1989

Performability Evaluation of Gracefully Degradable Systems

Christoph W. Lindemann

Report Number:

89-887

PERFORMABILITY EVALUATION OF
GRACEFULLY DEGRADABLE SYSTEMS

Christoph W. Lindemann

CSD-TR-887
June 1989

PERFORMABILITY EVALUATION OF GRACEFULLY DEGRADABLE SYSTEMS

Christoph W. Lindemann
Computer Sciences Department
Purdue University
West Lafayette, IN 47907

ABSTRACT

This paper presents a method for evaluating the performability of repairable degradable systems based upon combining Generalized Stochastic Petri Nets and product-form queueing network models. The method takes into consideration the transient behavior during a change in the system structure. To reduce the computational effort for the solution a hierarchical decomposition method is employed. Since it is assumed that all tasks in the system belong to a single class, the aggregation of a submodel to a flow-equivalent transition leads to an exact solution.

1. INTRODUCTION

The evaluation of systems with degradable performance (e.g. communication networks, distributed systems, etc.) requires unified performance-reliability measures because such systems are able to operate at varying degrees of performance. We follow the terminology introduced by Meyer [12] and call these different performance outcomes accomplishment levels. The performability of a system S is defined as the probability measure

$$p_S(B) \; = \; \textit{the probability that the system performs at a level in } B$$

In this general definition B denotes a measurable subset of the - eventually uncountable - set of accomplishment levels A. Performability unifies the usual notions of performance and reliability, and contains both of them as special cases. To evaluate non-repairable systems Meyer has defined a two-dimensional discrete stochastic process. He determined the state probabilities by an aggregation over the state space [12]. This approach assumes that the system is nearly completely decomposible in the sense of Courtois [8] and neglects the transient behavior during a change in the system configuration caused by the occurrence of a failure. Meyer, Movaghar, and Sanders have defined stochastic activity networks (SANs) which allow the description of a system at a level higher than the underlying stochastic process [13]. As in Generalized Stochastic Petri Nets, GSPNs [1], there exist two different types of transitions in a SAN model (there called activities), timed activities and instantaneous activities.

Additionally, cases and gates may be associated with activities of a Stochastic Activity Network. Therefore, SAN models provide more flexibility than GSPNs, but by the same token their solution process is more complex and thus requires more computational effort.

This paper presents an approach for determining the performability of repairable degradable systems based upon combining Generalized Stochastic Petri Net and product-form queueing network, PFQN, models. The proposed modeling technique represents explicitly the transient behavior during the reconfiguration process. The bulk arrival of tasks at a fault-free processor is modeled by enabling a single intermediate transition. We employ a hierarchical decomposition method to reduce the computational effort for the solution. A compact GSPN model is defined in which the processor subsystem and the structure process is represented in detail. The I/O subsystems which have already been evaluated separately are represented in this GSPN model by one flow-equivalent transition with a marking dependent firing rate. The organization of this paper is as follows. Section 2 provides a general description of the decomposition method. The approach for evaluating the performability is introduced in section 3. Section 4 illustrates this approach by considering a system consisting of two processors and three I/O subsystems. It is shown how to derive the performability of this system from the steady-state solution of the compact GSPN model.

2. DESCRIPTION OF THE HIERARCHICAL DECOMPOSITION METHOD

Balbo, Bruell, and Ghanta have introduced a method for combining GSPN and PFQN models for systems with several job classes. They have presented approximate models for priority queueing schemes, software blocking phenomena, and other complex system behavior [2], [3], [4]. We follow a similar idea, but restrict ourselves to single-class queueing networks which do not possess a product-form solution. In particular, we consider models possessing one or more parts which can be represented by a PFQN. We assume that these submodels which satisfy the BCMP conditions can be identified. Each of these submodels is then represented by means of a PFQN and studied in isolation. The throughput is determined for each feasible number of customers that may use it. A compact GSPN model is defined in which each part of the model already analyzed separately is replaced by a flow-equivalent transition with a marking dependent firing rate. Due to the existence of user-friendly software tools the solution process can be completely automated. We have employed the packages GreatSPN [7] and RESQ [11] for solving GSPN and PFQN models, respectively.

This decomposition method introduces some error into the solution, only if there exists an interaction between a flow-equivalent transition and the complementary subnet of the compact GSPN. Such a case is given for instance when the firing rate of a flow-equivalent transition is defined using the number of tokens in two or more places [2], [3]. If it is assumed that all tasks belong to a single class, the firing rate of the flow-equivalent transition can be defined by using only the number of tokens of its single input place. In this case there is no interaction between a flow-equivalent transition and the complementary subnet of the compact GSPN. Thus, the aggregation of a GSPN subnet to a flow-equivalent transition leads here to an exact solution. A semi-formal proof for this observation is given in [10].

3. THE MODELING APPROACH FOR EVALUATING THE PERFORMABILITY

We consider a class of systems consisting of N processors and L different Input/Output subsystems, and model a system as a central server network [9]. To construct the performability model the following assumptions are made:

(a) The fault-tolerance is achieved by reconfiguration of a system with multiple functional units of the same type,

(b) This reconfiguration process can be done in zero time,

(c) Only processor failures are considered,

(d) At least one processor is available, total breakdown cannot be tolerated,

(e) All tasks processed by the system belong to a single class, they have the same distribution of service time and the same I/O routing probabilities,

(f) Service times as well as the failure/repair times have an exponential distribution.

Thus, each processor is modeled as a M/M/1 system and each I/O subsystem is represented by a M/M/m system. A task enters the queue of the i-th processor with probability p_i ($i=1,2,..,N$). After a task has obtained a CPU burst, it continues its execution at one of the I/O subsystems with probability q_j ($j=1,2,..,L$) or its execution is completed and the task exits the system with probability p_{N+1}. In the latter case it is assumed that this task is immediately replaced by a new arriving task at one of the processors. Tasks which have obtained service from an I/O subsystem are returning to one of the processors and the whole process repeats itself. Since we consider systems with graceful degradation the model has additionally to represent the structure process which is particular to such systems [12]. Hence, the routing probabilities $p_1,p_2,..,p_N$ change dynamically due to processor failures or completed repairs. This feature violates against the conditions of the BCMP theorem [5]. Therefore, the entire system cannot be modeled by means of a PFQN. Since the nonproduct-form properties of the model are restricted to the subsystem comprised of the processors, we propose to employ the decomposition method described in the previous section. Since the part of the model which is comprised of the L Input/Output subsystems satisfies the conditions of the BCMP theorem, it is solved in isolation (off-line analysis [9]) using a PFQN. The objective of this off-line analysis lies in the construction of a flow-equivalent server [6]. This flow-equivalent server will be encoded in a compact GSPN model by a single timed transition with a marking dependent firing rate. As a result of this aggregation the GSPN must only provide a detailed description of the processor subsystem and the structure process.

4. AN EXAMPLE: A TWO-PROCESSOR SYSTEM

In this section the feasibility of our approach is illustrated by an application. We consider a gracefully degradable system comprised of two processors and three different I/O subsystems with two similar devices each. It is shown how to derive the performability from the steady-state solution of the compact GSPN.

4.1 THE PFQN PART OF THE MODEL

Each of the three I/O subsystems is modeled as a M/M/2 queueing system (see Figure 1). The throughput of this submodel is determined for each feasible number of tasks k that may use it ($k = 1,2, \ldots, P$). For this purpose a dummy station is introduced to the queueing network. As a result a P-dimensional throughput vector is defined which determines the marking dependent firing rate of a flow-equivalent transition. A compact GSPN is defined in which all the I/O subsystems are represented by this flow-equivalent transition.

4.2 THE COMPACT GSPN MODEL

The GSPN model depicted in Figure 2 provides a detailed description of the processor subsystem. The I/O subsystems already evaluated separately are represented by the flow-equivalent transition T_{14}. Its marking dependent firing rate is defined using the throughput vector of the PFQN submodel. The subnet which represents the structure process is suitable for both symmetric and asymmetric two-processor systems. This means, this subnet distinguishes between a failure of the first processor and a failure of the second one. The repair times of the processors are also represented by different timed transitions in the GSPN.

The task control flow as well as the appropriate routing probabilities are represented in the GSPN model by the decision places P_1 and P_8. The routing probabilities are defined using a random switch for the corresponding immediate transitions (e.g. $(p_1; p_2)$ for the transitions t_1 and t_2). Each of the P tokens depicted in Figure 2 in place P_1 represents a task in the system. In each tangible state these P tokens are distributed among the places P_2, P_3 and P_9 modeling service requests at the corresponding resources. The initial structure state of the system, namely that both processors are working fault-free, is represented by a token in place P_4. The presence of this token enables the timed transition T_5. Its firing rate is given by the cumulative lifetime of both processors. The failure of a processor is modeled by the firing of this transition. This event causes the moving of the token from the place P_4 to P_5. In this vanishing state only the immediate transitions t_6 and t_7 are enabled. The one that fires is determined by the specific processor which fails. We will discuss only one alternative because the other behaves correspondingly. Say, the first processor fails and thus, the transition t_6 fires. Now, only the immediate transition t_8 is enabled. It fires so often until the place P_2 contains no more token. This course of events models explicitly the bulk arrival of tasks at the other fault-free processor during the reconfiguration process of the system. The transition t_1 is then disabled by the inhibitor arc from place P_6. Therefore, tasks requesting service at a processor (tokens in place P_1) are forced to the ready queue of the second processor (place P_3). In the current structure state of the system the repair time of processor 1 is represented by the firing delay of the timed transition T_{10}. Its firing moves the token from place P_6 to P_4 and terminates the disabling of transition t_1. Thus, the system is brought back to its original structure state in which both processors are working fault-free and the whole process starts over.

The model description reveals that the method described in this paper provides a complete description of the system. It considers not only the task control flow in a fixed structure state, but also the transient behavior during a reconfiguration period of the system. The bulk arrival of tasks at the other fault-free processor is modeled by enabling the single immediate transition t_8 or t_9, respectively, and so moving all the tokens currently located in place P_2 to P_3 or vice-versa.

The computational complexity for solving a GSPN model depends only on the number of its tangible states because they determine the number of states of the underlying Markov chain [1], [7]. The state space cardinality of the compact GSPN can be derived using a well-known formula from combinatorics which determines the number of ways to distribute P tokens among N places [14].

$$|S_{compact}| = \binom{P+2}{2} + 2\binom{P+1}{1} = O(P^2)$$

In the compact GSPN exists only a quadratic dependence between the number of tangible states and the marking parameter P. A detailed GSPN representation of the entire system would lead to a state space cardinality of order $O(P^4)$ [10]. Thus, the decomposition/aggregation method yields to a significant reduction of the state space.

4.3 DERIVATION OF THE PERFORMABILITY

We define the accomplishment levels as the configurations in which the system can operate. Thus, the set of accomplishment levels A is defined as $A = \{ok, f1, f2\}$ where:

ok = *System is working fault-free*
$f1$ = *Processor1 has failed*
$f2$ = *Processor2 has failed*

To define the performability model one has to consider the following values associated with each accomplishment level:

(a) The probability that the system operates at this particular configuration

(b) The reward rate associated with this configuration.

The probabilities of the structure states are directly obtained from the steady-state solution of the GSPN. They are given by the following formulas:

$P_{ok} = P(\#P4 = 1)$
$P_{f1} = P(\#P6 = 1)$
$P_{f2} = P(\#P7 = 1)$

The reward rate associated with an accomplishment level is defined as the throughput of the system assuming that it is working in the appropriate configuration. In other methods proposed earlier these reward rates were determined by computing separately the throughput of each feasible configuration of the system [12], [13]. A major advantage of the method described in this paper lies in that both the reward rates and the performability are directly obtained from the steady-state solution of the compact GSPN. The utilization of a processor in a specific structure state is determined by a conditional marking probability which can be computed by GreatSPN [7]. Each reward rate is derived by the product of the corresponding utilization of a processor with its service rate.

$U_{ok1} = P(\#P2 > 0 \mid \#P4 = 1)$

$U_{ok2} = P(\#P3 > 0 \mid \#P4 = 1)$

$U_{f1} = P(\#P2 > 0 \mid \#P7 = 1)$

$U_{f2} = P(\#P3 > 0 \mid \#P6 = 1)$

$D_{ok} = U_{ok1}*S_1 + U_{ok2}*S_2$

$D_{f1} = U_{f1}*S_1$

$D_{f2} = U_{f2}*S_2$

The combination of the probabilities for the structure states with the corresponding reward rates defines the effectivity of the system $eff(S)$ [12].

$eff(S) = D_{ok}*P_{ok} + D_{f1}*P_{f1} + D_{f2}*P_{f2}$

4.4 NUMERICAL RESULTS

To illustrate the technique presented in this paper we give a numerical example. Suppose both processors have a service rate of 100 requests/sec, a failure rate of 0.005/h and a repair rate of 0.3/h. The service rates of the I/O subsystems are 40 requests/sec, 25 requests/sec and 20 requests/sec, respectively. The routing probabilities are assumed as $p_1 = p_2 = 0.5$, $p_3 = 0.1$, $q_1 = 0.4$, $q_2 = 0.3$, and $q_3 = 0.2$

The difference in the order of magnitude between the values for the service rates and the values for the failure/repair rates may cause stiffness. We overcome this problem by employing the Gauss elimination algorithm for solving the linear system defined by the global balance equations of the underlying Markov chain. The version of this algorithm provided by GreatSPN still yields a good numerical accuracy for solving GSPN models in which the firing rates are differing up to eight orders of magnitude [7]. Since this method requires subtantially more computation time than the iterative Gauss-Seidel method, the Gauss elimination method can only be employed in practise for solving GSPN models with a small state space. The following results are obtained for the probabilities of the structure states:

$P_{ok} = 0.96774$

$P_{f1} = 0.01613$

$P_{f2} = 0.01613$

Since the routing probabilities p_1 and p_2 are equal and both processors have the same service rate, the corresponding reward rates associated with the accomplishment levels $f1$ and $f2$ are equal. Therefore, the reward rates for the accomplishment level $f2$ are omitted in Table 1.

P	D_{ok}	D_{f1}	$eff(S)$
1	23.810	23.807	23.810
2	46.306	45.058	46.270
3	65.764	61.727	65.640
4	81.668	73.725	81.418
5	94.438	82.078	94.047
6	104.728	87.827	104.187
7	113.102	91.769	112.433
8	120.012	94.454	119.195
9	125.782	96.272	124.838
10	130.652	97.484	129.590

Table 1. Numerical results

ACKNOWLEDGEMENTS

The author would like to thank Dan Marinescu for his suggestions which have improved the quality of this paper. Furthermore, he is grateful to Detleff Schmid and Helena Szczerbicka for their support during the earlier steps of this work at the University of Karlsruhe, West Germany.

REFERENCES

[1] M. Ajmone-Marsan, G. Balbo and G. Conte, ''A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of a Multiprocessor System'', ACM Trans. Comp. Systems, 2, 1984, pp. 93–122.

[2] G. Balbo, S.C. Bruell and S. Ghanta, ''Modeling Priority Schemes'' Performance Evaluation Review, Special Issue Vol. 13, No. 2, August 1985, pp. 15–26.

[3] G. Balbo, S.C. Bruell and S. Ghanta, ''Combining Queueing Network and Generalized Stochastic Petri Net Models for the Analysis of Some Software Blocking Phenomena'', IEEE Trans. SE, 12, 1986, pp. 561–576.

[4] G. Balbo, S.C. Bruell and S. Ghanta, ''Combining Queueing Networks and Generalized Stochastic Petri Nets for the Solution of Complex Models of System Behavior'' IEEE Trans. Comp., 38, 1988, pp. 1251–1268.

[5] F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios, "Open, Closed and Mixed Networks with Different Classes of Customers", Journal ACM, 22, 1975, pp. 248–260.

[6] K.M. Chandy, U. Herzog and L. Woo, "Parametric Analysis of Queueing Network Models" IBM Journal Res. Dev., 19, 1975, pp. 43–49.

[7] G. Chiola, "GreatSPN USERS' MANUAL", Department of Computer Science, University of Turin, September 1987.

[8] P.J. Courtois, "Decomposability, Instability and Saturation in Multiprogramming Systems", Comm. ACM, 18, 1975, pp. 371–377.

[9] P.J. Denning and J.P. Buzen, "The Operational Analysis of Queueing Systems", ACM Computing Surveys, 10, 1978, pp. 225–261.

[10] C.W. Lindemann, "Developing Performability Models for Systems with Graceful Degradation", Diploma Thesis, University of Karlsruhe, West Germany, April 1988 (in German).

[11] "RESQ Users' Guide", IBM Corp. 1982.

[12] J.F. Meyer, "Closed-Form Solutions of Performability" IEEE Trans. Comp., 31, 1982, pp. 648–657.

[13] J.F. Meyer, A. Movaghar and W.H. Sanders, "Stochastic Activity Networks: Structure, Behavior and Application", Proc. Int. Workshop on Timed Petri Nets, July 1985, IEEE-CS Press, pp. 106–115.

[14] K.S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Applications", Prentice Hall 1982.

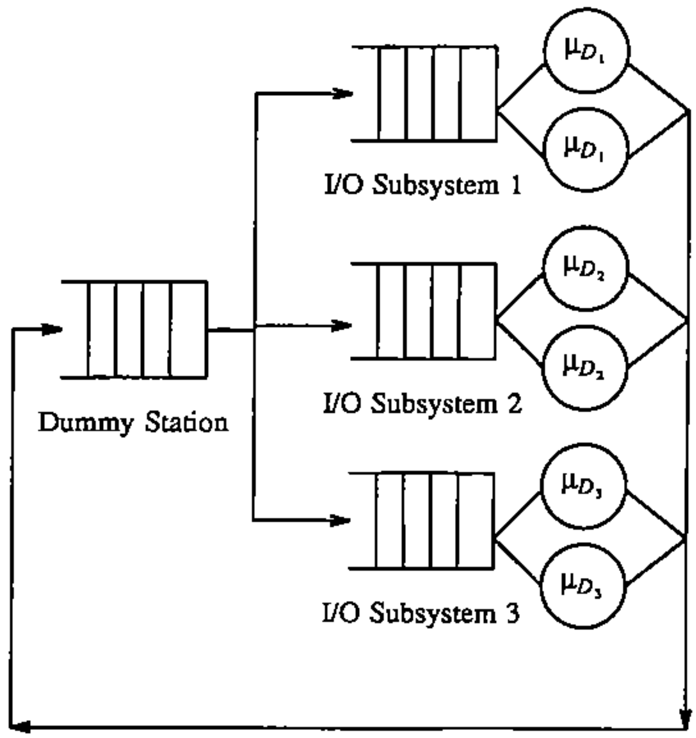

Figure 1. PFQN part of the model

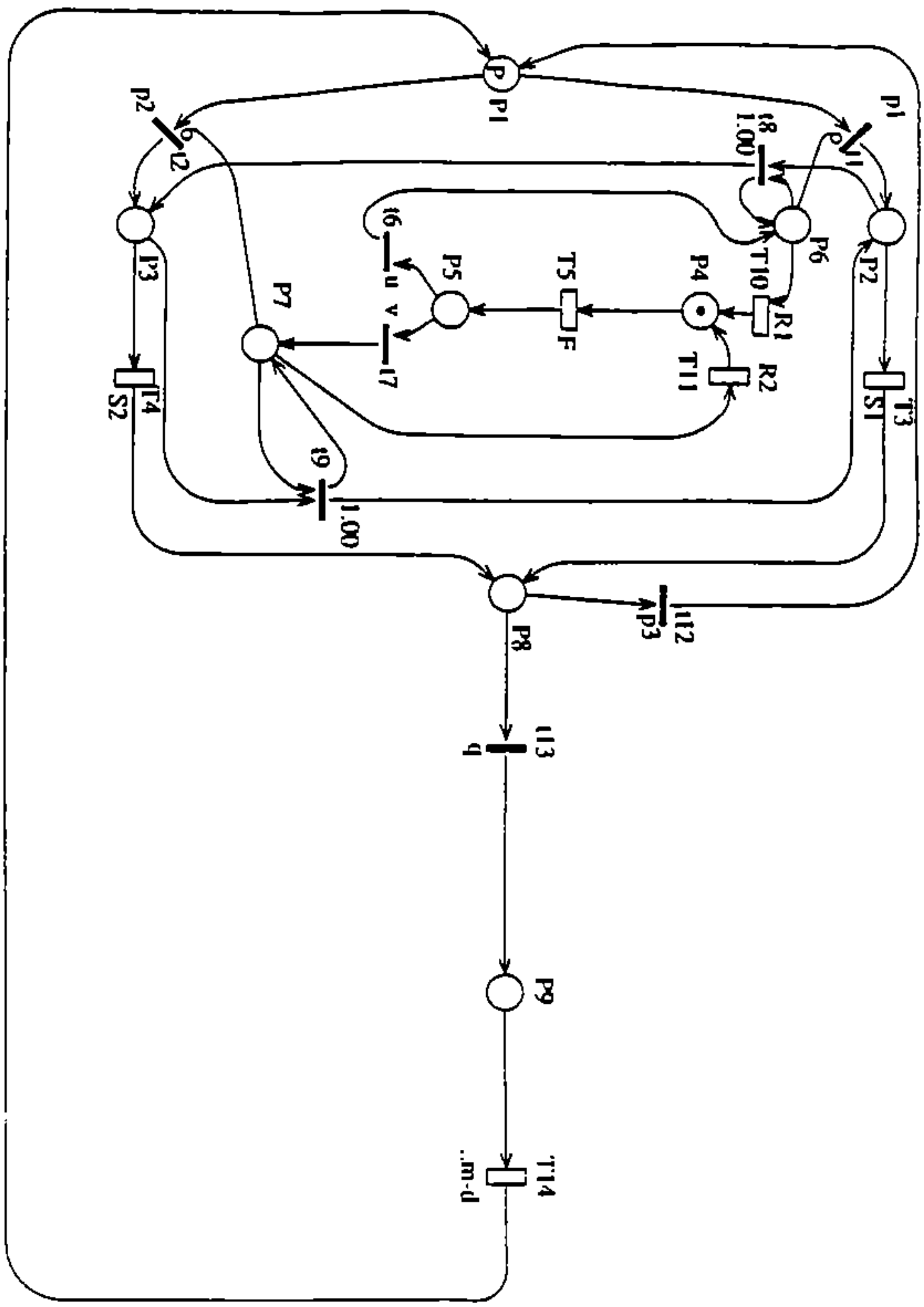

Figure 2. The compact GSPN model

FORM K
APPROVED FOR USE IN
PURDUE UNIVERSITY

Christoph Lindemann has written a paper that has been accepted for presentation at the Midwest ~~Conference~~ Symposium on Circuits and Systems in August, 1987. I think it is appropriate that it be published as a C.S. department technical report.

[illegible signature]