysis of small subsystems needs less memory and time. Despite the need to iteratively repeat the algorithm, the results are computed faster than with standard methods. The disadvantage is that only approximate results are computed, although experiences show that the error is acceptable in most cases. This method is based on the ideas presented in [16] for uncolored models.

In addition to the theoretical development of new techniques, the applicability can only be shown when the algorithms are being implemented in a software tool. The modeling and analysis tool TimeNET [21], developed at TU Berlin, offers non-Markovian colored Petri net modeling and numerical analysis as well as simulation algorithms.

The described propositions are expected to have the potential of capturing both stochastic and deterministic behavior in one model, thus allowing to analyze real-time systems exhibiting those two types of timing. The analysis can be used to check the functionality of the system under investigation, as well as answer performance and dependability issues under real-time constraints.

In the future, modeling and analysis techniques based on Petri nets should be investigated and advanced for the application area of real-time systems.

References

[1] M. Ajmone Marsan and G. Chiola, On Petri Nets with Deterministic and Exponentially Distributed Firing Times, in: G. Rozenberg, ed., Advances in Petri Nets 1987, Lecture Notes in Computer Science, Vol. 266 (Springer Verlag, 1987) 132-145.

[2] G. Balbo and M. S. (Eds.), Performance Models for Discrete Event Systems with Synchronisations: Formalisms and Analysis Techniques (Universidad de Zaragoza, Spain, 1998), MATCH Advanced School.

[3] G. Bucci and E. Vicario, Compositional Validation of Time-Critical Sys tems Using Communicating Time Petri Nets, IEEE Transactions on Software Engineering 21 (12) (1995) 969-992.

[4] G. Chiola, M. Ajmone Marsan, G. Balbo, and G. Conte, Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications, IEEE Transactions on Software Engineering 19 (2) (1993) 89-107.

[5] G. Ciardo, R. German, and C. Lindemann, A Characterization of the Stochastic Process Underlying a Stochastic Petri Net, IEEE Transactions on Software Engineering 20 (1994) 506-515.

[6] G. Ciardo and R. Zijal, Well-De,ned Stochastic Petri Nets, in: Proc.4th Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'96) (San Jose, CA, USA, 1996, IEEE Computer Society Press) 278-284.

[7] J. Freiheit and A. Zimmermann, Extending a Response Time Approximation Technique to Colored Stochastic Petri Nets, in: Proc. 4th Int. Workshop on Performability Modeling of Computer and Communication Systems (PMCCS), (College of William and Mary, Williamsburg, VA, USA, 1998) 67-71.

[8] R. German, Analysis of Stochastic Petri Nets with Non-Exponentially Distributed Firing Times, Dissertation, Technische Universit-at Berlin, 1994.

[9] R. German, Cascaded Deterministic and Stochastic Petri Nets, in: Proc. 3rd Int. Meeting on the Numerical Solution of Markov Chains (Zaragoza, 1999) 111-130.

[10] R. German, Performance Analysis of Communication Systems, Modeling with Non-Markovian Stochastic Petri Nets (John Wiley and Sons, 2000).

[11] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezze, A Unified High Level Petri Net Formalism for Time-Critical Systems, IEEE Transactions on Software Engineering 17 (2) (February 1991) 160-172.

[12] W. Halang, R. Gumzej, M. Colnaric, and M. Druzovez, Measuring the Performance of Real-Time Systems, Int. Journal of Time-Critical Computing Systems 18 (2000) 59-68.

[13] K. Jensen, Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, EATCS Monographs on Theoretical Computer Science (Springer Verlag, 1992).

[14] A. Mazzeo, N. Mazzocca, S. Russo, and V. Vittorini, A Systematic Approach to the Petri Net Based Speci,cation of Concurrent Systems, Real-Time Systems 13 (1997) 219-236.

[15] T. Murata, Petri Nets: Properties, Analysis and Applications, Proceedings of the IEEE 77 (4) (1989) 541-580.

[16] C. J. Perez-Jimenez, J. Campos, and M. Silva, On approximate performance evaluation of manufacturing systems modelled with weighted T-systems, in: Proc. of the Symp. on Discrete Events and Manufacturing Systems, IMACS-IEEE SMC Multiconf. on Computational Engineering of Systems Applications (CESA 96) (Lille, France, 1996).

[17] B. Ravindran, L. Welch, and C. Kelling, Building Distributed, Scalable,Dependable Real-Time Systems, in: Proceedings of the 10th IEEE Int. Conf. on Engineering of Computer Based Systems (1997) 452-459.

[18] R. Zijal, Analysis of Discrete Time Deterministic and Stochastic PetriNets, Dissertation, Technische Universit-at Berlin, October 1997.

[19] A. Zimmermann, K. Dalkowski, and G. Hommel, A Case Study In Modeling And Performance Evaluation Of Manufacturing Systems Using Colored Petri Nets, in: Proc. of the 8th European Simulation Symposium (Genoa, Italy, 1996) 282-286.

[20] A. Zimmermann, J. Freiheit, and A. Huck, A Petri net based design engine for manufacturing systems, Int. Journal of Production Research, special issue on Modeling, Specification and Analysis of Manufacturing Systems(accepted for publication).

[21] A. Zimmermann, J. Freiheit, R. German, and G. Hommel, Petri Net Modelling and Performability Evaluation with TimeNET 3.0, in: 11th Int. Conf. on Modelling Techniques and Tools for Computer Performance Evaluation (Schaumburg, Illinois, USA, 2000) 188-202, LNCS 1786.

[22] A. Zimmermann and G. Hommel, Modelling and Evaluation of Manufacturing Systems Using Dedicated Petri Nets, Int. Journal of Advanced Manufacturing Technology 15 (1999) 132-137.

Designing a Cost Effective Communications Satellite for Mars

Barbara B. Pfarr
NASA/Goddard Space Flight Center
Greenbelt, MD 20771
Barbara.Pfarr@gsfc.nasa.gov

Abstract

This presentation described the concept for a NASA mission that consists of a communications satellite orbiting the planet Mars. The objective was to describe the uncertainties of the real-time system necessary to support this mission and elicit feedback on technologies and algorithms that might benefit this concept.

MARSAT is a concept for a communications satellite or constellation of 3 satellites that will orbit around Mars whose primary purpose is a communications link from rovers, orbiting satellites, balloons and some day astronauts back to Earth. A dedicated communications satellite or set of satellites around Mars is desirable to track Mars missions throughout their life spans, including the crucial period of Mars approach. It should provide higher powered communications centrally, reducing the need for high power communications on each rover, orbiting satellite, balloon, etc. and provide some level of emergency support to these assets that would otherwise not be possible due to the long communications time to Earth. The mission's high level goals include breakthrough increases in communications bandwidth, seamless end-to-end data flow, IP-like protocols tailored to operate over long round-trip times, highly efficient relay communications for energy-constrained rovers, orbiting satellites, balloons and other assets on or around Mars, and high accuracy timing and navigation services. Key cost drivers include launch costs, ground station antenna time, and the operations costs of a 10-or-more year mission.

The key challenges of the on-board real-time system for MARSAT are 1) the number of assents on MARS that will need to use MARSAT is unknown 2) the workload will vary due to the every two year launch opportunities 3) the high level of fault tolerance required 4) weight and power system constraints 5) multiple data delivery options (including throughput, file transfer, store and forward) that add complexity to the system 6) ground/backup intervention requirements that add complexity to any onboard automation 7) the variable relay time from Earth to Mars of eight to forty minutes, depending on planetary alignment, and 8)unknown timing requirements that will depend significantly on the design of Mars assets, including storage capacity and data volumes. The research discussed at this conference could potentially enable a real-time satellite system with this set of challenges.

Performance Modeling of Time Dependent Priority Systems for Differentiated Services in the Internet

Gunter Bolch and Lassaad Essafi
Universität Erlangen
bolch@informatik.uni-erlangen.de

Abstract

Priorities, and recently specially time dependent priorities, are used to improve the QoS of the Internet. In the first part of the talk analytical methods to determine QoS parameters as throughput or delay times in time independent and time dependent priority systems are introduced. These methods are used in the second part of the talk to obtain QoS parameters of the "Proportional Differentiated Service Archicture" of the internet. In this architecture time dependent priorities are used to achieve a given delay ratio between the traffic classes in the routers of the internet. Using the analytical results it is shown that the given delay ratio is met under heavy load conditions and a new method is introduced to meet the given delay ratio also under medium load conditions.

Resource Allocating Policies for Dynamic Optimization of QoS

Jeffery Hansen
CMU – Pittsburgh
hansen@cmu.edu

Abstract

As computer systems grow in size and complexity, assuring that these systems deliver the expected level of service is becoming more and more difficult. Most distributed systems, including the Internet, provide only a best-effort service model often resulting in slow, unsatisfactory and inconsistent service. The Amaranth project is developing technologies to provide distributed systems with quantitative resource-efficient QoS (Quality of Service) guarantees with a focus on multimedia applications such as video, audio and web browsing. Amaranth technologies include QoS negotiation, pro-active resource management, traffic forecasting and probabilistic guarantees.

Among the challenges in QoS resource management is allocating existing resources such that user satisfaction is optimized. Amaranth measures user satisfaction by user-defined "utility curves" mapping specific quality settings (e.g., a video flow with CIF resolution at 30 frames-per-second) to a numeric rating. The goal is to maximize global utility, the sum of the utility ratings over all users, while meeting resource constraints. By applying local search techniques, near-optimal solutions to this NP hard problem can be found in linear time. Amaranth also introduces the concept of a resource reserve, resources deliberately left unallocated to accommodate future resource requests. Maintaining a resource reserve dramatically reduces the frequency of QoS reconfiguration events, occurring when resources for an existing task must be reclaimed to accommodate the minimum requirements for an incoming task admission request. By modeling reserve resources as a competing pseudo-task with an appropriate utility curve, global utility values close to the theoretical optimum can be obtained with a very low rate of QoS reconfigurations.

In order to make optimal use of allocated resources, Amaranth introduces the concept of probabilistic guarantees. Since most distributed multimedia applications can tolerate a small amount of packet loss without significant degradation in delivered service, roviding probabilistic guarantees results in a more effective tradeoff between quality of service and resource utilization. Under Amaranth's probabilistic guarantee model, the number of concurrent network flows that can be admitted on a fixed bandwidth link at a 99% guarantee level can be double the number that can be admitted under traditional bandwidth reservation schemes such as RSVP.

A General Scalable Technology for Software Execution Timeliness as a Quality of Service

E. Douglas Jensen
MITRE – Bedford
jensen@real-time.org

Abstract

Traditional real-time computing and computing concepts and technologies are focused primarily on device level, static, centralized, regulatory monitoring and control. In that context, it is presumed that real-time computing involves only hard deadlines in the microsecond timeframe, and that all these deadlines must always be met. But that is a relatively simple special case. Real-time computing in general is about acceptable predictability of timeliness; the time constraints need not be deadlines, their timeframes may be from microseconds to megaseconds, and the scheduling optimality criteria may be soft – e.g., minimize the mean tardiness weighted by importance – as well as hard. Many real-time applications are much more complex: higher level, more dynamic, more distributed – and yet have mission-critical and even safety–critical timeliness optimality and predictability requirements. Traditional real-time computing concepts and technologies do not scale up well to such applications. A rich model for expressing time constraints, and scheduling optimality and predictability, as timeliness quality of service can result in more scaleable, adaptable, and general concepts and technologies. The utility function model has been theoretically evaluated and experimentally demonstrated to be a cost-effective approach to timeliness quality of service for a variety of real-time systems from devices up to C^2 systems of systems.

Soft Real-Time Processing with Dynamic QoS Level Resource Management

Scott A. Brandt
Univ. California - Santa Cruz
sbrandt@cse.ucsc.edu

Abstract

High throughput time-constrained applications such as video display and processing, real-time hardware control, and virtual reality are being executed on desktop systems at rates formerly achievable only with special-purpose computers with real-time operating systems or no operating sytem at all. Unlike traditional real-time applications, these applications are executed on top of general purpose operating systems and their real-time constraints are non-critical. However, some level of real-time support is still needed in order to provide acceptable levels of application performance.

This work focuses on a soft real-time method called Quality of Service Levels in which applications are developed with multiple modes in which they can execute. Each mode provides a different output quality and consumes a different amount of resources. A centralized resource manager dynamically allocates the available resources among the running applications according to the levels that the applications have specified in an attempt to maximize the overall system quality. This work explores the QoS Level model of soft real-time application execution and presents result from and analysis of a working system based on this model. It includes a discussion of the development of the system, robust solutions to difficult implementation issues, resource allocation theory and an analysis of several resource allocation algorithms, and metrics for evaluating the performance of the system.

DeSiDeRaTa: QoS and Resource Management Tools

Lonnie R. Welch
School of Electrical Engineering and Computer Science
Ohio University
Athens, OH 45701
welch@ohio.edu

Abstract

Schedulability analysis (SA) approaches that are based on a priori information work well in many application domains and allow pre-deployment guarantees of real-time performance. However, certain real-time applications must operate in highly dynamic environments, thereby precluding accurate characterization of the applications' workloads by static models. The DeSiDeRaTa project is considering issues which arise when developing SA techniques that are appropriate for systems which experience large variations in workload. We are designing and implementing middleware that performs dynamic SA (DSA) and manages computing resources by assessing QoS metrics and resource utilization metrics that are determined a posteriori.

Time-constrained systems which operate in dynamic environments may have unknown worst-case scenarios, may have large variances in the sizes of the data and event sets that they process (and thus, have large variances in execution latencies and resource requirements), and may not be characterized statically, even by time-invariant statistical distributions. The DeSiDeRaTa project provides a specification language for describing such environment-dependent features of dynamic real-time systems. Also provided is an abstract model that is constructed (a priori) from the specifications, and is augmented (a posteriori) with the state of environment-dependent features. The model is being used to develop algorithms for QoS (quality-of-service) monitoring, QoS diagnosis, and resource allocation analysis. Experimental results show the effectiveness of the approach for specification of real-time QoS, detection and diagnosis of QoS failures, and restoration of acceptable QoS via reallocation of distributed computer and network resources.

In addition to applying the DeSiDeRaTa paradigm of path-based computing to the DynBench benchmark suite, we have also employed it successfully to model application systems in many domains, including C^2 systems (battle mgmt, aviation, space, and AAW); cooperating autonomous agents (squads of mobile robots and satellite constellations); and a web server.

Virtual Foundation for Real-time Systems

David L. Andrews
University of Kansas
dla@engr.uark.edu

Abstract

In this talk, the domain of inter-process communications (IPC) in distributed real time systems is discussed. Within this domain, IPC requirements for scalability across heterogeneous systems, functional relocation, fault tolerance, and support for application driven quality of service is outlined. The IPC services include logical, as opposed to fixed addressing, for communications channels allowing application relocation and balancing, addition and deletion of heterogeneous nodes, and fault tolerance. The need for one to many, and many to one operations to support data distribution and reduction, and system expansion are also discussed. IPC support for aperiodic and high bandwidth transient data patterns are also discussed. A specific example is presented from a large, complex real time embedded application. The example system is an asynchronous fibre optic ring based system with over 120 processor nodes running a critical real time application. The system requirements include guaranteed operation during high transient communications loads, fault tolerance, and graceful system degradation. The implementation of the real time IPC developed for the system is described.

Secure-RM: Security and Resource Management for Dynamic Real-Time Systems

Brett Tjaden
School of Electrical Engineering and Computer Science
Ohio University
Athens, OH 45701
tjaden@ohio.edu

Abstract

Mission critical real-time systems often function in environments that cannot be modeled with static approaches. Because of their (externally-driven) wide dynamic range of system operation, the number of data elements to be processed in an arbitrary period is unknown at the time of system engineering (other than in an extremely pessimistic worst case sense). While it may be possible to determine a theoretical upper bound on the number of data items, the construction and maintenance of system components to handle worst case conditions can be extremely costly. To accommodate such dynamic mission critical real-time systems, it is useful to design computing systems that allow reconfiguration and reallocation of resources by sharing a pool of distributed computational resources. Unfortunately, the problem of continuously providing critical system functions in such dynamic real-time environments is exacerbated when one considers attack vulnerability.

Thus, we are developing Secure-RM, a security management system that combines an intrusion detection system (IDS) with adaptive resource management middleware. The INBOUNDS IDS [1], a product of the Laboratory for Intelligent, Real-Time, Secure Systems (LIRTSS) at Ohio University, is a network-based, real-time, hierarchical software system for misuse detection and anomaly detection. Intrusion events, such as pre-attack probes and denial of service attacks, are detected using monitoring tools such as TCPTrace [2, 7-10] and are reported to Secure-RM, which employs artificial intelligence techniques (specifically, Bayesian and Belief Networks [3, 4]) for deriving impacts of attacks on operational functions and mission goals. A strong belief in an attack strategy triggers a resource reallocation for reflexive action. Secure-RM utilizes DeSiDeRaTa resource management middleware [5, 6] (a product being developed within LIRTSS under funding from DARPA) to perform semi-automatic course of action development and execution.

Some Issues on Making Parallel / Distributed Real-time Systems

Kenji Toda
ETL -Tsukuba – Japan
toda@etl.go.jp

Abstract

The problems of (mission critical) real-world applications are as follows. It is hard to know all cases during design phase, and it is difficult to develop programs for large-scale RT systems. Compatibility between dependability and high performance is also solved with our approach which deals with an unavoidable overload situation caused by unpredictability including hardware failure of a Imprecise computation which is used to control the precision of a task in order to accomplish the mission of system. We had a discussion on this model and the experimental evaluation of a car collision avoidance system where utility functions are used to control the precision by anticipating potential collisions and determine operations for avoiding the collision. We have shown the benefit and potential abilities of our approach .

A Probabilistic Model to Schedule Tasks in DROPS

Claude-Joachim Hamann
University of Technology
Dresden, Germany
claude.hamann@inf.tu-dresden.de

Abstract

The micro kernel underlying the DROPS system supports reservation priorities. An user level scheduler can assign and revoke reservation priorities allowing threads to run at an assigned scheduling priority during the reserved time. So a higher resource usage can be reached than using fixed priority scheduling only. The talk presents a stochastic model based on separating mandatory parts of an application from optional parts. All mandatory parts must meet their deadline but the "success" of an optional part is measured by an Quality-of-Service (QoS) parameter (percentage of completed parts). The model enables to calculate the reservation times depending on the requested QoS.

A Scheduling Taxonomy

Peter K. Ibach
HU Berlin
ibach@informatik.hu-berlin.de

Abstract

Since a multitude of different real-time systems have emerged during the past, there is the need for integrating the variety of issues that have been addressed mostly in separated ways.A Taxonomy is presented that groups the different properties of these systems into classes. Firstly it is stated that classification is a matter of perspective. Then, the Classification problem is approached from the perspective of scheduling and its various cascading aspects such as workload, scheduling algorithm and objective function. For taxonomy representation a graphical notation called "snowflake diagram" is introduced that spans a hierarchy of aspects and associated properties.

With focus on stochastic real-time systems, the role of predictability is emphasized in particular. A brief overview on prediction methods is presented. Stochastic real-time research, as concluded, should address the open questions in the domain of rather large, complex, dynamic, distributed, high level, real-time systems, such as how to reason about predictability as a spectrum ranging from maximum predictability (determinism) at one end-point to minimum predictability at the other end-point. Finally, controversial definitions and opinions of real time and related terminology are presented to stimulate discussion on how these specific issues of stochastics and predictability could be adequately incorporated.