RBC - REVISTA BRASILEIRA DE COMPUTAÇÃO JUL./SET. 1990 RIO DE JANEIRO V. 6 N. 1 P. 3-14

Análise de desempenho com redes de Petri estocásticas

DANIEL A. MENASCÉ
Departamento de Informática — PUC/RJ

NELSON L. S. FONSECA
Computer Science Department
University of Southern California

RESUMO

A utilização de Redes de Petri na modelagem de Sistemas de Computação tem ocupado um sólido espaço na Teoria de Análise de Desempenho. A introdução do conceito de tempo na estrutura de uma Rede de Petri pode ser feita de diversas maneiras, gerando-se, assim, diferentes tipos de modelos. Este artigo introduz as principais características das variações mais importantes das chamadas Redes de Petri Estocásticas.

ABSTRACT

The use of Petri Nets in the modelling of computer systems has occupied a distinguished role in the theory of Performance Evaluation. The introduction of the concept of time in the structure of a Petri Net can be done in different ways, generating different types of models. This paper introduces the most important characteristics of the main variations of the so called Stochastic Petri Nets.

PALAVRAS-CHAVE:
◆ Análise de Desempenho
◆ Cadeias de Markov
◆ Redes de Filas
◆ Redes de Petri Estocásticas

RECEBIDO EM: 19/09/89

ENDEREÇO DOS AUTORES PARA CORRESPONDÊNCIA:
Departamento de Informática — PUC/RJ
Rua Marquês de São Vicente 225
22543 — Rio de Janeiro — RJ
e-mail: MENASCE@BRLNCC.BITNET

Computer Science Department
University of Southern California
Los Angeles, California
e-mail: NFONSECA@POLLUX.USC.EDU

1 — INTRODUÇÃO

Os vertiginosos avanços tecnológicos na área de hardware têm possibilitado a concepção de um espectro de Sistemas de Computação, que objetivam, primordialmente, a exploração do paralelismo em todos os níveis de uma arquitetura.

Se por um lado os referidos avanços demandam um perfeito entendimento da sua extensão, por outro a especificação de uma arquitetura requer a investigação dos diversos caminhos que conduzem à concretização dos requisitos funcionais da mesma.

Desta forma, criam-se estruturas matemáticas capazes de avaliar o impacto no desempenho de um sistema, devido à concorrência entre seus componentes ativos por recursos escassos.

Os modelos de Redes de Filas obtiveram sucesso pela existência de uma solução de forma simples. No caso de redes com cadeias fechadas, o desenvolvimento de um algoritmo eficiente para o cálculo da Constante de Normalização permitiu a disseminação de modelos de Redes de Filas no contexto da análise de desempenho. Posteriormente, já na década de 80, formulou-se o algoritmo da Análise do Valor Médio, que conduz a um entendimento mais natural do fenômeno de enfileiramento. Estes algoritmos [8], apesar da pequena complexidade computacional, quando comparados com a resolução de uma cadeia de Markov, restringem, sobremaneira, o poder de abstração de comportamentos observáveis em um Sistema de Computação.

Assim sendo, tem-se empreendido um grande esforço na elaboração de aproximações em Redes de Filas [2] [3] [32], de modo a transpor as restrições impostas pelas hipóteses [6] que levam à solução de uma rede sob a forma do produto. Estas aproximações permitem a análise de situações, tais como: posse simultânea de recursos, enfileiramento por memó-

ria, serialização, bloqueio etc. Em contraposição à pequena complexidade e à alta precisão dos modelos aproximados reside o fato de que a formulação de uma aproximação depende exclusivamente de heurísticas, cujo entendimento, muitas vezes, não se constitui em uma tarefa trivial.

As Redes de Petri, devido ao seu grande poder de expressão, permitem a representação adequada das situações que não permitem um tratamento exato através dos modelos de Redes de Filas. A introdução do conceito de tempo como uma variável aleatória na dinâmica de funcionamento de uma Rede de Petri dá origem às chamadas Redes de Petri Estocásticas. Estas redes constituem modelos a partir dos quais pode-se obter medidas de desempenho relativas ao sistema modelado através da resolução de uma Cadeia de Markov derivada da Rede de Petri, conforme será mostrado mais adiante.

O objetivo deste trabalho consiste na investigação das três vertentes de Redes de Petri Estocásticas que obtiveram uma sólida repercussão na literatura, de modo a propiciar ao leitor uma introdução a esta nova área do conhecimento.

2 — REDES DE PETRI

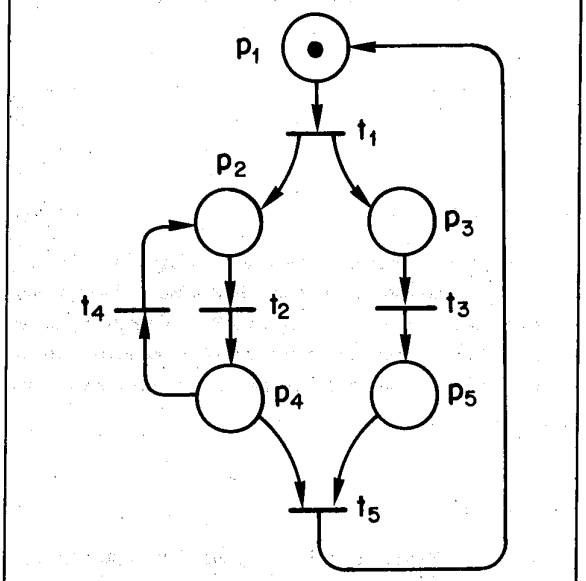
Redes de Petri constituem um modelo formal de uma computação [27]. Sua maior aplicabilidade refere-se aos comportamentos constituídos por uma seqüência de eventos, onde estes eventos podem se dar concorrentemente e cujas ocorrências estão, normalmente, sujeitas a regras de precedência e/ou concorrência.

Originariamente, as Redes de Petri (RP) foram concebidas para se estudar a sincronização e a comunicação entre processos, ou seja, o fluxo de informações em um sistema.

Um excelente tratamento a respeito das Redes de Petri pode ser encontrado em [25] e [26]. Como o objetivo deste artigo é o tratamento das Redes de Petri Estocásticas, apresentaremos a seguir uma breve introdução apenas às Redes de Petri. Pode-se definir uma Rede de Petri como um grafo bipartite e dirigido, isto é, um grafo cujo conjunto dos vértices pode ser particionado em dois subconjuntos distintos: o conjunto dos lugares e o conjunto das transições. Suas arestas unem exclusivamente elementos destes dois subconjuntos distintos, imprimindo um sentido de orientação a esta ligação. Mais de uma aresta pode ser utilizada para ligar dois vértices do grafo. Representa-se, normalmente, lugares por círculos e transições por barras finas, conforme pode ser visto na figura 1. Nota-se que somente existem arestas de lugares para transições ou de transições para lugares.

Associada à definição de RP está o conceito primitivo de **marcação**, que é uma distribuição de fichas pelos diversos lugares da rede. Formalmente, define-

FIGURA 1
UM EXEMPLO DE REDE DE PETRI



se Redes de Petri como uma quíntupla RP = (P, T, I, O, MO), onde P e T são os conjuntos dos lugares e transições, I e O são duas funções que mapeiam o conjunto das transições numa *bag* de lugares e MO representa a marcação inicial. Entende-se por *bag* uma generalização do conceito de conjunto onde se permitem múltiplas ocorrências de um mesmo elemento. A função entrada, I, relaciona todos os lugares que possuem arcos terminais numa dada transição e a função saída lista todos os lugares nos quais chegam arcos de uma certa transição.

O "funcionamento" de uma rede é determinado pela movimentação das fichas pelos lugares da rede. Para que se possa ter uma mudança na distribuição das mesmas, deve ocorrer um disparo de uma transição. Por outro lado, o disparo de uma transição só pode ocorrer quando a mesma está habilitada, ou seja, quando o número de fichas em cada um dos lugares de entrada é igual ou superior à multiplicidade dos arcos que vão destes lugares para a transição [26]. O disparo de uma transição gera uma nova marcação. O conjunto mínimo de marcações através do qual o "funcionamento" da rede pode ser caracterizado é denominado **conjunto de alcançabilidade**.

É interessante introduzir neste ponto, ainda que informalmente, a noção de tempo nas Redes de Petri, associando para isto uma interpretação à rede da figura 1. Considere que esta figura representa o processamento repetido de um programa paralelo composto de 5 tarefas T1, T2, T3, T4 e T5. A execução de cada tarefa está associada ao disparo das transições t1, t2, t3, t4 e t5, respectivamente. Como no sistema real as tarefas não executam instantaneamente, será associado um tempo, representado por uma variável aleatória à duração do disparo de cada uma das transições, de forma a representar o tempo de execução das respectivas tarefas. O programa inicia a execução através da tarefa T1. Após o seu término (disparo de t1), tem início a execução paralela das tarefas T2 e T3. A tarefa T2, ao terminar, pode ou não requerer a execução da tarefa T4 seguida do reinício de T2. Ao fim de cada nova execução de T2 pode ser necessário ou não executar T4. Se não for necessário executar T4 e T3 estiver concluída, a tarefa final, T5, do programa paralelo, executa.

Conforme será visto mais adiante, para que possamos utilizar as RPs para análise de desempenho é necessário gerar o conjunto de alcançabilidade da rede. A fim de se enunciar o processo de geração de um conjunto de alcançabilidade, estabelecer-se-ão algumas definições básicas. Uma marcação é dita morta quando não habilita nenhuma transição. Um lugar é k-limitado se o número máximo de fichas observável no mesmo, ao longo do "funcionamento" da rede, é igual a k. Uma rede que somente possua lugares k-limitados é denominada k-limitada. Em contraposição ao lugar k-limitado encontra-se o lugar ilimitado, que é caracterizado por possuir uma transição entrada com habilitação permanente, o que acarreta um aumento incessante do número de fichas do lugar.

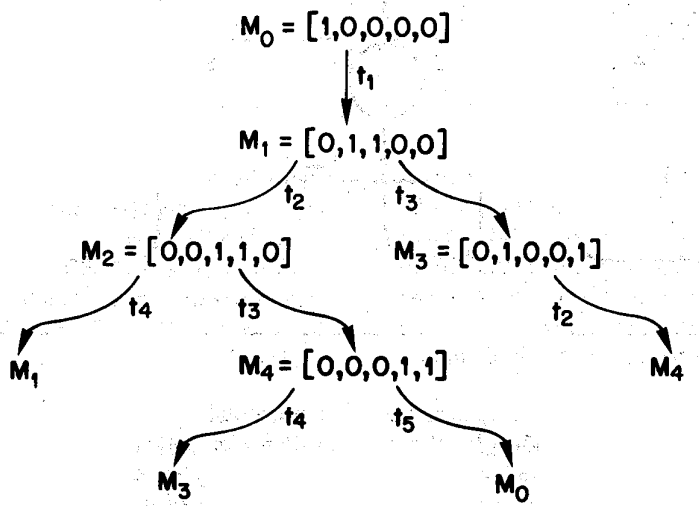
A enumeração do conjunto de alcançabilidade começa pela marcação inicial e a partir desta geram-se as próximas marcações, explorando-se a possibilidade de disparo de cada transição habilitada. A cada marcação gerada, repete-se o processo de geração de novas marcações, desde que a marcação corrente não seja uma marcação morta ou uma marcação duplicata.

Ao se encontrar uma marcação que é caracterizada pelo aumento do número de fichas em um determinado lugar e também pelo aumento do número total de fichas da rede, passa-se a caracterizar o lugar onde houve o aumento por um símbolo especial, que indica que o mesmo admite um número infinito de fichas.

A figura 2 ilustra o conjunto de alcançabilidade da rede da figura 1. Aconselha-se que o leitor siga a árvore de alcançabilidade passo a passo, para que possa ter uma melhor assimilação dos conceitos expostos acima.

A simplicidade do mecanismo de "funcionamento" de uma RP, aliada à flexibilidade de interpretação de sua estrutura fazem desta uma poderosa ferramenta para a modelagem de sistemas. Desta forma, é possível utilizar as RPs como um ferramental para

FIGURA 2
CONJUNTO DE ALCANÇABILIDADE DA RP DA FIGURA 1



a abstração dos níveis de uma arquitetura [1] [4]. Abaixo serão dados dois exemplos que ilustram esta facilidade.

O primeiro consiste na modelagem da CPU de uma arquitetura vetorial [17]. Neste tipo de CPU, normalmente, existem três tipos de unidades de processamento: o processador de instruções (IP), o processador escalar (SP) e o processador vetorial (VP). O primeiro é responsável pela busca, decodificação, preparo e execução de certas instruções, o segundo processador pela execução de instruções escalares e o terceiro por instruções vetoriais. O IP busca e decodifica uma instrução e caso ela seja do tipo escalar encaminha a mesma para o SP. Caso seja vetorial a instrução é enviada para o VP. Na impossibilidade de despacho de uma instrução para um determinado processador, o IP fica à espera da sua liberação.

A figura 3 ilustra a RP que abstrai o comportamento do processador vetorial. Nesta rede, têm-se as seguintes interpretações:

lugar 1 - IP preparando uma instrução;

lugares 2 e 3 - SP e VP respectivamente disponíveis;

lugares 4, 5 e 6 - SP, IP e VP respectivamente ocupados;

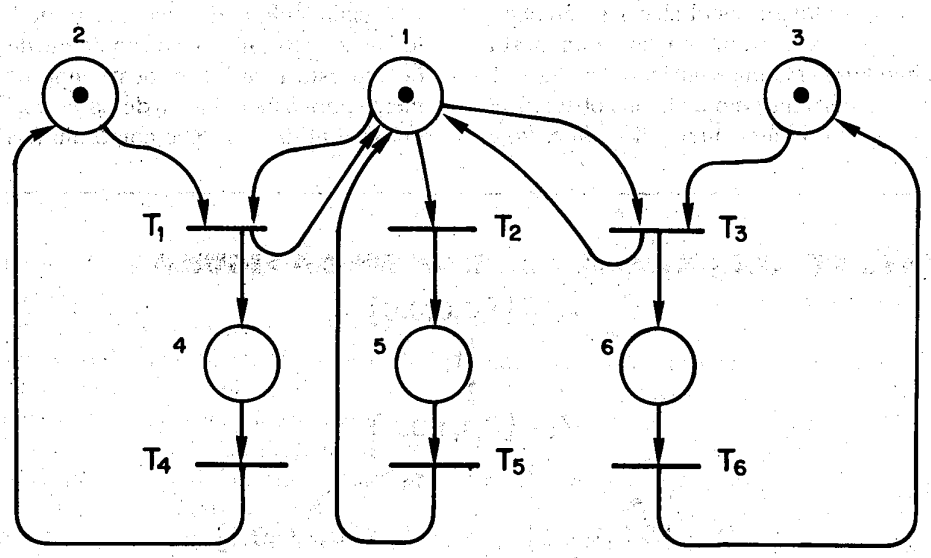
transições t1, t2 e t3 - Busca e preparação das instruções do tipo executadas por SP, IP e VP respectivamente;

transições t4, t5 e t6 - Execução de uma instrução do tipo SP, IP e VP.

O segundo exemplo refere-se a um sistema multiprocessador com cinco processadores, três memórias globais e dois barramentos. Neste sistema, os processadores permanecem ativos durante um certo período, acessando apenas a sua memória local. Findo este período, o processador faz uma referência a uma memória global, porém antes disto deve estar de posse de um barramento [13].

A figura 4 ilustra a dinâmica de compartilhamento das memórias globais. O número de fichas nos lugares p1 e p2 indicam, respectivamente, o número de processadores acessando a sua memória local e o número de barramentos disponíveis. Após o término da execução local (t1), o processador tenta acessar o módulo de memória global desejado. Caso a memória escolhida esteja ocupada, com probabilidade P (t3) = 1/m, onde m é o número de módulos de memória global, a transição t3 dispara se estiver habilitada e o processador espera pelo referido módulo (p5). Caso o módulo desejado esteja livre, com probabili-

FIGURA 3
UM MODELO DE RP PARA UM PROCESSADOR VETORIAL



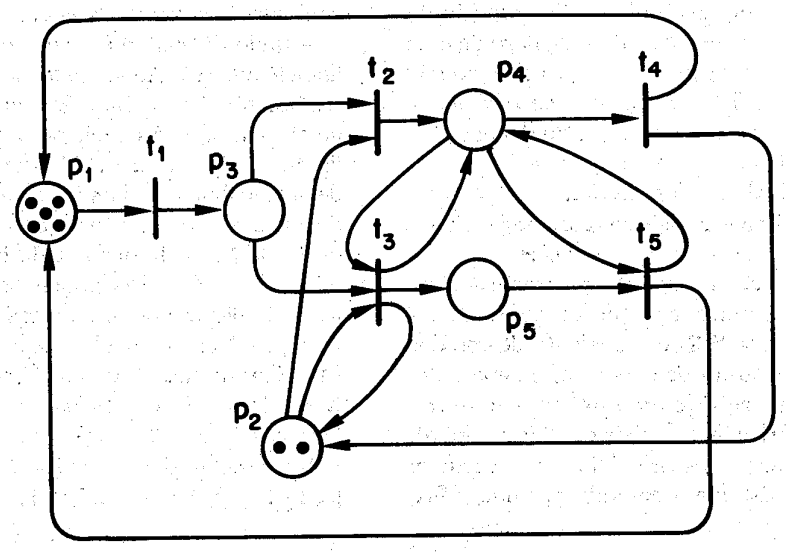
dade P (t2) = 1 - 1/m, a transição t2 dispara se estiver habilitada e o processador acessa a memória global. As transições t5 e t4 representam o término do acesso a módulos de memória onde existem e onde não existem respectivamente requisições pendentes.

3 — O CONCEITO DE TEMPO EM REDES DE PETRI

O conceito de tempo, referência para ordenação lógica de certos eventos, tal como a retransmissão de um pacote, após um certo período de espera por confirmação de recebimento, se confunde com as próprias questões de desempenho, como por exemplo: Qual é o impacto no tempo de execução de um *job* ao se utilizar um processador duas vezes mais veloz?

Assim sendo, a introdução do conceito de tempo em Redes de Petri possibilita a derivação de métricas de desempenho, a partir da especificação funcional de um determinado comportamento. Os primeiros trabalhos de utilização de Redes de Petri com temporização devem-se a Sifakis [31], Shapiro [30], Zubereck [34], Ramamoorthy [28] e Merlin [18].

FIGURA 4
UM MODELO DE RP DE UM MULTIPROCESSADOR



Existem basicamente duas formas de se introduzir o conceito de tempo na estrutura de uma Rede de Petri; associando-o a lugares ou a transições [18].

No caso de se ter lugares temporizados, uma ficha, ao chegar em um determinado lugar, passa por um período de indisponibilidade. Findo este período, a ficha está apta a participar da habilitação de uma ou mais transições [33].

A opção de se atribuir o conceito de tempo a transições engloba duas vertentes. No modelo de Zubereck [34] uma transição, ao se tornar habilitada, inicia o disparo instantaneamente, porém o mesmo se estende por um certo intervalo de tempo. Na estrutura de Merlin [18], existe um retardo entre a habilitação de uma transição e o seu disparo, sendo que este é instantâneo. As classes de redes, que serão estudadas no presente trabalho, fundamentam-se na estrutura de Merlin, sendo que o intervalo de tempo entre a habilitação e o disparo de uma transição segue uma determinada distribuição probabilística [15] [12] [24].

4 — SPN

A possibilidade de integração da estrutura de Redes de Petri com a já estabelecida Teoria dos Processos Estocásticos motivou estudos acerca da definição de uma Rede de Petri onde os intervalos de tempo

fossem definidos como variáveis aleatórias. Desta maneira, poder-se-ia representar o comportamento de sistemas complexos, fazendo-se uso de hipóteses estocásticas.

Os primeiros trabalhos que expressaram o intervalo de temporização como uma variável aleatória devem-se a Molloy [22] e Natkin [23]. Nestes tipos de redes define-se o intervalo de temporização como o período entre a habilitação de uma transição e o seu disparo.

Caso se considere todos os intervalos de tempo como exponencialmente distribuídos têm-se as chamadas *Stochastic Petri Nets* (SPNs) [22] [20].

Em uma SPN, com várias transições habilitadas simultaneamente, dispara primeiro aquela que tem o menor tempo de disparo. Este disparo pode habilitar outras transições. A grande vantagem do estabelecimento de uma variável exponencialmente distribuída consiste na utilização da propriedade de ausência de memória deste tipo de distribuição. No contexto de RPs, este fato implica que ao se ter o disparo de uma transição, todos os intervalos de temporização são reiniciados, isto é, a distribuição do tempo residual é igual a distribuição da temporização do disparo.

Assim sendo, define-se uma SPN como:
SPN = (P, T, A, MO, λ), onde λ = (λ1, λ2, ... , λm) é o conjunto das taxas de disparo associadas às transições. Estas taxas podem ser dependentes das marcações.

Molloy demonstrou em sua tese que existe um isomorfismo entre as marcações do conjunto de alcançabilidade de uma SPN e o conjunto de estados de uma cadeia de Markov de tempo contínuo e espaço de estados discreto (e em particular uma RP k-limitada é isomorfa a uma cadeia de Markov finita).

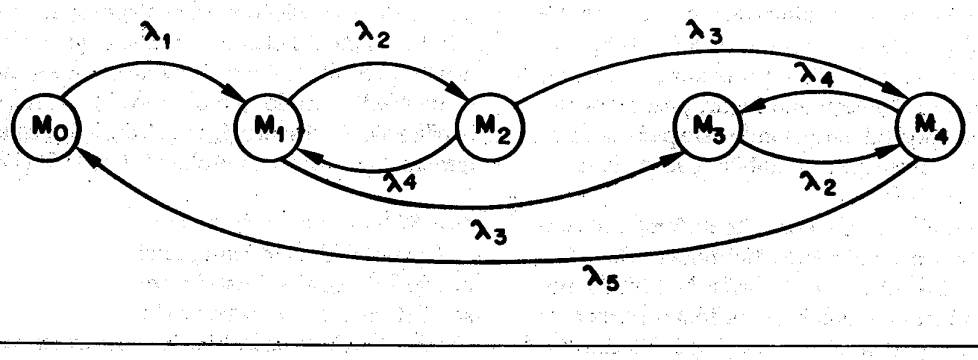
Desta forma, para se obter medidas de desempenho de um certo sistema modelado por uma SPN, considera-se a RP associada, isto é, a RP padrão sem temporização, e enumera-se o seu conjunto de alcançabilidade. Os estados da cadeia de Markov associada são idênticos às marcações da RP. As taxas de transição entre dois estados da cadeia de Markov consistem no somatório das taxas de disparo das transições da RP cujos disparos interligam as duas marcações correspondentes aos estados da cadeia de Markov.

Tendo-se a cadeia de um processo Markoviano, obtém-se a matriz de transição de estados e, caso o processo seja ergódico, calculam-se as probabilidades estacionárias de se encontrar o sistema em um determinado estado. A condição necessária e suficiente para que se possa afirmar que um processo estocástico enumerado a partir de uma SPN seja ergódico consiste no fato da marcação inicial da rede poder ser alcançável a partir de qualquer marcação do conjunto de alcançabilidade.

A título de ilustração, considere a RP representada pela figura 1. Ao se associar a todas as transições temporizações exponencialmente distribuídas, define-se uma SPN. A cadeia de Markov apresentada na figura 5 pode ser deduzida partindo-se do conjunto de alcançabilidade, ilustrado na figura 2. Observa-se pela figura 2, que a marcação M2 pode ser gerada a partir de M1 pelo disparo da transição t2 e que as marcações M1 e M4 podem ser geradas a partir de M2 pelo disparo de t4 e t3, respectivamente. Nota-se na figura 5 que o estado M2 é alcançável a partir de M1, com uma taxa de disparo igual a λ2 e que a partir de M2 pode-se obter os estados M1 e M4 com taxas respectivamente iguais a λ4 e λ3.

Supondo-se que as taxas de disparo são iguais a [22] λ1 = 2, λ2 = 1, λ3 = 1, λ4 = 3 e λ5 = 2, ob-

FIGURA 5
CADEIA DE MARKOV ASSOCIADA À RP DA FIGURA 1



têm-se as seguintes probabilidades estacionárias P [MO] = .1163, P [M1] = .1860, P [M2] = .0465, P [M3] = .5349 e P [M4] = .1163.

A partir das probabilidades acima obtidas, podem ser obtidas medidas de desempenho como exemplificado a seguir. Considere a interpretação da rede da figura 1 dada anteriormente (execução repetida de um programa paralelo). Considere que se deseja obter o tempo médio T de execução do programa paralelo. Observe que o estado MO é o estado no qual o sistema permanece durante a execução da tarefa seqüencial T1. Logo, durante uma fração de tempo P [MO], que é igual a 0.1163, o tempo médio gasto pelo programa é o tempo médio de execução da tarefa T1, que é igual a $1/\lambda 1 = 0.5$. Portanto, o tempo médio total de execução do programa é dado por:

$$T = \frac{1/\lambda 1}{P\,[MO]} = \frac{0.5}{0.1163} = 4.299 \text{ unidades de tempo}$$

Outras medidas de desempenho podem ser obtidas a partir das probabilidades estacionárias tais como: o número médio de fichas em um certo lugar, a taxa média de disparo por unidade de tempo de uma determinada transição, etc. Por exemplo, o número médio de fichas em um lugar i é dado por:

$$\sum_{j} j \cdot \sum_{e \in S(i,j)} P\,[e]$$

onde e é um estado da RP, P [e] é a probabilidade estacionária deste estado, e S (i,j) é o conjunto de estados do conjunto de alcançabilidade da RP com j fichas no lugar i. Assim, no exemplo anterior, o número médio de fichas no lugar 3 é dado por P [M1] + P [M2] = 0.2325.

Molloy mostrou que é possível definir uma distribuição de temporização generalizada, utilizando-se o método dos estágios, desde que esta distribuição possua uma transformada de Laplace racional.

Molloy estabeleceu, também, que quando se tem uma transição com taxa de disparo muito elevada, deve-se colocar todas as taxas em função de uma certa incógnita, digamos x, e no final da resolução, tomar-se o limite quando x tende ao infinito. Este método, porém, tende a apresentar problemas numéricos.

5 — GSPN

Existem situações de modelagem onde os tempos de duração dos eventos ocorridos no sistema podem diferir por ordens de magnitude, como por exemplo o tempo de troca de contexto no compartilhamento da CPU que é consideravelmente menor do que a fatia de tempo que os processos adquirem.

Marsan, Conte e Balbo [16] definiram a estrutura denominada *Generalized Stochastic Petri Net* (GSPN) que difere da SPN pela possibilidade de existirem transições imediatas. Associada a uma transição temporizada está uma distribuição exponencialmente distribuída, assim como nas SPNs. As transições imediatas disparam num intervalo nulo de tempo. Graficamente, representa-se uma transição temporizada como uma barra grossa e uma transição imediata como uma barra fina.

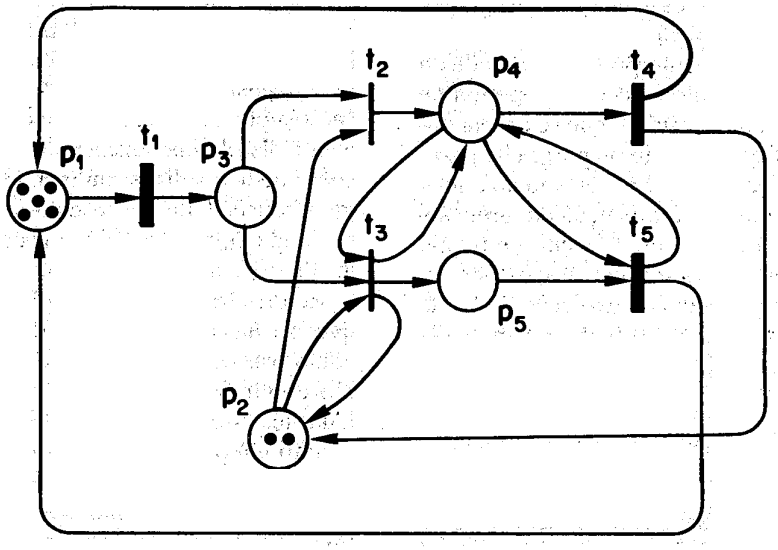
Retornando-se ao exemplo da figura 4, nota-se que as transições t1, t4 e t5 devem ser interpretadas como transições temporizadas, pois a transição t1 representa o fim de acesso à memória local, ou seja, a duração do seu intervalo de temporização indica o quanto um processador gastou fazendo acessos locais e as taxas de disparo de t4 e t5 indicam o quanto se gastou acessando um módulo global de memória. Por outro lado, as transições t2 e t3, que modelam respectivamente a escolha de um módulo de memória livre ou ocupado, não englobam nenhum aspecto temporal ou seja, indicam apenas uma condição lógica, logo devem ser representadas como transições imediatas. A figura 6 ilustra o mesmo sistema modelado na figura 4, utilizando-se, para tal o modelo GSPN.

A existência de dois tipos de transições acarreta numa mudança da semântica do disparo de transições. Caso em uma determinada marcação existam apenas transições temporizadas habilitadas, a regra de disparo permanece idêntica a do modelo SPN. Na situação de se ter transições temporizadas habilitadas e apenas uma transição imediata habilitada, então a transição imediata dispara com probabilidade igual a 1. Quando em uma marcação tem-se mais de uma transição imediata habilitada, então, tem-se que definir uma distribuição de probabilidade que indique a chance de cada uma das transições imediatas disparar. Esta distribuição é chamada de distribuição de chaveamento.

A estrutura GSPN inclui também os chamados arcos inibidores, ou seja, arcos que vão de um lugar para uma transição e que são representados por um círculo em sua extremidade ao invés de uma flecha. A transição na qual chega um arco inibidor só pode ser habilitada caso não haja nenhuma ficha no lugar do qual parte um arco inibidor. A figura 7 ilustra uma transição desabilitada por um arco inibidor.

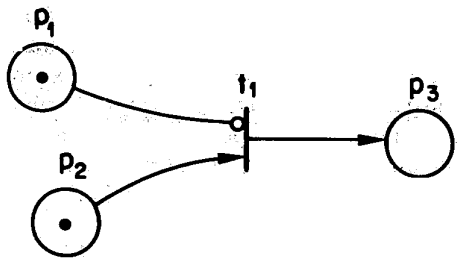
Ao se observar o processo estocástico produzido pela execução de uma GSPN, constata-se a existên-

FIGURA 6
MODELO GSPN CORRESPONDENTE À RP DA FIGURA 4



cia de múltiplas descontinuidades devido às transições imediatas. Chamam-se estados evanescentes aqueles estados nos quais o processo despende um intervalo nulo de tempo e que correspondem a marcações que habilitam transições imediatas. Os estados nos quais o processo despende um intervalo de tempo não nulo são chamados estados tangíveis.

FIGURA 7
EXEMPLO DE ARCO INIBIDOR



É possível, porém, definir uma cadeia de Markov embutida ao processo gerado por uma GSPN, caso sejam feitas algumas hipóteses que conduzam à construção de um processo estacionário de tempo contínuo, com espaço de estado finito e irredutível, tais como: o conjunto de alcançabilidade é finito, as taxas de disparo não dependem do tempo e a marcação inicial pode ser alcançada a partir de qualquer outra marcação [13].

Deste modo, é possível chegar-se às probabilidades estacionárias, elegendo-se um estado tangível como referência e calculando-se o tempo médio de retorno a cada estado tangível.

Desenvolveu-se [13] um método que define uma cadeia de Markov embutida, de forma a se ter apenas estados tangíveis. As probabilidades estacionárias deste espaço reduzido são utilizadas para se calcular as taxas de visita aos estados tangíveis da cadeia de Markov embutida e assim chegar à solução que originariamente se propunha.

Bobbio [7] postulou um método de obtenção da solução acima descrita sem considerar hipóteses de estacionaridade, ou melhor, seu método se aplica também à solução transiente.

A introdução de transições imediatas, apesar do grande poder de modelagem herdado, redunda na maior limitação prática do uso de GSPNs.

Relembrando: para cada conjunto de transições imediatas conflitantes é necessário que se defina uma distribuição denominada distribuição de chaveamento, que atribui probabilidades de disparo a cada uma

das transições. Desta forma, é essencial que se conheçam todas as possíveis marcações que tenham transições imediatas conflitantes, ou seja, para a elaboração correta de um modelo GSPN é fundamental que se conheça *a priori* o seu conjunto de alcançabilidade.

No que tange ao método de solução da cadeia de Markov embutida e reduzida, apesar dos estados evanescentes não alterarem a complexidade computacional da cadeia, a eliminação destes estados pode tornar o método proibitivo. Acrescente-se a este fato a possibilidade de geração de uma grande quantidade de estados evanescentes a partir de um reduzido número de transições imediatas.

Dada a necessidade de especificação das distribuições de chaveamento e a conseqüente inviabilidade de análise de sistemas complexos, buscou-se uma nova definição para as GSPN, na qual as distribuições de chaveamento pudessem ser definidas sem o conhecimento do conjunto de alcançabilidade [14].

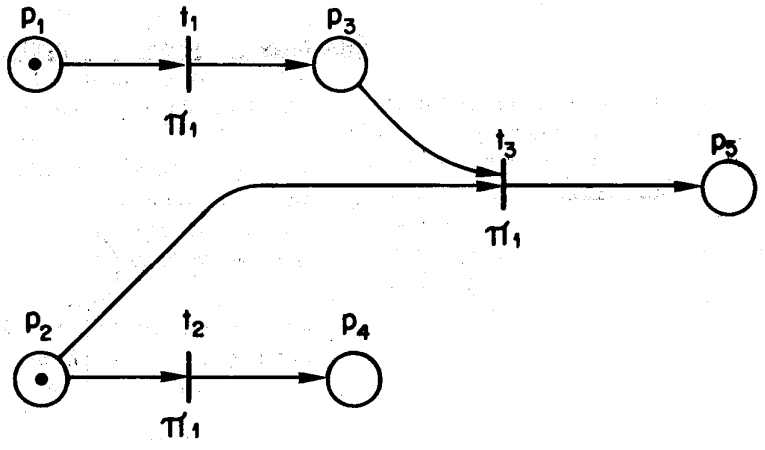
Antes de se definir o novo modelo GSPN, torna-se necessário que se entenda o conceito de confusão. A situação de confusão ocorre quando a determinação de um conflito pode depender de uma seqüência de disparo de transições que não estão em conflito entre si: considere a situação na qual o disparo de uma transição tz que não estava em conflito estrutural com tk habilita uma terceira transição tm que, por sua vez, pertence ao conjunto de conflito estrutural de tk.

A figura 8 ilustra uma situação de confusão. Nesta rede as transições t1 e t2 não estão em conflito. Porém, ao se disparar t1, t3 torna-se habilitada e em conflito com t2.

As novas redes, denominadas GSPN2, são definidas como: GSPN2 = (P, T, F(.), I(.), O(.), H(.), W(.), MO) onde a RP associada, ou seja a 7-upla acima definida excluindo-se W(.), é uma RP livre de confusão, ou seja, é uma rede onde a determinação de situações de conflito pode ser feita através de uma análise pictorial da rede.

As componentes P, T, I(.), O(.), MO são conceitos

FIGURA 8
EXEMPLO DE CONFUSÃO



comuns às definições anteriores, ou seja, são os conjuntos dos lugares e transições, as funções entrada e saída e a marcação inicial.

A função H(.) representa a função de inibição do conjunto de transições.

A função F(.) define a função de prioridade de disparo das transições. Fazendo-se uso desta função, generaliza-se o conceito de dois níveis de prioridade existentes numa GSPN, onde as transições imediatas têm uma prioridade superior às transições temporizadas. Nas GSPN2 as transições temporizadas continuam possuindo o menor nível de prioridade, porém pode-se definir níveis de prioridade de disparo entre as transições imediatas.

A função W(.) associa a cada transição um número real. No caso das transições temporizadas este número explicita a taxa de disparo da transição e no caso das transições imediatas este valor representa um peso que é usado para se calcular a probabilidade de disparo da transição. Estes pesos são determinados, apenas, a partir das características estruturais da rede, excluindo-se a necessidade de enumeração do conjunto de alcançabilidade.

A adoção da característica de livre confusão não impõe um tributo muito grande, no sentido de que a maioria dos modelos de GSPN, apresentados na literatura, possuem tal característica. Por outro lado, a abstração de sistemas cujo funcionamento depende fortemente do conceito de prioridade e a utilização de uma modelagem hierarquizada de um sistema são facilitadas pela introdução de níveis de prioridade para o disparo de transições imediatas.

6 — ESPN

ESPNs (*Extended Stochastic Petri Net*) foram concebidas inicialmente para a análise de sistemas tolerantes a falhas, porém seu grande poder de modelagem engloba a classe de problemas denominada *"performability"* ("performance" + "reliability") [10].

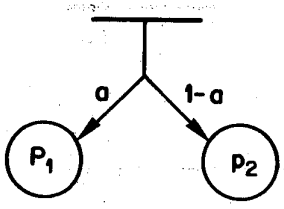
A grande diferença entre ESPN e os modelos estudados anteriormente consiste na flexibilidade dos intervalos de temporização associados às transições poderem seguir uma distribuição de probabilidade arbitrária.

Além dos arcos inibidores, a estrutura de uma ESPN inclui os arcos probabilísticos e os arcos alternativos-contador.

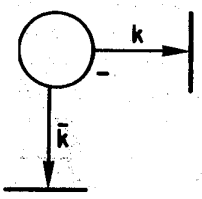
A semântica dos arcos probabilísticos é tal que a cada arco do conjunto de arcos que vão de uma transição a um conjunto de lugares de saída é associada uma probabilidade. Quando ocorre um disparo, apenas um dos lugares de saída recebe uma ficha - aquele escolhido de acordo com a probabilidade associada, figura 9.

O arco alternativo-contador habilita a transição de saída de um certo lugar, caso o número de fichas deste lugar entrada não alcance um determinado valor. Caso este limitante seja alcançado, então a tran-

FIGURA 9
EXEMPLO DE ARCOS PROBABILÍSTICO E ALTERNATIVO-CONTADOR



Probabilístico



Alternativo - Contador

sição ligada ao lugar pelo arco de multiplicidade limite é que se torna habilitada. A transição ligada a um lugar alternativo-contador pode disparar a cada vez que se inclui uma ficha no lugar entrada, porém o disparo desta transição não acarreta numa diminuição do número de fichas do lugar entrada. A figura 9 ilustra um arco alternativo-contador, onde o limitante é igual a k.

Além destas inovações, as regras de disparo de uma ESPN não permitem que haja habilitação simultânea de duas transições imediatas conflitantes, ou seja, duas transições cujo disparo de uma desabilita o disparo da outra.

Em termos de análise, pode-se dizer que no caso de se ter todos os intervalos de temporização exponencialmente distribuídos, a ESPN pode ser estudada pela cadeia de Markov correspondente.

Para que o processo gerado a partir de uma ESPN possa ser considerado semi-Markoviano três condições devem ser satisfeitas [9]. A primeira hipótese diz que uma transição exclusiva, ou seja, uma transição do tipo que para todas as marcações que a habilitam, ela é a única a ser habilitada. Pode ter uma distribuição qualquer. A segunda restrição impõe que o tempo de disparo de transições concorrentes seja exponencialmente distribuído. Denominam-se transições concorrentes aquelas que são habilitadas numa mesma marcação e cujo disparo de uma não desabilita a outra. A terceira condição impõe que o tempo de disparo de transições conflitantes, ou seja, transições habilitadas numa mesma marcação e cujo disparo de uma desabilita a outra, pode seguir uma distribuição qualquer, desde que a distribuição do tempo de disparo das transições que são reabilitadas, em conseqüência do fato de terem sido desabilitadas, não dependa do tempo de disparo residual da habilitação anterior.

É possível, por outro lado, transformar um processo gerado por uma ESPN, cujos tempos de disparo das transições concorrentes não é exponencialmente distribuído em um processo semi-Markoviano através da agregação de estados.

No caso da ESPN não se encaixar em nenhuma das situações anteriormente descritas, então, a análise do processo gerado pela ESPN é feita através de simulação.

7 — CONCLUSÕES

Foram estudadas, ao longo deste trabalho, as propriedades das três principais estruturas de Redes de Petri utilizadas em análise de desempenho.

Como em qualquer problema a ser estudado, existe o compromisso entre a flexibilidade de abstração e a complexidade computacional que se herda ao se adotar um determinado modelo. No caso de Redes de Petri Estocásticas, o espectro de soluções varia desde a resolução via cadeia de Markov até a adoção de simulação, passando-se pelas opções de processos semi-Markovianos ou Markovianos embutidos. Obviamente, a natureza do sistema a ser modelado ditará a ferramenta mais adequada ao problema, porém aconselha-se aos principiantes, sempre que possível, adotarem uma temporização exponencialmente distribuída, posto que a falta de memória da distribuição exponencial simplifica a solução do modelo.

A análise de um sistema de computação utilizando-se Redes de Petri requer a enumeração de todo o espaço de estados. No caso das Redes de Petri Estocásticas, esta restrição se traduz na enumeração de todos os estados do processo estocástico associado, o que em muitas situações pode levar a uma solução computacionalmente inviável.

Por outro lado, os modelos de Redes de Filas são uma abstração de alto nível da cadeia de Markov representativa do sistema, o que, certamente, imprime uma baixa complexidade computacional à resolução do modelo, mas pode implicar restrições quanto à gama de situações passíveis de modelagem através desta classe modelos.

A integração entre Redes de Petri Estocásticas e Redes de Filas parece ser o caminho natural para a adoção em uma metodologia sistemática onde cada uma destas ferramentas contribua com a sua melhor característica (poder de abstração ou baixa complexidade computacional). Ghanta [11] descreve, em sua tese, uma metodologia com esta finalidade, baseando-se na dualidade entre taxa de disparo de uma transição dependente da marcação e taxa de processamento dependente da população em um servidor de fluxo equivalente. Acredita-se que o desenvolvimento de um método eficiente para a integração de ambas ferramentas deva ser pautada em técnicas de análise que não requeiram a enumeração do conjunto de alcançabilidade, tal como em [21] [19].

8 — AGRADECIMENTOS

O trabalho de Nelson Luis S. da Fonseca durante a elaboração deste trabalho foi patrocinado pelo Centro Científico da IBM Brasil. Os autores gostariam de agradecer também ao revisor deste trabalho que deu contribuições significativas para melhorar a qualidade do mesmo.

REFERÊNCIAS BIBLIOGRÁFICAS

1. AGERWALA T. Putting Petri Nets to Work, *IEEE Computer*, Dezembro de 1979.
2. AGRAWAL S.C. Metamodelling - A Study of Approximations in Queueing Models, *The MIT Press*, Cambridge, Massachusetts, 1985.
3. ALMEIDA V.A.F. Approximation Solution Techniques for Queueing Network Models of Concurrent Processing and Other Non-Product Form Problems, *PhD Thesis University of Vanderbilty*, 1987.
4. BAER J.L. Modelling Architectural Features With Petri Nets, *Technical Report 86-06-04*, University of Washington, Junho de 1986.
5. BALBO G., CHIOLA G., FRANCESCHINIS G. and RAET G.M. On the Efficient Construction of the Tangible Reachability Graph of Generalized Stochastic Petri Nets, *Proceeding* of the International Workshop on Petri Nets and Performance Models, IEEE Computer Society Press, August 1987, pp 136-145.
6. BASKETT F., CHANDY K.M. and MUNTZ R.R. Palacios F.G, Open, Closed and Mixed Networks of Queues with Different Classes of Customers, *JACM*, 22,2 1975, pp 248-260.
7. BOBBIO A., CUMANI A. and DEL BELLO R. Reduced Markovian Representation of Stochastic Petri Nets Models, *System Science*, 10,2 1984, pp 5-22.
8. BRUELL STEVEN and BALBO G. *Computational Algorithms for Closed Queueing Networks*, Elsevier North Holland Inc., New York, 1980.
9. DUGAN J.B. Extended Stochastic Petri Nets: Application and Analysis, *PhD Thesis*, Duke University, 1984.
10. DUGAN J.B., CIARDO G. BOBBIO A. and TRIVEDI K. The Design of a Unified Package for the Solution of Stochastic Petri Net Models, *Proceedings* of the International Workshop on Tined Petri Nets, IEEE Computer Society Press, Torino, Julho de 1985, pp 80-87.
11. GHANTA S. On the Integration of Queueing Networks and Generalized Stochastic Petri Nets For The Performance Evaluation of Computer Systems, *PhD Thesis*, University of Minnesota, 1984.
12. HAAS P. J. Markovian Stochastic Petri Nets, RJ 6764, *IBM Research Report*, Almadem Research Center, 1989.
13. MARSAN M.A., BALBO G. and CONTE G. *Performance Models of Multiprocessor Systems*, The Mit Press, Cambridge, Massachusetts, 1986.
14. MARSAN M.A., BALBO G., CHIOLA G. and CONTE G. Generalized Stochastic Petri Nets Revisited : Random Switches and Priorities, *Proceeding* of the International Workshop on Petri Nets and Performance Models, IEEE Computer Society Press, Agosto de 1987, pp 44-53.
15. MARSAN M.A., BALBO G., BOBBIO A., CHIOLA G., CONTE G. and CUMANI A. On Petri Nets With Stochastic Timing, *Proceeding* of the International Workshop on Tined Petri Nets, IEEE Computer Society Press, Torino, Julho de 1985, pp 80-87.
16. MARSAN M. A., CONTE G. and BALBO G. A Class of generalized Stochastic Petri Nets For The Performance Evaluation of Multiprocessor Systems, *ACM Transaction of Computer Systems*, vol. 2,n2, Maio de 1984, pp 93-122.
17. MENASCÉ D.A. and ALMEIDA V.A.F. Analytic Models of Supercomputer Performance in Multiprogramming Environments, *The International Journal of Supercomputer Applications*, The Mit Press Journals, vol. 3, n2, Summer 1989, pp 71-91.
18. MERLIN J.A. and FARBER D.J. Recoverability of Communication Protocols - Implication of a Theorical Study, *IEEE Transaction on Communication*, COM - 24 (9), Setembro de 1987, pp 1036-1043.
19. MOLLOY M.K. Fast Bound For Stochastic Petri Nets, *Proceeding* of the International Workshop on Tined Petri Nets, IEEE Computer Society Press, Torino, Julho de 1985, pp 244-249.
20. MOLLOY M.K. Performance Analysis Using Stochastic Petri Nets, *IEEE Transaction on Computer Systems*, vol. c-3, n9, Setembro de 1982.
21. MOLLOY M.K. Structurally Bounded Stochastics Petri Nets, *Proceeding* of the International Workshop on Petri Nets and Performance Models, IEEE Computer Society Press, Agosto de 1987, pp 156-163.
22. MOLLOY M.K. On the Integration of Delay and Throughput Measures in Distributed Processing Models, *Phd Thesis*, University of California at Los Angeles, 1981.
23. NATKIN S. Les Reseaux de Petri Stochastiques - Theorie, Techniques de Calcul Applications, *These D'Etat*, University of Paris VI, Junho de 1985.
24. PAGNONI. Stochastic Nets and Performance Evaluation, *Proceedings* of Advanced Course on Petri Nets - Part I, Springer Verlag, 1986.
25. PETERSON J. *Petri Net Theory and The Modelling of* Systems, Prentice Hall, Englewood Cliffs, 1981.
26. PETERSON J. Petri Nets, *ACM Computing Surveys*, vol. 9, n3, Setembro de 1977.
27. PETRI C.A. Communication With Automata, *PhD thesis*, Tech Rep RADC-TR-65-3777, Rome Air Development Center, Rome NY, 1966.
28. RAMAMOORTHY C.V., GARY J Ho. Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets, *IEEE Transaction on Software Engineering*, vol. 6, n5, Setembro de 1980.
29. REISIG W. *Petri Net - An Introduction*, Springer Verlag, Berlim, 1982.
30. SHAPIRO S.D. A Stochastic Petri Net With Application to Modelling Occupancy Times for Concurrent Task Systems, *Networks*, vol. 9, 1979, pp 375-379.
31. SIFAKIS J. Use of Petri Nets for Performance Evaluation, *Proceedings* of The Third International Workshop on Modelling and Performance Evaluation of Computer Systems, 1977, pp 27-36.
32. SOUZA E SILVA E. and MUNTZ R.R. Approximative Solutions For A Class of non-Product Form Queueing Network Models, *Performance Evaluation*, vol. 7, (3) Agosto de 1987.
33. WONG C. Y., DILLON T.S. and FORWARD K.E. *Proceedings* of the International Workshop on Tined Petri Nets, IEEE Computer Society Press, Torino, Julho de 1985, pp 244-249.
34. ZUBEREK W.M. Timed Petri Nets and Preliminary Performance Evaluation, *Proceedings* of the 7th Annual Symposium on Computer Architecture, La Baule, 1980.