Using Stochastic Recursive Equations as a Tool for Modeling and Performance Evaluation of Parallel Processing Systems

F. Baccelli, A. Jean-Marie, Z. Liu
INRIA, Centre Sophia Antipolis
2004 route des Lucioles
B.P. 93
06902 Sophia Antipolis, France

Abstract

This report presents the approach of stochastic recursive equations (SRE) for the modeling and performance analysis of parallel systems. The aim of the paper is twofold. Firstly, we show how performance measures of many parallel systems can be adequately described and obtained using stochastic recursive equations. Secondly, we give an assesment on the capability of the SRE approach.

1 Introduction

In this report, we present modeling and performance analysis of several parallel processing systems using the approach based on stochastic recursive equations (SRE). The aim of the paper is twofold. Firstly, we show how performance measures of many parallel systems can be adequately described and obtained using stochastic recursive equations. Secondly, we give an assesment on the capability of the SRE approach.

The use of such equations for performance analysis of parallel systems appeared first in [7, 8, 9]. In this paper, we extend the class of equations studied there to cope with the systems described in [1], which will be referred to as the "Qmips model" in what follows.

The presentation of the paper is organized as follows. In Section 2, we specify our instance of the Qmips model defined in [1]. This instance is taken as the basic problem and is analyzed in detail in Section 3. In Section 4, we establish recursive equations for extensions of this model. In Section 5, we show recursive equations for other models proposed in [1]. Finally, in Section 6, we overview the principle and general techniques of the SRE approach, and evaluate this approach from different aspects.

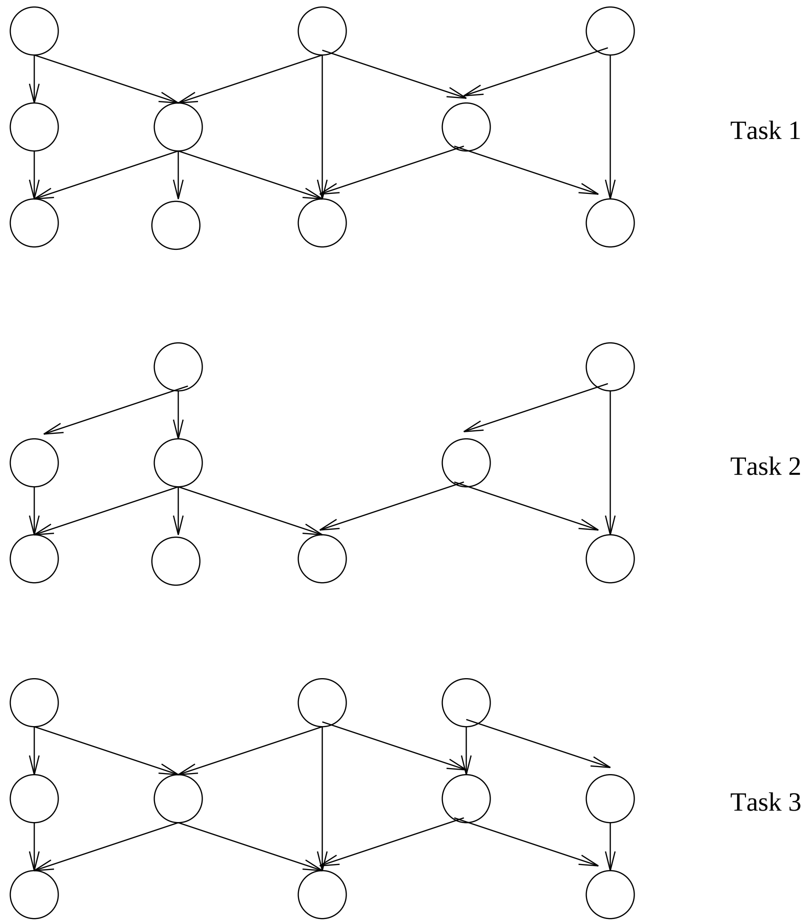

Figure 1: Workload model of the basic system

2 The Basic Problem

The basic model under consideration in this paper is the Qmips model of [1] with variants (d)-(b)-(b)-(b).

Workload Model : (d)

The workload of the system consists of a sequence of tasks, each task is represented by a task graph.

Let $0 = a_1 \leq a_2 \leq \cdots \leq a_n \leq \cdots$ be the arrival times of the tasks, with $\tau_n = a_{n+1} - a_n$ being the interarrival time. Task n is characterized by a directed acyclic graph $G_n = (V_n, E_n)$, referred to as the task graph, where the vertices in V_n represent the subtasks of task n, and the arcs in E_n represent the precedence constraints between the subtasks. An example is illustrated in Figure 1.

The semantics of task graphs are the following:

- vertices: sequential subtasks;
- arcs: precedence constraints (and communications);
- predecessors and successors: "and" logic.

Thus, the precedence graph is such that if $(i, j) \in E_n$, then subtask j can start execution only after subtask i has completed execution.

There is no special requirement regarding statistical assumptions for establishing recursive equations characterizing the dynamics of the system. However, in order to derive stability conditions and other performance measures, we will need assumptions like stationarity and ergodicity.

Multiprocessor Type : (b)

The processors have local memories. In our discussion, the processors can be heterogeneous, instead of identical processors as required in the specification of [1]. Denote by $\mathcal{K} = \{1, \cdots, K\}$ the set of $K \geq 1$ (possibly heterogeneous) processors in the system.

Operating Mode : (b)

- multiprogramming
- static mapping
- scheduling: FIFO (at task level). We also assume that the execution is nonidling.
- no breakdowns

We further assume that the executions (of subtasks) are nonpreemptive.

There is a static mapping $\mathcal{A}_n : V_n \to \mathcal{K}$ that assigns subtasks in V_n to the processors. Let $V_n^k = \{i | i \in V_n,\ \mathcal{A}_n(i) = k\}$ be the set of subtasks assigned to processor $k \in \mathcal{K}$. It is assumed that G_n defines a total order on the subtasks of V_n^k, for each $k \in \mathcal{K}$.

Let b_n^k, e_n^k be the first and the last subtasks, respectively, executed on processor k among those of V_n^k. According to the FIFO rule, subtask b_{n+1}^k is executed only after subtask e_n^k is finished, $k \in \mathcal{K}$. We will assume that the subtasks are executed as soon as possible.

Communication Network : (b)

communication overheads are independent of the status of the system.

Without loss of generality, we assume that the communication between subtasks i and j, $(i, j) \in E_n$, is instantaneous. If this is not the case, we can insert a communication subtask in between i and j. The communication subtask will be executed by a communication processor or channel. The resulting task graph fulfills the above assumption.

The basic problem under consideration is illustrated in Figure 2.

A task is said to be finished if all its subtasks have completed execution. Let C_n be the completion time of task G_n. Denote by $R_n = C_n - a_n$ the response time of task G_n.

For all $j, k \in \mathcal{K}$, let $q_n^{j,k}$ be the indicator function of the existence of a path in G_n from b_n^j to e_n^k. The processing times of subtasks are random variables. Denote by $\sigma_n^{j,k}$ the length of the critical path in G_n from subtask b_n^j to e_n^k, provided $q_n^{j,k} = 1$.

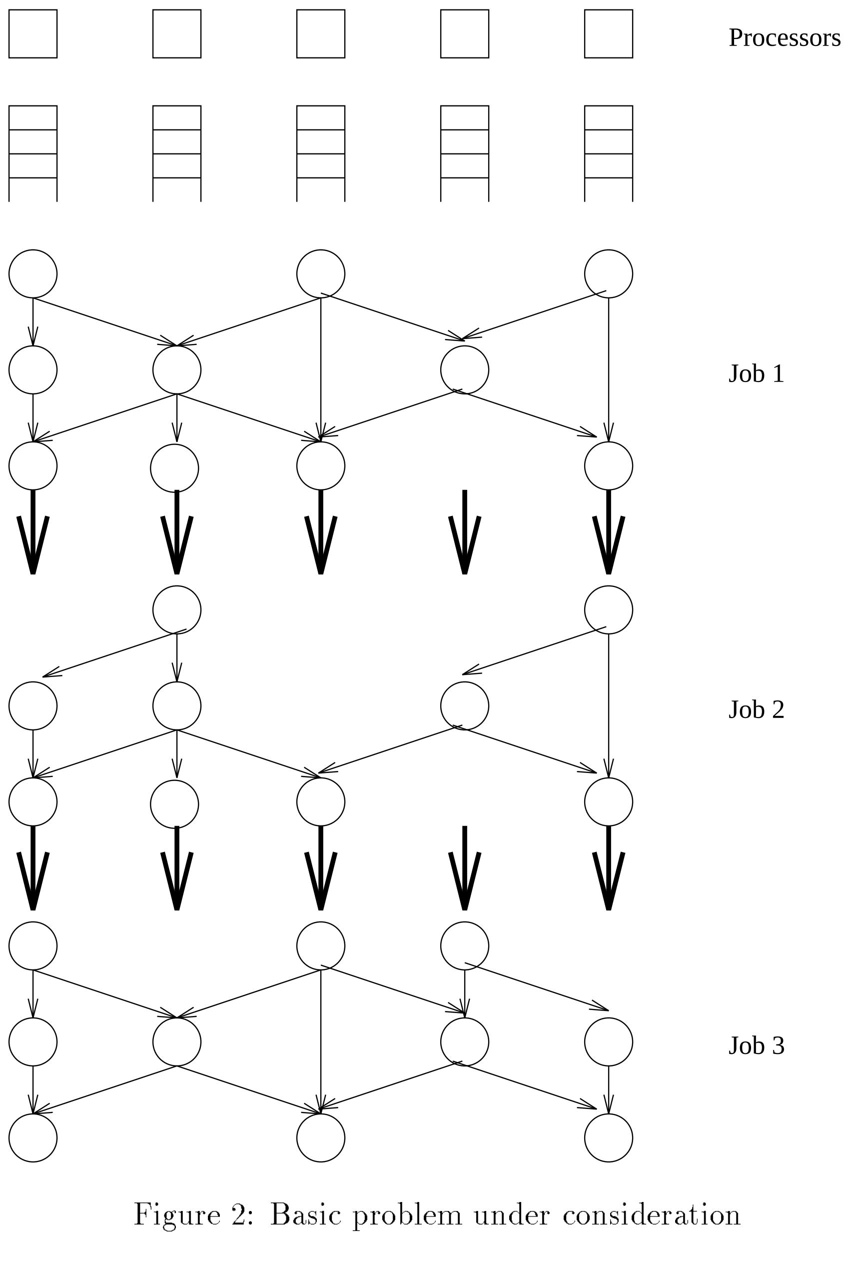

Figure 2: Basic problem under consideration

3 Analysis of the Basic Problem via Stochastic Recursive Equations

The above described parallel processing model is more general than that considered in [8] where the task graphs were identical. However, it turns out that the techniques of [8] still apply.

3.1 The Stochastic Recursive Equations

Let W_n^k: the waiting time of task n on processor k, $k \in \mathcal{K}$. It is checked that these random variables satisfy the following **recursive equations**:

$$\begin{aligned} W_1^k &= 0, \qquad k \in \mathcal{K}; && (1) \\ W_{n+1}^k &= \max_{j \in \mathcal{K}} q_n^{j,k} (W_n^j + \sigma_n^{j,k} - \tau_n)^+, \qquad n \geq 1, \quad k \in \mathcal{K}; && (2) \\ R_n &= \max_{j,k \in \mathcal{K}} q_n^{j,k} \left(W_n^j + \sigma_n^{j,k} \right), \qquad n \geq 1, && (3) \end{aligned}$$

Observe that when $q_n^{j,k}$ are deterministic, the above system can be described by a stochastic marked graph. An example is illustrated in Figure 3, where the executions of the tasks are further constrained by multiprogramming degrees. This last extension of the model will be discussed in the next section.

3.2 Maximum Throughput and Stability

We now assume that
The sequences of random variables $\{\tau_n\}_n$, $\{q_n^{j,k}\}_n$ and $\{\sigma_n^{j,k}\}_n$ are integrable and jointly stationary and ergodic.

Define $D_{m,n}$ to be the length of the critical path in the concatenated graphs $G_m, \cdots, G_n$:

$$D_{m,n} = \max_{\substack{j_m, j_{m+1}, \cdots, j_n, j_{n+1} \in \mathcal{K} \\ q_l^{j_l, j_{l+1}} = 1,\ m \leq l \leq n}} \sum_{l=1}^{n} \sigma_l^{j_l, j_{l+1}}.$$

It can be shown that the process $\{D_{1,n}\}_n$ is subadditive

$$D_{m,n} \leq D_{m,l} + D_{l,n}, \qquad m < l < n.$$

Thus, appealing to Kingman's Theorem on sub-additive ergodic processes entails that there exists a constant μ such that

$$\mu = \inf_{n \geq 1} \frac{E[D_n]}{n} = \lim_{n \to \infty} \frac{E[D_n]}{n} = \lim_{n \to \infty} \frac{D_n}{n}, \qquad a.s. \tag{4}$$

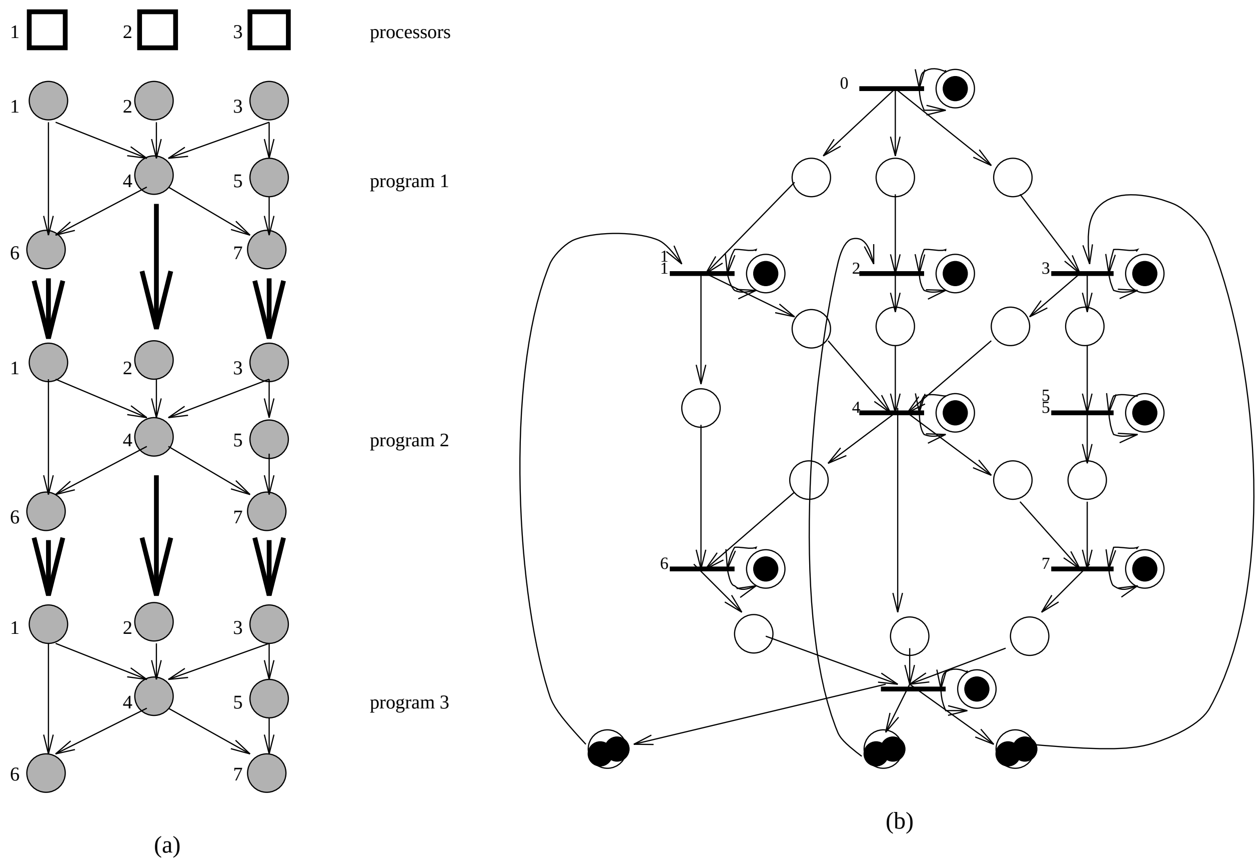

Figure 3: Marked graph representation of the system.

Note that $1/\mu$ is the **maximum throughput** of the system.

The computation of constant μ can be carried out using

simulation : due to the almost sure convergence in (4), the computation can be done by simulating a single sample path [3];

bounds : upper and lower bounds can be obtained by stochastic ordering (that we will present later in this section) [8, 9, 11];

large deviation : an upper bound can be derived by large deviation techniques [5, 6].

In order to derive the stability condition, we consider the **Loynes schema**:

$$\begin{aligned} M_1^k &= 0, \qquad k \in \mathcal{K}; & (5)\\ M_{n+1}^k \circ \theta &= \max_{j \in \mathcal{K}} q^{j,k}(M_n^j + \sigma^{j,k} - \tau)^+, \qquad n \geq 1, \quad k \in \mathcal{K}, & (6) \end{aligned}$$

where θ is the shift operator of the stationary and ergodic sequences $\{\tau_n\}_n$, $\{q_n^{j,k}\}_n$ and $\{\sigma_n^{j,k}\}_n$.

The variables M_n^k have the following properties:

- Increasingness: $M_{n+1}^k \geq M_n^k$.

- Equivalence in law: $M_n^k \circ \theta^n = W_n^k$.

These properties allow us to establish the **stability condition** of the system (cf. [10]):

(i) If $\mu < E[\tau]$, then the random vectors $\vec{W}_n$ converge weakly to a finite random vector $\vec{W}_\infty$ when n tends to ∞;

(ii) If $\mu > E[\tau]$, then the random vectors $\vec{W}_n$ diverge almost surely to ∞ when n tends to ∞.

Remarks:

- The same stability condition holds for response times R_n;
- The same stability condition holds for non-zero initial workloads of the processors;
- Under $\mu < E[\tau]$, it can further be proved that the vector $\vec{W}_n$ converges in coupling, or in total variation.

3.3 Functional Equations

The remainder of this section focuses on the analysis of the transient as well as stationary characteristics of the waiting and response times. In this subsection we derive functional equations for the transient and stationary joint distribution functions of the waiting and response times. In the next subsection we develop upper and lower bounds for these performance measures.

We will assume that the three sequences $\{\tau_n\}_{n=0}^\infty$, $\{\sigma_n^{j,k},\ 1 \le k \le K\}_{n=0}^\infty$, $\{q_n^{j,k},\ 1 \le j,k \le K\}_{n=0}^\infty$ are mutually independent. The random vectors/variables in each of these sequences are independent and identically distributed.

Let $\Lambda(y_{1,1}, \cdots, y_{j,k}, \cdots, y_{K,K})$ be the joint distribution function in $I\!R^{K^2}$ of the random vector $(\sigma^{j,k},\ 1 \le k \le K)$, where $\sigma^{j,k}$ is the generic version of $\sigma_n^{j,k}$. Let $T(z)$ be the distribution function of τ. For sake of easy notation, it will be assumed that Λ is absolutely continuous with respect to the Lebesgue measure. Let $\Phi = \{\vec{a} \in \{0,1\}^{K^2} \mid \forall k \in \mathcal{K}, a^{k,k} = 1\}$. For all $\vec{a} \in \Phi$, let $Q(\vec{a})$ be the probability that $\vec{q} = \vec{a}$. Denote by $U(t_1, \cdots, t_K)$ the step function:

$$U(t_1, \cdots, t_K) = \begin{cases} 0, & \text{if } \exists j:\ t_j < 0; \\ 1, & \text{if } \forall j:\ t_j \ge 0, \end{cases}$$

For all $n \ge 1$, the distribution function F_n of random vector $\vec{W}_n$ satisfies the following **integral equations**:

$$F_1(x_1, \cdots, x_K) = U(x_1, \cdots, x_K) \tag{7}$$

$$F_{n+1}(x_1, \cdots, x_K) = \sum_{\vec{a} \in \Phi} Q(\vec{a}) \cdot U(x_1, \cdots, x_K) \cdot \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} F_n(v_1, \cdots, v_K) d\Lambda(\vec{y}) \right) dT(z) \tag{8}$$

where

$$v_j = \min_{\{k \ | \ a^{j,k}=1\}} (x_k - y_{j,k} + z)$$

The distribution function G_n of the response time R_n is determined by

$$G_n(x) = \sum_{\vec{a} \in \Phi} Q(\vec{a}) \cdot U(x, \cdots, x) \cdot \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} F_n(s_1, \cdots, s_K) d\Lambda(\vec{y})$$

where

$$s_j = \min_{\{k \ | \ a^{j,k}=1\}} (x - y_{j,k})$$

The stationary waiting times $\vec{W}_\infty$ and the stationary response time have the distribution functions F_∞ and G_∞, respectively:

$$\begin{aligned} F_\infty(x_1, \cdots, x_K) &= \sum_{\vec{a} \in \Phi} Q(\vec{a}) \cdot U(x_1, \cdots, x_K) \\ &\quad \cdot \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} F_\infty(v_1, \cdots, v_K) d\Lambda(\vec{y}) \right) dT(z) \quad (9) \\ G_\infty(x) &= \sum_{\vec{a} \in \Phi} Q(\vec{a}) \cdot U(x, \cdots, x) \cdot \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} F_\infty(s_1, \cdots, s_K) d\Lambda(\vec{y}) \quad (10) \end{aligned}$$

3.4 Bounds

Solutions to the above integral equations seem to be difficult to obtain. We propose some upper and lower bounds based on stochastic comparison techniques. We will use $\leq_{st}$ and $\leq_{icx}$ to denote the stochastic ordering and the increasing convex ordering, respectively (see [15]). These stochastic orders are defined as follows:

- $\leq_{st}$: (strong) stochastic ordering. $X \leq_{st} Y$ iff

  $$\forall f \text{ increasing } : E[f(X)] \leq E[f(Y)],$$

  provided the expectations exist.

- $\leq_{icx}$: increasing convex ordering. $X \leq_{icx} Y$ iff

  $$\forall f \text{ increasing and convex } : E[f(X)] \leq E[f(Y)],$$

  provided the expectations exist.

We will also use the notion of association: $X_1, \cdots, X_n$ are associated if for all pairs of increasing functions h, g:

$$\mathrm{cov}[h(X_1, \cdots, X_n), g(X_1, \cdots, X_n)] \geq 0.$$

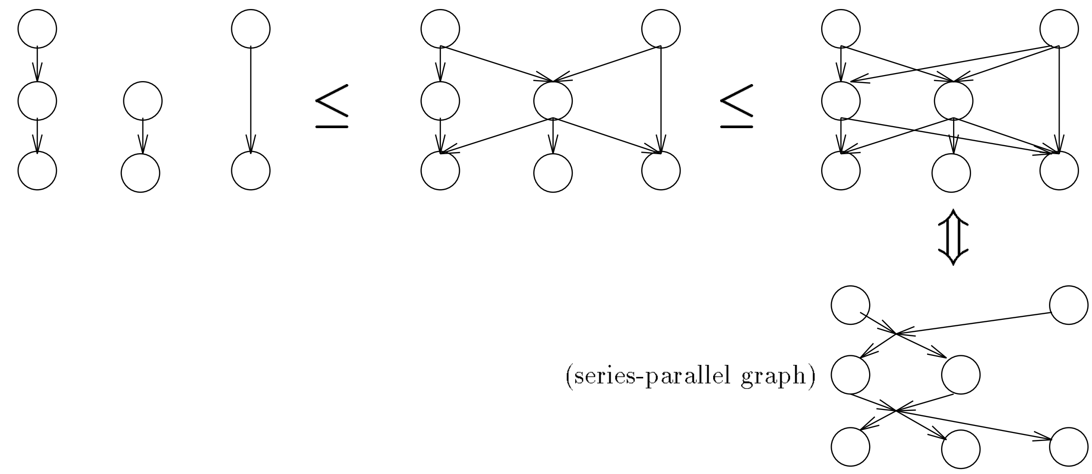

Figure 4: Using deterministic precedence constraints to bound random precedence constraints.

Bounds from precedence constraints Let $\{\bar{q}_n^{j,k},\ 1 \leq j,k \leq K\}_n$ be the sequence of precedence relations satisfying

$$(q^{1,1}, \cdots, q^{K,K}) \leq_{st} (\bar{q}^{1,1}, \cdots, \bar{q}^{K,K}).$$

Let $\overline{W}_n^k$ and $\overline{R}_n$ be the waiting time and response time variables obtained with the new precedence patterns. Appealing to Strassen's Theorem [16] entails that for all $n \geq 1$, we have

$$\left(W_n^1, \cdots, W_n^K\right) \leq_{st} \left(\overline{W}_n^1, \cdots, \overline{W}_n^K\right) \tag{11}$$

$$R_n \leq_{st} \overline{R}_n \tag{12}$$

Remarks:

- The bounds obtained in this way are simple but rough.
- One can use deterministic precedence constraints to bound random precedence constraints. An example is given in Figure 4.

Bounds from timing Let $\{\tilde{\tau}_n\}_{n=1}^{\infty}$, $\{\tilde{\sigma}_n^{j,k}, 1 \leq j,k \leq K\}_{n=1}^{\infty}$, and $\{\tilde{q}_n^{j,k}, 1 \leq j,k \leq K\}_{n=1}^{\infty}$ be the sequences of interarrival times, service requirements and precedence relations satisfying the ordering relations

$$\begin{aligned} -\tilde{\tau} &\leq_{icx} -\tau \\ (\tilde{\sigma}^{1,1}, \cdots, \tilde{\sigma}^{K,K}) &\leq_{icx} (\sigma^{1,1}, \cdots, \sigma^{K,K}) \end{aligned}$$

Let $\widetilde{W}_n^k$ and $\widetilde{R}_n$ be the waiting time and response time variables obtained with the new arrival and service patterns. Applying Strassens's Theorem for the multivariate $\leq_{icx}$

ordering (see [14]), we get

$$\left(\widetilde{W}_n^1, \cdots, \widetilde{W}_n^K\right) \quad \leq_{icx} \quad \left(W_n^1, \cdots, W_n^K\right) \tag{13}$$

$$\widetilde{R}_n \quad \leq_{icx} \quad R_n \tag{14}$$

Using the monotone convergence theorem and the Loynes schema, the transient bounds stated above are readily extended to the steady state provided it exists.

$$\begin{array}{ll} \left(W_\infty^1, \cdots, W_\infty^K\right) \leq_{st} \left(\overline{W}_\infty^1, \cdots, \overline{W}_\infty^K\right), & R_\infty \leq_{st} \overline{R}_\infty \\ \left(W_\infty^1, \cdots, W_\infty^K\right) \leq_{st} \left(\widehat{W}_\infty^1, \cdots, \widehat{W}_\infty^K\right), & R_\infty \leq_{st} \widehat{R}_\infty \\ \left(W_\infty^1, \cdots, W_\infty^K\right) \geq_{icx} \left(\widetilde{W}_\infty^1, \cdots, \widetilde{W}_\infty^K\right), & R_\infty \geq_{icx} \widetilde{R}_\infty \end{array} \tag{15}$$

As an example, we consider the case of deterministic precedence constraints $\tilde{q}_n^{j,k} = q^{j,k}$. Take $\tilde{\tau} = E[\tau]$, $\tilde{\sigma} = E[\sigma]$. Then, due to a result of [8],

$$W_\infty^k \geq_{icx} \max_{1 \leq m \leq N} \left(\max_{h_0, h_1, \cdots, h_{K-1}, h_m = k} \prod_{i=1}^{m} q^{h_{i-1}, h_i} \sum_{i=1}^{m} E[\sigma^{h_{i-1}, h_i}] - mE[\tau] \right)^+, \tag{16}$$

where

$$N = \left\lfloor \frac{\nu + K \cdot \max_{1 \leq u,v \leq K} L^{u,v}}{KE[\tau] - \nu} + 1 \right\rfloor \cdot K,$$

and $L^{j,k}$ is the length of the critical path from processor j to processor k in the K concatenated graphs:

$$L^{j,k} = \max_{j = h_0, h_1, \cdots, h_{K-1}, h_K = k} \left[\prod_{i=1}^{K} q^{h_{i-1}, h_i} \sum_{i=1}^{K} E\sigma^{h_{i-1}, h_i} \right],$$

the constant ν is the "cycle time"

$$\nu = \max_{1 \leq d \leq K} \frac{1}{d} \left\{ \max_{h_1, \cdots, h_d} \left(L^{h_1, h_2} + \cdots + L^{h_{d-1}, h_d} + L^{h_d, h_1} \right) \right\}.$$

Upper Bound by association Let $\Lambda^{j,k}$ and T^- denote the marginal distribution functions of $\sigma_n^{j,k}$ and $-\tau_n$.

Define the sequence $\widehat{W}_n^k$, $(1 \leq k \leq K,\ n \geq 1)$, of distribution functions by the following recursion:

$$\begin{aligned} \widehat{F}_0^k &= U \\ \widehat{F}_{n+1}^k &= \sum_{\vec{a} \in \Phi} Q_n(\vec{a}) \cdot U \cdot \prod_{\{j \ \mid \ a^{j,k} = 1\}} \left(\widehat{F}_n^j * \Lambda^{j,k} * T^-\right) \end{aligned}$$

and the distribution function $\widehat{G}_n$ by the relation

$$\widehat{G}_n = \prod_{1 \leq j,k \leq K} \left(\widehat{F}_n^j * \Lambda^{j,k}\right)$$

where U denotes the step function:

$$U(t) = \begin{cases} 0, & \text{if } t < 0; \\ 1, & \text{otherwise.} \end{cases}$$

Let $\widehat{W}_n^k$ and $\widehat{R}_n$ be the random variables with distribution functions $\widehat{F}_n^k$ and $\widehat{G}_n$, respectively.

Then, using the notion of associated random variables, it can be shown that for all $n \geq 1$, $1 \leq k \leq K$, we have

$$W_n^k \leq_{st} \widehat{W}_n^k \tag{17}$$

$$R_n \leq_{st} \widehat{R}_n \tag{18}$$

Note that under the weak convergence assumption, the above upper bound remains true in the stationary regime.

4 Extensions and Restrictions

In this section, we describe the extensions and also the restrictions of the stochastic recursive equations in the analysis of the Qmips model of [1].

4.1 General Discussions

Multiprocessor Type : (c) shared memory.

Recursive equations can still be found (see below). However, they have a more complicated form.

Operating Mode :

- (b) scheduling according to processor sharing:
  In this case, it is difficult to obtain recursive equations. Below, we will show implicit (but constructive) equations on state variables of such systems.
- (b) scheduling according to priorities:
  The extension to this case is very easy. It suffices to renumber the customers in accordance to the priorities so that the same type of equations are valid.
- (c) fault management (breakdown and repair):
  - operation dependent failures
  - operation independent failures

  The extensions to these cases are relatively simple, and will be discussed later in this section.

Communication Network : (c) explicit models

- crossbar.

  This extension is easy: it suffices to associate a server to each pair of processors.
- bus, ring, multistage network.

  This is similar to the shared memory case. We can obtain some (relatively complicated) recursive equations.
- grid, hypercube.

  The extension to these cases are not impossible. However, the approach by recursive equations is probably not an adequate tool for the aim of performance analysis.

4.2 Shared Memory

For sake of simplicity, we consider a set of parallel processors, each of which executes a sequence of tasks (having a single subtask). We assume that each task reads data from the shared memory at the beginning of execution and writes data to the shared memory at the end of execution. We assume that read accesses have no waiting cost and that its duration is included in the task execution time. The general case can be analyzed in an analogous way.

Let there be K senders. We define the following variables, where $n \geq 1$, $1 \leq i \leq K$.

Input data :

- a_n^i: the n-th arrival time at processor i;
- σ_n^i: the n-th service time of processor i;
- α_n^i: the duration of the n-th write access to the shared memory of processor i.

State variables :

- X_n: time epoch when the n-th write access to the shared memory is requested by one of the processors;
- Y_n: time epoch when the n-th write access to the shared memory is completed;
- A_n^i: time epoch when n-th write access of processor i is requested;
- χ_n: the identity of the processor which requests the n-th write access to the shared memory;
- ν_n^i: the number of the first write access requested by processor i after time X_n.

It can be checked that the following recursive equations hold:

$$
\begin{aligned}
\chi_1 &= 0; && (19)\\
\nu_1^i &= 1, \qquad 1 \leq i \leq K; && (20)\\
A_1^i &= a_1^i, \qquad 1 \leq i \leq K; && (21)\\
X_{n+1} &= \min_{1 \leq i \leq K} \left\{ A^i_{\nu_n^i + \mathbf{1}(\chi_n = i)} \right\}, \qquad n \geq 1; && (22)
\end{aligned}
$$

$$
\begin{aligned}
\chi_{n+1} &= \arg\min_{1\le i\le K}\left\{A^i_{\nu^i_n+\mathbf{1}(\chi_n=i)}\right\}, \qquad n\ge 1; && (23)\\
\nu^i_{n+1} &= \nu^i_n+\mathbf{1}(\chi_n=i), \qquad 1\le i\le K; \quad n\ge 1; && (24)\\
Y_{n+1} &= \max\{Y_n, X_{n+1}\}+\alpha^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}}, \qquad n\ge 1; && (25)\\
A^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}+1} &= \max\left\{a^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}+1},\ Y_{n+1}+\tau^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}}\right\}, \qquad 1\le i\le K, \quad n\ge 1. && (26)
\end{aligned}
$$

4.3 Fault Tolerance

We consider two types of failures: operation dependent failures and operation independent failures.

- For operation dependent failures, when the processing times of the subtasks are i.i.d., and the time to failure is exponentially distributed, the total processing times of the subtasks are still i.i.d., so that the initial recursive equations dont change.
- For other cases, consider a single processor executing a sequence of tasks (having a single subtask). We have similar forms of recursive equations in general.

  Let X_n be the n-th service starting time, δ_n the duration of the n-th failure.

  - Operation dependent failures: let β_n be the number of tasks having successfully completed in between the $n-1$-st and n-th failure.

    $$X_{n+1}=\max(a_{n+1},\ X_n+\sigma_n,\ X_{\gamma(n)}+\sigma_{\gamma(n)}+\delta_{\mu(n)})$$

    where

    $$\mu(n)=\sup\{j|\sum_{k=1}^{j}\beta_k<n\}, \qquad \gamma(n)=\sum_{k=1}^{\mu(n)}\beta_k.$$

  - Operation independent failures: let α_n be the duration of n-th on period.

    $$X_{n+1}=\max(V(a_{n+1}),Y_n),$$

    where $V(t)$ is the first time epoch when the server is available after time t

    $$V(t)=\inf_{k\ge 1}\left\{\alpha_k+\sum_{j=1}^{k-1}\alpha_j+\delta_j\ge t\right\},$$

    and Y_n is the n-th service completion time

    $$Y_n=X_n+\sigma_n+\sum_{j=\nu(n)}^{\mu(n)}\delta_j,$$

    where

    $$
    \begin{aligned}
    \nu(n) &= \inf\left\{k\ge 1 \mid \alpha_1+\textstyle\sum_{j=2}^{k}\alpha_j+\delta_j\ge X_n\right\},\\
    \mu(n) &= \inf\left\{k\ge 1 \mid \textstyle\sum_{j=\nu(n)}^{k}\alpha_j\ge\sigma_n\right\}.
    \end{aligned}
    $$

4.4 Finite degree of multiprogramming

Apart from the above extensions, we can further use the same type of equation to describe the systems with finite degree of multiprogramming. That is, at most d tasks are allowed to execute simultaneously in the system. See equations (27, 28, 29) below.

$$\begin{aligned} W_1^k &= 0, && (27) \\ W_{n+1}^k &= \max\left\{ (R_{n+1-d} - \sum_{i=n+1-d}^{n} \tau_i), \;\; \max_{j \in \mathcal{K}} q_n^{j,k} (W_n^j + \sigma_n^{j,k} - \tau_n)^+ \right\}, && (28) \\ R_n &= \max_{j,k \in \mathcal{K}} q_n^{j,k} \left(W_n^j + \sigma_n^{j,k} \right). && (29) \end{aligned}$$

In a similar way, we can establish simple recursive equation for the case that the multiprogramming degrees are associated with the processors.

4.5 Processor Sharing

Consider the execution of N tasks on K shared processors (such as coprocessors in a single machine). These tasks belong to a single task graph G. The case of execution of multiple task graphs can be analyzed in a similar way.

For $1 \leq i \leq N$, denote by σ_i, B_i, C_i and $p(i)$ the service time, execution beginning time, the execution completion time, and the set of predecessors of task i. For binary variables v_i, $1 \leq i \leq N$, function $A(v_i,\ 1 \leq i \leq N)$ refers to the number of tasks having no predecessor if all tasks i for which $v_i = 1$ are removed from the task graph G. Note that the values of function $A(v_i,\ 1 \leq i \leq N)$ can be obtained by static analysis of the structure of task graph G.

Let $h(t)$ (resp. $v(t)$) be the number of tasks running in parallel (resp. the rate at which the tasks are being run) at time t. Then, it is easy to check that the state variables verify the following implicit but constructive equations:

$$\begin{aligned} h(t) &= A(\mathbf{1}(C_i \leq t),\ 1 \leq i \leq N), \qquad t \geq 0; && (30) \\ v(t) &= \frac{K \wedge h(t)}{h(t)}, \qquad t \geq 0; && (31) \\ B_i &= \max_{j \in p(i)} C_j, \qquad 1 \leq i \leq N; && (32) \\ C_i &= \inf\{t : \int_{B_i}^{t} v(t) dt \geq \sigma_i\}, && (33) \end{aligned}$$

where, by convention, the maximum over an empty set is zero.

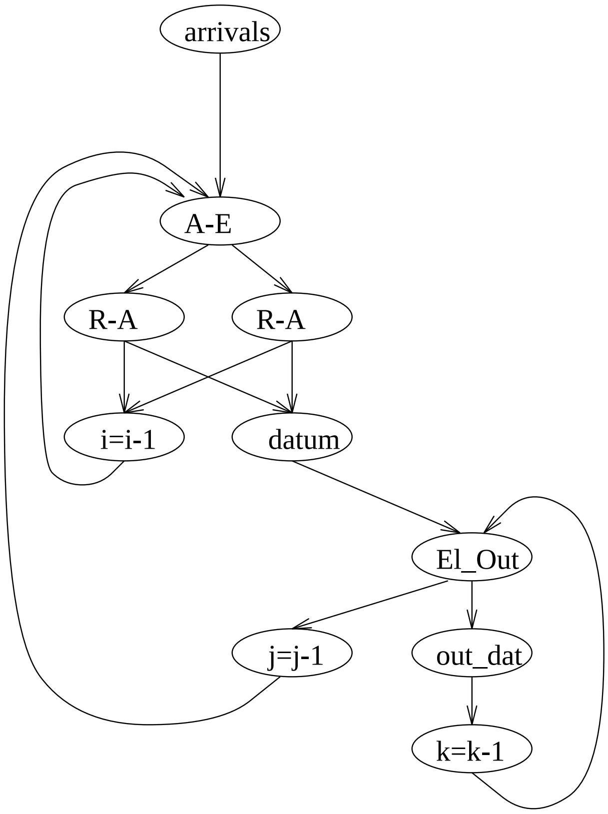

Figure 5: Flow chart representation of the UTO3 example.

5 Other Examples

5.1 UTO Example 3

Consider the the third problem provided by the UTO group. According to the Poisson assumption of the arrival processes of signals, the superposition of these arrival processes is still a Poisson process with rate $\lambda = \lambda_1 + \lambda_2 + \lambda_3$. The class of the n-th signal to arrive is represented by the random variable $u_n \in \{1, 2, 3\}$. The random variables u_n are i.i.d. with $P\{u_n = i\} = \lambda_i/\lambda$, $i = 1, 2, 3$.

A simple analysis of the structure of the Occam program allows us to conclude that the problem can in fact be described by a marked graph. In order to help the understanding, we provide in Figure 5 a flow chart representation (of the loop of 20 times), where A-E (resp. R-A) corresponds to the communications between processes Acquire and Elaborate (resp. Reads and Asks). Note that we assumed that the communications are of rendezvous type, except that signals arrive asynchronously. The communication delays are random variables with distributions depending on the type of signals.

It is now clear that this system can be described and analyzed using the $(\max, +)$ recursive equations typical to marked graphs.

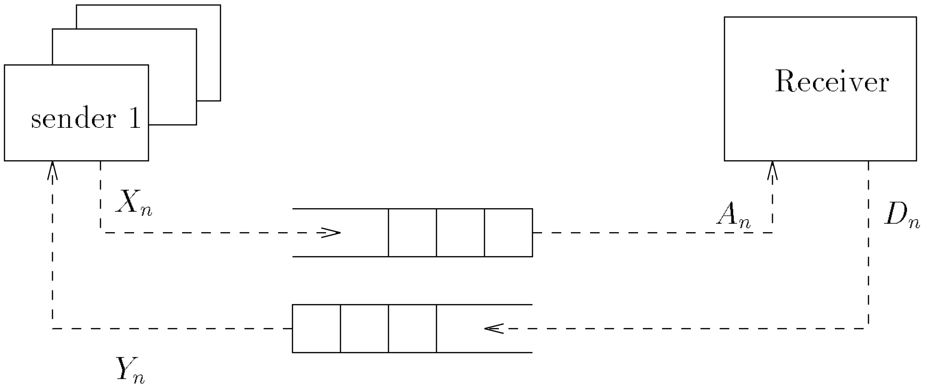

Figure 6: Send-and-Wait Protocol.

5.2 UER Example

Consider now the problem of send-and-wait protocol provided by the UER group. The model is depicted in Figure 6.

Let there be K senders. We define the following variables, where $n \geq 1$, $1 \leq i \leq K$.

Input data :

- τ_n^i: the n-th set-up time of sender i;
- σ_n^i: the n-th service time of the receiver for type i messages;
- α_n^i: the n-th transmission time of type i messages;
- β_n^i: the n-th acknowledgement time of type i messages.

State variables :

- X_n: time epoch when the n-th message is sent out by one of the senders;
- A_n: time epoch when the n-th message arrives at the receiver;
- D_n: time epoch when the n-th acknowledgement message is sent out;
- Y_n: time epoch when the n-th message in the system is acknowledged;
- S_n^i: time epoch when n-th message is sent out by sender i;
- χ_n: the identity of the sender which sends the n-th message in the system;
- ν_n^i: the number of the first message sent out by sender i after time X_n.

It can be checked that the following recursive equations hold:

$$\begin{aligned}
\chi_1 &= 0; && (34)\\
\nu_1^i &= 1, \qquad 1 \leq i \leq K; && (35)\\
S_1^i &= \tau_0^i, \qquad 1 \leq i \leq K; && (36)\\
X_{n+1} &= \min_{1\leq i\leq K} \left\{ S^i_{\nu_n^i + \mathbf{1}(\chi_n = i)} \right\}, \qquad n \geq 1; && (37)
\end{aligned}$$

$$\chi_{n+1} = \arg\min_{1\le i\le K}\left\{S^i_{\nu^i_n+\mathbf{1}(\chi_n=i)}\right\}, \qquad n\ge 1; \tag{38}$$

$$\nu^i_{n+1} = \nu^i_n + \mathbf{1}(\chi_n = i), \qquad 1\le i\le K; \quad n\ge 1; \tag{39}$$

$$A_{n+1} = \max\{A_n, X_{n+1}\} + \alpha^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}}, \qquad n\ge 1; \tag{40}$$

$$D_{n+1} = \max\{D_n, A_{n+1}\} + \sigma^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}}, \qquad n\ge 1; \tag{41}$$

$$Y_{n+1} = \max\{Y_n, D_{n+1}\} + \beta^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}}, \qquad n\ge 1; \tag{42}$$

$$S^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}+1} = Y_{n+1} + \tau^{\chi_{n+1}}_{\nu^{\chi_{n+1}}_{n+1}}, \qquad 1\le i\le K, \quad n\ge 1. \tag{43}$$

6 Evaluation of the Method based on Stochastic Recursive Equations

To conclude the paper, we provide an overview and an evaluation of the SRE approach. The reader is referred to [7, 8, 9] as well as previous sections for examples of using SRE technique for the performance analysis of parallel systems, and to [11, 10, 5] for the analysis of SRE's. The description of some subclasses of Petri nets by SRE can be found in [2, 4].

6.1 Basic Ideas and Concepts

Principle :

The basic idea of the SRE approach is to describe the temporal evolution of a discrete event system as mathematical relations between state variables.

The paradigm is a set of stochastic recursive equations

$$X_{n+1} = f_n(X_n, \xi_n), \tag{44}$$

where X_n is the state at the n-th event, and $\{\xi_n\}$ is the input sequence, f_n is the mapping characterizing the dynamics of the system.

In this regard, the SRE approach is methodologically analogous to the modeling of dynamic systems in physics, economy, biology and the "classical system theory".

Basic modeling elements :

State variables under consideration are usually

- daters $X_i(n)$: time epoch of the n-th event of type i;
- counters $N_i(t)$: number of events of type i occurred before time t;
- routers $u_{i,k}(n)$: indicator function for the n-th event of type i to generate an event of type k.

These variables are determined (or "driven") by the input variables $\{\xi_n\}$.

Statistical assumptions :

In general, the description of the system by stochastic recursive equations requires no statistical assumptions. The performance analysis, however, requires usually that the input sequence $\{\xi_n\}$ be stationary and ergodic.

Underlying mathematical concepts :

Very often, one uses

- ergodic theory;
- sample path comparison, monotonicity, coupling;
- stochastic ordering, majorization,

in the analysis of the system. More detailed presentations are provided in the next subsection.

Results on performance measures :

- first order quantities: throughput, stability conditions;
- second order quantities: delays, response times, waiting times, their distributions and correlations.

6.2 Technical Aspects

Description power :

The description power is unlimited a priori. All SRE uses are equations. It can thus simulate a programming language.

Structural and numerical tractability :

In order to use the SRE approach for the modeling and performance analysis of a system, it is desirable that the system have

- finite dimension;
- simple structure;
- explicit dynamics.

Sometimes, it is also desirable that the system have properties like continuity, monotonicity and convexity.

Examples of such systems are marked graphs which have linear equations in (max, +) algebra, Jackson-like queueing networks and Free Choice Petri nets.

Standard performance evaluation techniques :

- Exact analysis based on functional equations.
- Computation of bounds using stochastic comparison techniques:
  - Internal monotonicity: $X_n \leq_1 X_{n+1}$;

- External monotonicity: $\{\xi_n\} \leq_1 \{\xi'_n\} \quad \Rightarrow \quad \{X_n\} \leq_2 \{X'_n\}$;
- System monotonicity: $\{f_n\} \leq_1 \{f'_n\} \quad \Rightarrow \quad \{X_n\} \leq_2 \{X'_n\}$.

- Computation of asymptotics using large deviation techniques.
- Approximations: heavy traffic analysis, light traffic analysis, etc.
- Simulations: time-driven simulation techniques, with the advantages of being less time-consuming and easier to parallelize than event-driven simulations.

Analysis and composition of complex models :

Complex models can be studied using a "system theory" approach.

- Analysis of simple blocs is carried out by the aforementioned methods.
- Combination of blocs are carried out in accordance with the type of the connectors:
  - "or" connectors: this amounts to the superposition and thinning of the flows;
  - "and" connectors: this amounts to the fork and the join of the events;
- Composition of blocs is equivalent to the composition of functions.

Customer classes and priorities :

For systems containing multiple classes customers (or elements) and priorities, some dynamics are difficult to describe using recursive equations. Examples are LIFO, processor sharing, preemption, etc. This problem is currently under investigation.

6.3 Methodological Steps in Modeling Parallel Systems

Level of abstraction :

Stochastic (sometimes implicit) recursive equations can be used a priori to model parallel systems at any level of observation. However, as most other modeling tools, the higher the level of abstraction, the easier the solutions.

Phases of the modeling process and their interdependences :

Until now, there are very few studies on this issue. Top-down and bottom-up approaches should be possible.

Experience with the modeling of large systems :

On an abstract point of view, the size of the system to be modeled is irrelevant in the SRE approach. There is no particular difficulty in getting the solutions to the equations if the system is well structured (e.g., the system topology has a repeating pattern). Very good numerical results have been obtained (by, e.g. simulation) using the SRE's for such systems.

Tool support :

Two levels of tools are useful:

- At user level, it is desirable that the recursive equations be transparent to the users. These equations are just an internal representation of the system. Therefore, it is necessary to have tools generating SRE's from frequently used models like Petri nets, queueing networks and task graphs.
- At solution level, it is necessary to have tools for dependency analysis, structure analysis, equation simplifications. Note that in the process of equation simplifications, it is not only useful to reduce the number of variables, but also, and possibly more importantly, to rewrite equations into some special forms for which analysis techniques are available.

6.4 Relationships to Other Modeling Methods

Analysis, correspondences, equivalences :

The marked graphs (resp. the repeated task graphs with "and" logic in the precedence relation) are equivalent to the class of SRE's which have deterministic dynamics, i.e., $f_n = f$ a.s., and for which f is linear in $(\max, +)$ algebra.

Transformation techniques :

Techniques have been developed (for instance in the case of systolic array design) for dependency analysis, elimination of variables, rewriting of equations, change of time scale, optimization of data flow.

Complementation and supplementation :

The SRE's can be used as complementary and supporting tools:

- for multi-paradigm modeling, SRE's can be used as the unifying internal description;
- for solutions, stochastic comparisons based on SRE's can yield bounds by considering solvable models.

6.5 Advantages of the SRE approach

To summarize, the SRE approach

- puts emphasis on the description of the dynamics of the systems;
- carries out analysis under general statistical assumptions;
- permits random topological structure (or dynamics) of the system under consideration;
- allows one to develop time-driven simulation techniques based on the recursive equations.

References

[1] UER: descriptions of the parallel processing examples.

[2] F. Baccelli, "Ergodic Theory of Stochastic Petri Nets", *The Annals of Probability 20*, 1 (1992), pp. 375–396.

[3] F. Baccelli, M. Canales, "Parallel Simulation of Stochastic Petri Nets Using Recursive Equations." *ACM-Tomacs*, to appear.

[4] F. Baccelli, G. Cohen, B. Gaujal, "Recursive Equations and Basic properties of Timed Petri Nets", *Jounal of Discrete Event Systems 1* (1992), pp. 415–439.

[5] F. Baccelli, G. Cohen, G. J. Olsder, J.-P. Quadrat, *Synchronization and Linearity*, J. Wiley, 1992.

[6] F. Baccelli, P. Konstantopoulos, "Estimates of cycle times in stochastic Petri nets", in *Proc. of US-French Workshop on Applied Stochastic Analysis*, I. Karatzas, D. Ocone (Eds), Lecture Notes in Control and Information Sciences, **177**, 1991, pp. 1-20.

[7] F. Baccelli, Z. Liu, "On the Stability Condition of a Precedence-Based Queueing Discipline." *Advances in Applied Probabilities*, Vol. 21, pp. 883-898, 1989.

[8] F. Baccelli, Z. Liu, "On the executions of parallel programs on multiprocessor systems—a queueing theory approach", *Journal of the ACM*, **37** (1990), pp. 373-414.

[9] Z. Liu, F. Baccelli, "Generalized Precedence-Based Queueing Networks." *Mathematics of Operations Research*, Vol. 17, pp. 615–639, 1992.

[10] F. Baccelli, Z. Liu, "On a class of stochastic recursive sequences arising in queueing theory", *The Annals of Probability*, **20** (1992).

[11] F. Baccelli, A. M. Makowski, "Queueing models for systems with synchronization constraints", *Proc. IEEE* **77** (1989), pp. 138-161.

[12] J. F. C. Kingman, "The ergodic theory of subadditive stochastic processes," *J. Roy. Statist. Soc.*, Ser. B, **30** (1968), pp. 499-510.

[13] R. M. Loynes, "The Stability of Queues with Non-Independent Inter-arrival and Service Times", *Proc. cambridge Ph. Soc.* **58** (1962), pp. 497-520.

[14] L. Rüschendorf, "Ordering of Distributions, and Rearrangement of Functions", *Annals of Probability*, **2**, pp.276-283, 1981.

[15] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models*, English translation (D.J. Daley editor), J.Wiley and Sons, New York, 1984.

[16] V. Strassen, "The Existence of Probability Measures with Given Marginals", *Ann. Math. Stat.*, **336**, 423-439, 1965.