Effect of Stochastic Load Dependencies on Queue Sojourn Times

Matthias Ivers, Rolf Ernst

* Institute of Computer and Network Engineering
Technical University Braunschweig
{ivers,ernst}@ida.ing.tu-bs.de

Abstract: We discuss the applicability and consequences of probabilistic modeling for safety critical real-time systems. While deterministic task models characterize a task execution by its worst-case execution time (WCET), probabilistic models allow a better model especially for tasks with highly variable execution times by using a distribution on possible execution times. We apply a queuing sojourn time analysis of tasks with probabilistic execution times with upper and lower bounds guaranteed under any stochastic dependency. The results show how big the effect of dependencies really is and that execution time dependency is vital to the calculation of response times. We propose the use of Fréchet-bounds and probability boxes to make real-time systems with execution-time dependencies accessible to probabilistic real-time analysis.

Keywords: Probabilistic models, Conditional Probability, Real-time tasks, Execution time

1. INTRODUCTION

Real-time systems rely on the timely execution of all their functions to guarantee a system response before a deadline imposed by the environment. In hard real-time systems the violation of a single deadline is considered fatal. A real-time analysis computing upper bounds for the system response time is performed on a model of the system.

Deterministic real-time analysis models tasks only by an upper bound of their execution time; the worst-case execution time (WCET). This is a safe but pessimistic approach if the execution is strongly state dependent or if worst case guarantees can be replaced by probabilistic guarantees. This model will lead to severe overdimensioning if the WCET is much higher than the average execution time.

Stochastic analysis, pioneered for soft or firm real time systems design, tackles this pessimism by a more precise model: instead of a single worst case execution time, a task is characterized by a series of execution times together with a probability.

This precision comes with a higher computational cost and further restrictions on the analysed system: the analysis requires the knowledge of every dependency between job run times. (In this work “dependency” signifies a relationship between execution times probabilities. In other works the term dependency is also commonly used in the sense of “one task waiting for another task.”) We want to assess the effects of (uncaptured) dependencies between tasks. The dependency information between probability distribution functions is of vital importance to the correctness of the results.

2. RELATED WORK

Timing analysis of real-time systems traditionally characterizes tasks by their period and worst case execution time (WCET). The WCET of tasks is determined by different analytical methods. Methods like by Staschulat et al. (2005) split the analysis of tasks into an analysis of the run time of code segments without (or with a limited number of) conditional branches followed by the synthesis of an upper bound WCET for all possible paths through the program.

This static property is then used in a scheduling and system analysis to calculate end-to-end response times. Due to the abstraction, deterministic models yield inferior accuracy for applications with highly variable behaviour like multimedia streaming. In deterministic performance analysis, certain deterministic variations of streams are modeled in a context-aware multi-frame model. Henia et al. (2005) can make the analysis ‘application-aware’ to handle this situation as long as the load variation happens in a deterministic fashion. Purely probabilistic variations are not caught.

For the analysis of multimedia streaming applications specialized load models have been suggested. These models try to capture a detailed presentation of the load variance. Izquierdo and Reeves (1999) survey 19 different models proposed for VBR streaming applications. The models features vary with application content. Vartakar and Marculescu (2002) identify critical dependency parameters for streaming video application that probabilistic models fail to model. Krunz (2000) concentrates on the auto-correlation of VBR video streams and gives examples for the effects on buffer sizing. All these load models are specifically tailored to match the exact application and analysis method. The proposed models do consider the intra stream dependencies of the load and also consider multiple levels of auto-correlation within the stream load. We aim for a compositional system level analysis and the amount of specialization required for these models does not
seem suit our goals, as some parts (or the environment) of the system might not be fully known during analysis.

Concerning the scheduling analysis for probabilistic systems, frameworks proposed by Manolache (2005); Díaz et al. (2002); Atlas and Bestavros (1998) use ETPDFs or similar tools to model tasks with variable execution time. Although the probabilistic model is more precise, Bernat et al. (2002) rightly point out that they are only applicable if the tasks are stochastically independent. This requirement limits the applicability of the proposed models.

Calculations with random variables with unknown dependencies have been studied extensively (see Chebyshev (1874)) and are a common tool in risk theory. Bernat et al. (2002) try to identify a single worst case dependency which they use to calculate the worst case distribution. Bernat et al. (2005) propose the use of the supremal convolution, which is an upper bound on all possible convolutions. Further they use copulas to separate the modeling of stochastic behaviour of a single task and the stochastic dependency between tasks.

In this work we focus on the cause and effect of stochastic dependencies in systems of multiple, concurrent, potentially interacting tasks. We integrate the safe calculations with stochastic variables under unknown dependencies with the stochastic scheduling analysis proposed by Díaz, Kim et al. Our methods give more insight into the system response not only by giving a reliable upper bound on the stochastic response times, but as well a lower bound. The difference between lower and upper bound gives direct feedback about the potential impact of dependencies.

3. PROBLEM STATEMENT

Given a set of tasks $S = \{τ_1, τ_2, ..., τ_n\}$, $τ_i = (c_i, d_i, M)$. Where $c_i$ is the execution time and $d_i$ is the relative deadline of task $τ_i$ and $M$ is real number between 0 and 1. $c_i$ is a discrete random variable with probability mass function $f_{c_i}(c) = P(c_i = c)$ and cumulative probability function $F_{c_i}(c) = P(c_i \leq c)$. (Where $P(x)$ is the probability that in the specific system the event $x$ is observed.) Furthermore a series of jobs $\tau_{i,j} = (\tau_i, a_{i,j})$ with task $\tau_i$ and an arrival time $a_{i,j}$ is given.

We assume a static priority preemptive scheduling. The tasks $\tau_i \in N$ are w.l.o.g. ordered by priority. Jobs violating their deadlines are instantly killed.

For each job $\tau_{i,j}$ the response time is given by the random variable $R_{i,j}$. Job $\tau_{i,j}$ is said to fulfill its QoS requirement $(d_i, M)$ if $F_{R_{i,j}}(d_i) \geq M$. We are seeking to find for all jobs $\tau_{i,j}$ responses time $F_{R_{i,j}}(R_{i,j})$ to the distributions of the response time $R_{i,j}$ fulfilling $\forall r \in N, F_{R_{i,j}}(r) \geq F_{R_{i,j}}(r)$. These bounds should be valid under arbitrary dependencies.

4. INDEPENDENT JOBS

First, we reproduce the results for the response time analysis of independent tasks. In the following we will extend the theory to handle unknown dependencies.

One main differentiator of the works is the definition how two random variables are added: Given two independent

![Fig. 1. Stochastic task interrupted - approximate and exact solution](image-url)

random variables $X$ and $Y$ the distribution of the sum $Z = X + Y$ can be calculated as:

$$F_Z(z) = \sum_{z=x+y} F_X(x) \cdot F_Y(y)$$

This is the basic convolution that is assumed for the following works. We will later extend this addition to handle unknown dependencies.

Tia et al. (1995) reinterpret the classical scheduling formula for probabilistic systems. For a job $\tau_{i,j} = (\tau_i, a_{i,j})$ and time interval $[a_{i,j}, \tau_{i,j} + t]$ let $\tau_{i,j}, \tau_{i,j}, ... \tau_{i,n}$ be the finite series of higher priority job arrivals from $a_{i,j}$ until $a_{i,j} + t$. $\tau_{i,n}$ is the last job started before $a_{i,j} + t$. First an upper bound to the response time distribution at time $t$ is defined:

$$R_{i,j}(t) = E_c + E_{\chi_1} + ... + E_{\chi_n}$$

(1)

With $E_{\chi_n}$ being the execution time distribution of $\tau_{i,n}$.

The probability for completion before the deadline is

$$\max\{F_{R_{i,j}}(t) | t \in E\}$$

(2)

Where $E$ is the set containing the $d_i$, the deadline of task $\tau_{i,j}$, and all arrival times of higher priority tasks before $d_i$. The authors remark the potential problems of the assumed independence in their approach.

Eq 1 is a straightforward extension of the deterministic case. It does not differentiate 'when' the interference happens. Consider a task with random execution time executing for 2 time units and then being interrupted (by a deterministic task) for 4 time units. The left part of figure 1 follows the assumption implicitly made in eq 1. The right side of the figure shows how taking the offset between the preemption and the activation into account improves the response time analysis.

Díaz et al. (2002) propose the operation "convolve from $r$" that respects the condition of the interference: only instances that take longer than $r$ are interfered.

The random variable $Y$ is added if and only if $X$ exceeds $r$. To achieve this, $F_X(x)$ is split into $F_X^{(0,r)}(x)$ and $F_X^{(r,\infty)}(x)$ where $x \in I \Rightarrow F_X^{(0,r)}(x) = F_X(x)$ otherwise $F_X^{(r,\infty)}(x) = 0$.

The sum $Z$ of $X$ from $r$ and $Y$ under independence (written $Z = X + Y$) has the distribution:

$$F_Z(z) = F_X^{(0,r)}(z) + \sum_{z=x+y} F_X^{(r,\infty)}(x) \cdot F_Y(y)$$

The operation $+_{r}$ is non-associative. We define the operation $+_{r}$ to be left-associative. For better readability parentheses are omitted.

The response time distribution is given by

$$R_{i,j} = E_c + \delta_{x_0} E_{\chi_0} + \delta_{x_1} E_{\chi_1} + ... + \delta_{x_r} E_{\chi_r}$$

(3)

$\delta_{x_n} := a_{x_n} - a_i$ is the arrival time difference between the analyzed job $\tau_{i,j}$ and $\tau_{i,n}$.
Interference which occurs only under the condition that the execution took longer than a specific time now affects only that part of the distribution. The probability for completion within the deadline is thus $F_{\tau_{i,j}}(d_i)$.

5. DEPENDENCIES IN EXECUTION TIMES

The previous analysis assumed for all random variables independence. The results are only valid if we can assure for any two jobs $\tau_{i,j}, \tau_{i',j'}$ in the system that the execution time distribution of one job does not change when another job has a certain execution time ($\mathcal{E}_{i,j} = c'$).

$$\forall (i,j) \neq (i',j'), P(\mathcal{E}_{i,j} = c) = P(\mathcal{E}_{i,j} = c | \mathcal{E}_{i',j'} = c')$$

(4)

We will depict two different types of dependency. First a dependency between the jobs of a single source and then a dependency between the jobs of different sources.

An **intra source dependency** is a dependency between two jobs of the same 'source'. This kind of dependency typically occurs in streams of jobs which have an inherent recurring regularity. An intra source dependency exists if

$$\exists n > 0, P(\mathcal{E}_{i,j} = x | \mathcal{E}_{i,j-n} = y) \neq P(\mathcal{E}_{i,j} = x)$$

(5)

An intra source dependency exists whenever the execution time of a job depends on the execution time of previous jobs.

An example for this scenario is the processing of MPEG streams. MPEG streams consist of a series of so-called I-, B-, and P-Frames. An I-Frame will typically be followed by a number of B- and P-Frames. Conversely the occurrence of two consecutive I-Frames is rare. As the different types of frames have very different transmission lengths and execution times associated with them, this qualifies as an intra source dependency.

We speak of an **inter source dependency** if two or more sources change the load characteristic at the same time. An inter source dependency exists if there are two jobs $\tau_{i,j}, \tau_{i',j'}$ with $(i,j) \neq (i',j')$ that satisfy

$$P(\mathcal{E}_{i,j} = c | \mathcal{E}_{i',j'} = c') \neq P(\mathcal{E}_{i,j} = c)$$

(6)

As an example consider variable bit-rate video streams with isolated processing of audio and video. Streaming rates depend on the 'level of action' inside a stream. Cam scenes are probably accompanied by calm sounds and scenes with higher activity typically have high activity at both levels, audio and video. If one task is 'nearly idle', the other is probably as well. This is an inter source dependency.

5.1 Modeling of Dependency

We can model the above-mentioned dependencies by sources emitting colored tasks. A colored task $\tau_i = (c_{k_0}, c_{k_1}, c_{k_2}, ..., d_i)$ is a deadline $d_i$, a set of colors $k_0, k_1, ...$ and for each color a response time $c_{k_0}$. A colored job $\tau_{i,j} = (\tau_i, \alpha_{i,j}, k_{i,j})$ is a task with an activation time $\alpha_{i,j} \in \mathbb{N}$ and colored token $k_{i,j} \in \mathbb{K}$.

The execution time $\mathcal{E}_{i,j}$ of task $\tau_{i,j}$ is chosen as determined by the color $k_{i,j}$. No other data than the color changes the odds of the random execution time $\mathcal{E}_{i,j}$.

More formally

$$P(\mathcal{E}_{i,j} = c | k_{i,j} = k) = P(\mathcal{E}_{i,j} = c | k_{i,j} = k \land \phi)$$

(7)

\[ \text{Fig. 2. A Probability Box} \]

\[ \text{where } \phi \text{ is any formula which does not contain } c_{i,j}. \]

\[ \text{Using this model, we consider the job source as the generator of stochastic dependencies within the system.} \]

\[ \text{This model holds for a big class of systems and resembles multi-modal tasks in deterministic analysis.} \]

6. TAKING DEPENDENCIES INTO ACCOUNT

Analyzing a job $\tau_{i,j}$, we have to consider all intra source dependencies for jobs of higher priority which can be activated multiple times before the deadline $d_i$ (i.e. jobs of smaller period than $\tau_i$). Additionally for all concurrently running higher priority jobs, we have to consider a potential inter source dependency for the sum of jobs of different priorities.

For this we have to use distribution functions that are able to represent uncertainty (i.e. distribution functions which can bound the probability for the value to be within a certain interval). To reason about interval bounds of stochastic variables, we will introduce so-called probability boxes.

6.1 Probability Bounds & Probability Boxes

Ferson et al. (2002) use probability boxes (p-box) to introduce interval arithmetic for random variables. Given a cumulative distribution $F_X : \mathbb{N} \rightarrow [0, 1]$, a p-box is a pair of functions $F_X, \overline{F}_X : \mathbb{N} \rightarrow [0, 1]$ with

$$ \overline{F}_X(x) \leq F_X(x) \leq F_X(x)$$

(8)

Execution time distributions $F_X$ map a time $x$ to the probability that the execution time is less than or exactly $x$ time units. Execution time p-boxes $(\overline{F}_X, F_X)$ map a time $x$ to a minimum and maximum probability that the execution time is less than or exactly $x$ time units (e.g. figure 2: the interval for $x = 11$ is marked by the vertical arrow).

Probability boxes can also be used to read an execution time interval for a given probability. E.g. the horizontal arrow in figure 2 shows that the WCRT (the response time with an accumulated probability of 100%) ranges from 10 to 16 time units. These values say nothing about the best-case response time; the meaning is that there may exist a system with a worst-case response time of only 10 time units.
Probability bounds can be efficiently described with probability boxes. The remaining question is how the sum of stochastic variables should be safely calculated. Makarov (1981); Frank et al. (1987); Williamson and Downs (1990) study arithmetic on stochastic variables with unknown dependencies.

To formalize dependencies between stochastic variables copulas are used. Copulas model the relation between (typically available) marginal distributions and their joint distribution, i.e.: Given a two-dimensional distribution function \( F_β(x,y) \) with (one-dimensional) marginals \( F_β(x) \) and \( F_β(y) \). Then there exists a copula \( C \) such that
\[
F_β(x,y) = C(F_β(x), F_β(y))
\]
(9)

In our situation, 2 marginals \( \mathcal{F} \) and \( \mathcal{G} \) for different tasks/jobs are given. The unknown dependency is modelled only by \( C \). Obviously \( C \) is a function \( C : [0,1] \times [0,1] \rightarrow [0,1] \). Furthermore it has been proven that all copulas satisfy
- \( C(a,0) = C(0,a) \) and \( C(a,1) = C(1,a) \) for all \( a \in [0,1] \)
- they are 2-increasing: i.e. \( C(a_2, b_2) - C(a_1, b_2) - C(a_2, b_1) + C(a_1, b_1) \geq 0 \) for all \( a_1 \leq a_2, b_1 \leq b_2 \).

Given these constraints, there exists a unique smallest and a unique largest copula. Namely \( W(a,b) := \max(a+b-1,0) \) and \( M(a,b) := \min(a,b) \). These copulas are handy as all copulas satisfy
\[
W(a,b) \leq C(a,b) \leq M(a,b)
\]
(10)

This observation lead to the Fréchet-bounds, that give upper and lower bounds on the effect the dependency between marginals can have on the joint distribution:
\[
\max(F_β(x)+F_β(y)-1,0) \leq F_β(x,y) \leq \min(F_β(x), F_β(y))
\]
(11)

Another commonly used copula is \( \Pi(x,y) := x \cdot y \) which models stochastic independency of two random variables.

Copulas and especially the \( \Pi \) copula give lower bounds for the probability \( P(X = x_1 \land Y = y_1) \) given the probability for \( P(X = x_1) \) and \( P(Y = y_1) \). This is closely related to the sum of random variables, yet a little extension is necessary as we are not interested in the probability of \( P(X = x_1 \land Y = y_1) \), but instead we search for any \( z_1 \) the probability \( P(X + Y = z_1) \).

Williamson and Downs (1990) give probability bounds for the sum of two random variables \( Z = \mathcal{E}_1 + \mathcal{E}_2 \). That we present here extended to probability boxes: The sum of two stochastic variables described by probability boxes \( (\mathcal{F}_1, \mathcal{F}_2) = (\mathcal{F}_1, \mathcal{F}_2) + (\mathcal{F}_2, \mathcal{F}_2) \) is calculated as follows:
\[
\mathcal{F}_2(t) = \inf_{c_1 + c_2 = t} \{ \min(\mathcal{F}_1(c_1) + \mathcal{F}_2(c_2), 1) \}
\]
(12)
\[
\mathcal{F}_2(t) = \sup_{c_1 + c_2 = t} \{ \max(\mathcal{F}_1(c_1) + \mathcal{F}_2(c_2) - 1, 0) \}
\]
(13)

The proposed operation for the addition of two random variables has been proven to be safe and furthermore point-wise best-possible by Williamson and Downs (1990). That means for any bound tighter than \( (\mathcal{F}_1, \mathcal{F}_2) \), one can find a dependency between the underlying random processes, that would lead to a violation of these (tighter) bounds.

6.2 Adaptions to Scheduling Analysis

We will now integrate Fréchet-bounds and probability boxes into the scheduling analysis formula using the “convolve from” operation. In order to do that, we basically have to lift the “convolve from” operation from simple distributions to probability boxes, i.e. the inputs of the convolve from operation are now probability boxes, and the output of the convolve from operation are probability boxes as well.

As this lifting from distributions to probability boxes is intriguingly conclusive, we show directly the “convolve from” operation using probability boxes and the Fréchet-bounds:

The sum of \( X \) from \( r \) and \( Y \) under any dependence \( (Z = X + r, Y) \) has following bounds to the distribution:
\[
F_Z(z) = F_X^0(r)(z) + \inf_{z = x + y} \{ \min(F_X(r, \infty)(x) + F_Y(y), 1) \}
\]
(14)
\[
F_Z(z) = F_X^0(r)(z) + \sup_{z = x + y} \{ \max(F_X(r, \infty)(x) + F_Y(y) - 1, 0) \}
\]
(15)

The operation \( + \) is non-associative. We define the operation \( + \), to be left-associative. For better readability parentheses are omitted. The response time distribution is given by
\[
\mathcal{R}_{i,j} = \mathcal{E}_1 + \delta_{x_0} \mathcal{E}_x_0 + \delta_{x_1} \mathcal{E}_x_1 + ... \delta_{x_m} \mathcal{E}_x_m
\]
(16)
\( \delta_{t_i} \) is the arrival time difference between the analyzed task and \( t_i \).

The equation is well defined, as we can write \( \mathcal{E}_i \) for the bounding p-box \( (\mathcal{F}_1, \mathcal{F}_2) \) with \( \mathcal{F}_1 = \mathcal{F}_2 = \mathcal{F}_2 \). Following our previous reasoning about the Fréchet-bounds we can see that
\[
\forall i \mathcal{R}_{i,j}(t) \geq \mathcal{F}_X(t) \geq \mathcal{F}_Z(t)
\]
(17)

Using this function for our scheduling analysis, we calculate safe and sharp bounds to the distributions. Additionally, we also get a notion of how much the missing dependency information is affecting the system, as we also calculate a safe and sharp lower bound.

7. COMPARISON

Kim et al. (2005) present a scheduling analysis for priority-driven periodic real-time systems. They give an example for the calculation of the response time distribution of a job based on a given inter-arrival pattern of different jobs and the job’s service times given as a distribution.

Figure 3 presents task activations and the distributions of \( t_1, t_2 \) and \( t_3 \). The six graphs of figure 4 show the remaining workload distribution at different times. All graphs show results assuming independence and the proposed p-box based analysis. Díaz’ result (independence) is always within the p-box.

After 3 time units (tu), \( t_1 \) is activated: as the current workload is zero, the workload becomes exactly \( t_1 \). In the first graph only a single line is visible, as both approaches are equal at 3tu.

Until 10tu, no task is activated. Modeling this, the graph is shrunk to the left accumulating probabilities
for negative response times at 0. For details on shrinking see Kim et al. (2005) or the examples below. At 10u, a second job starts. The task execution time distribution is added to the workload distribution (second graph).

The third graph shows the workload distribution at the activation time of $t_3$ (16u). The next graphs show the workload of the task $t_3$ at 16u considering the preemption at 16, 16+6 and 16+8. The graphs are no longer shrunk, instead the activation time of the interrupting tasks is added starting from 3, 6 or 8u (“convolve from r”). The last graph shows the actual response time distribution of this invocation of $t_3$. This last graph is the job response time.

The big height of the p-box demonstrates how sensitive the system is to execution time dependencies. Remarkably the result assuming independence is quite far away from the ‘worst-case’ p-box bound. As the Fréchet-bounds used to calculate the p-box are known to be sharp, we can assure that a wrongly assumed independency can practically lead to misleading optimistic results.

8. CONCLUDING EXAMPLES

To conclude the paper we give two specific examples how dependencies, that produce this big difference, might look like.

8.1 $x=3 \ y=1.0$ example

Figure 5 shows the workload on the left and new tasks on the right. All distributions are colored 'a', 'b', 'c' showing the dependencies between the different task activations. As $t_3$ finishes within 3u, we do not give graphs for the succeeding invocations of $t_2$.

8.2 $x=10 \ y=0.66$ example

Figure 6 gives another example. For the distributions at $t = 16 + 3$ and beyond, only the case 'd' of the graph is affected, as this is the only case which is inside the manipulated part of the queue distribution (+n only affects the part of the distribution which is > n).
Activation Stream:

\[ t_1 \quad t_1 \quad t_3 \quad t_2 \quad t_2 \quad t_2 \]

Task Models:

\[ 0.33 \quad 0.33 \quad 0.33 \quad 0.5 \quad 0.5 \]

Fig. 3. Activation Diagram and Task Models

Fig. 4. First steps of Kim and Díaz’ example (at time: 3, 10, 16, 16+3, 16+6, 16+8)

The given dependency leads to \( F_R(10) \approx 0.667 \). Under independence the guarantee is far more optimistic with \( F_R(10) \approx 0.993 \).

REFERENCES


