Availability Evaluation of Database Systems under Periodic Checkpoints

Reinaldo Vallejos Campos*
Technical University F. Santa Maria
Electronics Department, Valparaiso, Chile

Edmundo de Souza e Silva¹
Federal University of Rio de Janeiro, NCE
CXP. 2324, CEP 20001, Rio de Janeiro, Brazil

Abstract

Checkpointing roll back and recovery is a common technique to insure data integrity, to increase availability and to improve the performance of transaction oriented database systems. Parameters such as the checkpointing frequency and the system load have an impact on the overall performance and it is important to develop accurate models of the system under study. We find expressions for the system availability from a model that, unlike previous analytical work, takes into account the dependency among the recovery times between two checkpoints. Furthermore, our model can incorporate details concerning the contention for the system resources.

keywords: availability modeling, performance modeling, checkpointing, database recovery, uniformization, Markov models.

1 Introduction

Transaction oriented database systems are being widely used in a variety of different applications and there is an increasing need for guaranteeing data integrity and providing high data access availability and fast response time. One of the most common techniques to insure data integrity in presence of failures is to save periodically, in a secondary storage device, all information necessary to restart the system from a known state. When a fault occurs, the system is restored from that state and the information saved is used to re-execute all the transactions until the time when the fault occurs. The process of making periodic copies of the system state is called checkpointing. After a checkpointing is performed, all transactions that modify the file system are saved in a special file called audit trial. When a failure occurs (hardware or software), the system is first restored to the state of the last checkpoint. This operation is called roll back. Then the recovery phase starts and all the transactions saved in the audit trial file are executed, restoring the system.

¹The work of E. de Souza e Silva was partially supported by a grant from CNPq(Brazil) and NSF CCR-9215064.
to the state immediately before the fault. Normal processing of new transactions is reasserted immediately after recovery. Note that the recovery time is equal to the time it takes to execute all the transactions from the last checkpointing, and it is typically shorter than the available period until the failure, and depends on the system load.

Many models were developed in the literature to study this systems, e.g. [3, 4, 5, 6, 8, 9]. They differ, in general, in terms of the modeling assumptions made, the measure of interest to be obtained and the technique used to solve the model. Gellmbe [3] assumed that the roll back and recovery operation for the \( i \)-th failure is proportional to the elapsed time between the end of the last checkpoint and the time of occurrence of this failure. Furthermore, the number of failures in the available period is given by an independent Poisson process. Under these assumptions it was shown that, in order to maximize the system availability, the available time must be a constant. Subsequently in [4] it was shown that this result is also valid when the failure rate is not constant. In [8] several different models where considered and one of the goals was to determine the most important assumptions in order to obtain an accurate model. One conclusion of that work indicates that the recovery time, i.e. the time to re-execute all transactions which arrived from the last checkpoint to the current failure, should be accurately modeled and the dependencies between recoveries from distinct failures in the same cycle should be preserved to obtain a realistic model. More recently, in [6], the Laplace transform of the time the system is available to perform useful work in a fixed interval was obtained. In that work it is also assumed that the recovery time is proportional to the elapsed time from the end of the last checkpointing operation to the present failure. This is a common assumption in the literature, but it does not model exactly the real time spent processing transactions until a failure occurs. For instance, if the transaction system is modeled by a single server queue, the time spent executing transactions until a failure occurs is the sum of the busy periods during a finite period whose length is a random variable.

In this work we calculate the system availability from a more accurate system model than those considered in previous analytical work. In particular the recovery time is obtained from a separate system model that takes into account the contention for the system resources and is incorporated in the final global model. We show that the availability is given by simple expressions of the model parameters. In section 2 we present the model assumptions and introduce our notation. In section 3 we develop the approach to calculate the measures of interest. A simple example is presented in section 4 and in section 5 we present our conclusions.

2 The Model

We begin by introducing the basic notation we use throughout the paper. Consider a cycle between two checkpoints. Figure 1 presents a detailed description of typical events in a cycle. In this figure \( C^* \) is a random variable which indicates the total time needed to store all the database state information. If a failure occurs during a checkpointing operation, the system has to restart the checkpointing process. In Figure 1, \( m \) failures occur during this process and \( \tau_f^i \) is the time of the \( i \)-th failure. We let \( C \) be the total amount of time needed to complete the operation. The number of failures that can occur during this time is given by a Poisson process with rate \( \gamma_C \).

After all the state is stored, the system is ready to execute transactions and is made available
to the user. We assume that $D$, the total amount of time the system is available to execute transactions between two checkpoints, is a constant.

![Diagram of a cycle between two checkpoints]

Figure 1: A cycle between two checkpoints.

During the available period the system may have to recover from failures. Similarly to the checkpointing period, the number of failures in the available interval is assumed to be Poisson with rate $\gamma_d$. In Figure 1 we represent the amount of time spent with recovery operations by vertical lines. The time of the first failure from the last checkpoint is $t_{f1}$ and $t_{b1}$ is the total processing time during this interval. (The thick lines represent the system busy periods.) Each roll back is assumed to last a fixed amount of time $t_r$ and $r_i$ is the total amount of time of the roll back and recovery operation after the $i$-th failure. Let $x_i$ be the duration of the $i$-th busy period in the cycle. From the figure, $r_1 = x_1 + x_2 + t_r$, $r_2 = x_1 + x_2 + x_3 + x_4 + t_r$, and so on. We assume that there can be no failures during the roll-back and recovery operations.

Most previous work assume that the recovery times are independent variables each of them depending only on the interval between the last checkpoint and the associated failure (see [8] for a discussion of many model assumptions commonly made). In this paper those intervals are dependent random variables, and each is equal to the accumulated time processing transactions as indicated above. In order to obtain the busy periods, we need a model of the contention for the system resources. The contention for resources can be modeled by a homogeneous continuous time Markov process $\mathcal{X} = \{X(t) : t \geq 0\}$ with finite state space $S = \{s_i : i = 1, \ldots, M\}$ and generator matrix $Q$. For instance, we can use a simple model where the system resources are represented by a single exponential queue and transactions arrive according to a Poisson process. Clearly, more complex models can also be used. For example we can consider phased type service times and arrival process.
Initially, we assume that no transactions can arrive during checkpointing, roll back and recovery operations. This assumption is reasonable considering that in general the system is not made available to the user during these operations and, as we will show later, the availability obtained in this case is an upper bound for the availability when transactions are allowed to queue during checkpointing and roll back and recovery. It should be noted that the Markovian state of the contention model at the beginning and end of a checkpoint period is the same. Furthermore, the state is also the same immediately before a failure and immediately after the recovery operation.

Later we obtain a lower bound for the availability when arrivals are allowed to queue during the checkpointing and recovery operations. In this case we assume that, after any of these operations, the system starts from an "overload state", i.e. a state where the system is overloaded with transactions. For instance, when the resource model is a single exponential queue the "overload state" is the one where the buffer is full with transactions to be processed.

3 The Mathematical Development

The objective of this section is to present the main development needed to obtain the availability, i.e. the long term fraction of time the system is available to execute transactions. We first study the system during a cycle and obtain the quantities needed for calculating the availability. Those quantities are conditioned on the state of the contention model at the beginning of the cycle. In order to obtain the availability, we define an embedded Markov chain at the beginning of a cycle and calculate the probability distribution of the states of the contention model at the embedded points. The final measure is obtained by using results from Markov chains with rewards, as it was done in [1].

3.1 Expected Values During a Cycle

We refer to Figure 1 which shows the events for a cycle and indicates the notation used. Let \( T_i \) be the expected duration of a cycle, given that the state of the contention model at the beginning of the cycle is \( s_i \). The cycle length is the sum of the duration of the checkpointing operation \( C \), the available time \( D \) (we recall that \( D \) is a constant) and \( F_i \), defined as the total amount of time spent in the roll back and recovery operations, conditioned that \( s_i \) is the state of the contention model at the beginning of the cycle:

\[
E[T_i] = E[C] + D + E[F_i].
\]  

(1)

Note that, since we assumed that no transactions arrive during the checkpointing, \( s_i \) is also the state at the beginning of the available period. Furthermore, the duration of the checkpointing is independent of \( s_i \).

We first obtain \( E[C] \). As discussed in previous sections, each time a failure occur during a checkpointing, the operation has to be restarted. Conditioning on the time of the first failure and solving the resulting equation we obtain

\[
E[C] = \frac{e^{\gamma_c C^*} - 1}{\gamma_c}
\]  

(2)
where we recall that $C^*$ is the total time to perform a checkpointing when no failures occur, and $\gamma_c$ is the rate of failures during this operation.

Let $N_D$ be the random variable which indicates the total number of failures during the available period. Conditioning on the value of $N_D$ and recalling that the number of failures in the available interval is a Poisson process with rate $\gamma_D$ we have:

$$E[F_i] = \sum_{l=0}^{\infty} e^{-\gamma_D l} \frac{(\gamma_D l)^l}{l!} E[F_i | N_D = l].$$  \hfill (3)

The value of $F_i$ is the sum of the time spent with roll back and recovery after each failure. Let $F_i(k, l)$ be the roll back and recovery time just after the $k$-th failure during the available period, given that $l$ failures occur. Then,

$$E[F_i | N_D = l] = E[\sum_{k=1}^{l} F_i(k, l)].$$  \hfill (4)

### 3.1.1 The Upper Bound Model

We first consider the assumption that no transactions are allowed to arrive during the operations of checkpointing and roll back and recovery. The lower bound model is considered in the subsequent subsection.

In order to evaluate (4) we first observe that, given $N_D = l$, the $l$ failure times are distributed as the order statistics of $l$ independent and identically distributed (i.i.d.) random variables uniform on the interval $(0, D)$. Second, we note that $F_i(k, l)$ is the same function for different values of $k$. The function depends on the length of the interval from the end of the last checkpoint until the $k$-th failure and not on the order of the particular failure. This is true since the state of the system immediately before a fault is the same as the state immediately after the fault, due to the assumption that no transactions arrive during roll back and recovery. From these observations, and since we are summing over all the $l$ recovery intervals, we can consider the times at which the failure events occur as unordered independent random variables uniformly distributed over the available interval. Finally, from the previous section, $F_i(k, l)$ is the sum of the constant roll back time $t_r$ plus the busy period from the beginning of the cycle until the time of the $k$-th failure. Therefore, we can write:

$$E[F_i | N_D = l] = l(t_r + E[B_i])$$  \hfill (5)

where $E[B_i]$ is the expected busy period during an interval $(0, \theta)$ conditioned on $s_i$, the state of the contention model at the beginning of the interval, and $\theta$ is a random variable uniformly distributed over $(0, D)$.

Substituting (5) into (3) we finally obtain:

$$E[F_i] = \gamma_D D(t_r + E[B_i])$$  \hfill (6)
\[ E[B_i] = \frac{1}{D} \int_{t=0}^{D} E[B_i|\theta = t]dt \]  

(7)

Let \( B \subset S \) be the subset of states that represent a busy system in the contention model. (The contention model is a Markov process \( \mathcal{X} \) with state space \( S \).) Then \( E[B_i|\theta = t] \) is the expected value of the total accumulated time in \( B \) during an interval of length \( t \). This quantity can be easily evaluated by Uniformizing [7] the Markov process \( \mathcal{X} \) and then calculating the expected cumulative reward after assigning a reward equal to 1 to states in \( B \) and 0 otherwise. From equation (8) in reference [2],

\[ E[B_i|\theta = t] = t \sum_{n=0}^{\infty} e^{-\Lambda t} \frac{\Lambda t^n}{n!} \left[ \sum_{j=0}^{n} \frac{\sum_{u: s_u \in B} \pi_u(j)}{n+1} \right] \]  

(8)

where \( \pi_u(j) \) is the probability that the discrete time chain \( Z = \{Z_n : n = 0, 1, \ldots\} \), obtained after randomizing \( \mathcal{X} \), is in state \( s_u \) after \( j \) steps. The vector \( \pi(j) = (\pi_1, \ldots, \pi_M(j)) \) satisfies the recursion \( \pi(j+1) = \pi(j)P \), where \( P = Q/\Lambda + I \) is transition probability matrix for chain \( Z \), \( \Lambda \) is the uniformization rate and \( I \) is the identity matrix. Details concerning the uniformization procedure and the calculation of useful performance measures in Markov models can be found in [2]. Since we are analyzing a cycle that starts in state \( s_i \), the initial distribution \( \pi(0) \) is a vector where all elements are zero except the \( i \)-th element which is equal to one.

Substituting (8) in (7), we obtain:

\[ E[B_i] = \frac{1}{DA^2} \sum_{n=0}^{\infty} \sum_{j=0}^{n} \sum_{u: s_u \in B} \pi_u(j) \int_{t=0}^{D} e^{-\Lambda t} \frac{\Lambda t^{n+1}}{(n+1)!} dt. \]  

(9)

The term in brackets in the equation above is the \((n+2)\)-stage Erlangian density and thus the integral in (9) is the corresponding distribution. Then,

\[ E[B_i] = \frac{1}{DA^2} \sum_{n=0}^{\infty} \sum_{j=0}^{n} E_{n+2, \Lambda}(D) \sum_{u: s_u \in B} \pi_u(j) \]  

(10)

where \( E_{n+2, \Lambda}(D) = 1 - \sum_{k=0}^{n+1} e^{-\Lambda D} \frac{(\Lambda D)^k}{k!} \) is the \((n+2)\)-stage Erlangian distribution.

3.1.2 The Lower Bound Model

We now assume that, after a checkpointing or recovery operations, the system always starts in an "overload state". Consider the single queue resource model. Intuitively, if the model starts from a state indicating a full buffer, the busy period between the end of a recovery operation and the next failure is always greater than the busy period when the model starts from any other state, i.e. any state with less transactions waiting for being processed. (This can easily be rigorously proven.) Let \( s_0 \) be this overload state.

Equation (4) is still valid but not equation (5), since, unlike the previous case, \( F_0[k, l] \) is dependent on the order of the particular failure. Let \( B_0[k, l] \) be the busy period during the \((k-1)\)th and
the kth failure, given that l failures occurred between (0, D). We can write:

\[ F_o(k, l) = t_r + \sum_{j=1}^{k} B_o(j, l) \text{ and so } E[F_o(k, l)] = t_r + \sum_{j=1}^{k} E[B_o(j, l)] \]  

(11)

From the assumption that the system starts from the same state (s_o) after each checkpointing or roll back operation, \( E[B_o(j, l)] \) depends only on the length of the jth interval between the (j - 1)th and the jth failure. We recall that, conditioned on l failures, the failure times are distributed according to the order statistics of l i.i.d. random variables in (0, D), and it is well known that the length of the jth interval has the same distribution as the length of the first interval, for any j.

As a consequence \( E[F_o(k, l)] = t_r + kE[B_o(1, l)] \) and so

\[ E[F_o|N_D = l] = lt_r + \sum_{k=1}^{l} kE[B_o(1, l)] = lt_r + \frac{l(l+1)}{2} E[B_o(1, l)] \]  

(12)

It remains to calculate \( E[B_o(1, l)] \).

Let \( \delta \) be the random variable equal to the length of time from the end of the checkpointing until the first failure and \( f_o(t) \) be its density function. \( \delta \) is distributed according to the first order statistics of a set of \( l \) uniformly i.i.d. random variables, and its density is given by

\[ f_o(t) = \frac{l}{D} \left( 1 - \frac{t}{D} \right)^{(l-1)} \]  

(13)

Conditioning on the length of the interval until the first failure we have

\[ E[B_o(1, l)] = \int_{t=0}^{D} E[B_o(1, l)|\delta = t] \frac{l}{D} \left( 1 - \frac{t}{D} \right)^{(l-1)} dt. \]  

(14)

We now note that equation (8) gives an expression for the expected busy period during an interval of length t when the initial state is s_o. Substituting (8) into (14) we obtain:

\[ E[B_o(1, l)] = \frac{1}{\Lambda} \sum_{n=0}^{\infty} I(n, l) \sum_{k=0}^{n} \sum_{u:s_u \in \mathcal{B}} \pi_u(k) \]  

(15)

where

\[ I(n, l) = \int_{t=0}^{D} e^{-\Lambda t} (\Lambda t)^{n+1} \frac{l}{(n+1)!} \left( 1 - \frac{t}{D} \right)^{(l-1)} dt \]  

(16)

It can be shown that

\[ I(n, l) = \frac{l}{(\Lambda D)^j} \sum_{j=1}^{\infty} e^{-\Lambda D} \frac{\Lambda D)^{(n+j)}}{(n + j)!} \frac{(j + l - 2)!}{(j - 1)!} \]  

(17)

Substituting (17) into (15) and the result into (12), we obtain, after unconditioning on l and some simple algebraic manipulations

\[ E[F_o] = \gamma_DDt_r + \sum_{l=1}^{\infty} e^{-\gamma_DD} (\gamma_DD)^l \frac{l(l+1)D}{l!} 2(\Lambda D)^{l+1} \sum_{k=0}^{\infty} \{ \sum_{u:s_u \in \mathcal{B}} \pi_u(k) \} \sum_{j=1}^{\infty} \frac{(j + l - 2)!}{(j - 1)!} E_{k+j+l, \Lambda}(D) \]  

(18)

This expression can be easily evaluated recursively.

We have obtained all quantities needed to calculate the expected cycle length conditioned on the state at the beginning of the cycle. In what follows we obtain a simple closed form expression for the system availability, for the first model.
3.2 The System Availability

From the quantities obtained for a cycle we can calculate performance and availability measures of the system. The development follows the approach of [1] to obtain performance/availability measures for scheduled maintenance policies. We first identify an embedded Markov chain \( \mathcal{Y} = \{ Y_k : k = 0, 1, \ldots \} \) at time points \( \{ \tau_k : k = 0, 1, \ldots \} \) (\( \tau_0 = 0 \)) where \( \tau_k \) is the beginning of the \( k \)-th checkpointing cycle. The states of this embedded chain correspond to the states of the contention model for the transactions. Now consider a measure \( M \) and suppose we want to calculate its steady state expected value. We associate a reward \( M_i \) with state \( s_i \) of the embedded chain \( \mathcal{Y} \) which is equal to the accumulated time the process \( X(t) \) spends in a subset of states \( \mathcal{R} \subset \mathcal{S} \) during a checkpointing cycle, given that \( Y_{k-1} = s_i \). Let \( M(t) \) be the accumulated reward during \( (0, t) \). It can be shown that [1]

\[
\lim_{t \to \infty} \frac{M(t)}{t} = \frac{1}{E[T_i]} \sum_{i=1}^{M} E[M_i] \beta_i
\]

where \( \beta_i \) is the \( i \)-th element of the steady state probability vector for the embedded Markov chain.

We now consider the long term fraction of time the system is available to process transactions, i.e. \( A = \lim_{t \to \infty} A(t)/t \), where \( A(t) \) is the total amount of time the system is available during \( (0, t) \). From (19) and since the total available time in a cycle is always equal to \( D \),

\[
A = \frac{D}{\sum_{i=1}^{M} E[T_i] \beta_i}
\]

where \( E[T_i] \) is given by equations (1), (2), (6) and (10) for the first model or (1), (2) and (18) for the second model.

It remains to calculate \( \beta \). For the first model, \( \beta \) is the steady state distribution vector for the embedded Markov chain \( \mathcal{Y} \). Since the total available period is a constant \( D \), we assumed that no transactions arrive during checkpointing roll back and recovery and that the system does not process any transactions until recovery is done, then process \( X(t) \) evolves only during the available periods. Note that, this is the same problem as to calculate the stationary distribution of an embedded Markov chain from process \( \mathcal{X} \) defined at points between constant length intervals. As a consequence, it is not difficult to see that the stationary distributions for both \( \mathcal{X} \) and \( \mathcal{Y} \) are the same, i.e. \( \beta = \pi \).

We need one more observation. We do not need to calculate \( E[T_i] \) above for each \( s_i \) and then use equation (19). Instead, \( \sum_{i=1}^{M} E[T_i] \beta_i \) can be obtained without performing the summation. This was observed before in [10] for the measures obtained in that paper, and the same approach is valid here. In order to obtain \( E[T_i] \), \( E[B_i] \) must be calculated. Let \( E[B_\beta] \) be the quantity obtained by (10) when we set the initial distribution \( \pi_0(0) \) to \( \beta \), i.e. \( \pi(0) = \beta \). Then it is not difficult to show that \( E[B_\beta] = \sum_{i=1}^{M} E[B_i] \beta_i \). But since \( \beta \) is the steady state distribution vector for \( \mathcal{X} \), then it is also the stationary distribution for the discrete time chain \( \mathcal{Z} \). Since chain \( \mathcal{Z} \) is started from the stationary distribution, \( \pi_u(j) = \beta_u \) for all \( j \) and any entry \( u \). From this observation we can write

\[
\sum_{j=0}^{n} \sum_{u \in B} \pi_u(j) = (n + 1) \rho
\]

where \( \rho = \sum_{u \in B} \pi_u \) and we recall that \( \pi_u \) is the \( u \)-th entry of the stationary distribution vector \( \pi \) for process \( \mathcal{X} \).
Substituting (21) into (10) and noting that \( \sum_{n=0}^{\infty} (n + 1)E_{n+2,\Lambda}(D) \) is half of the moment \( E[N(N - 1)] \), where \( N \) is a Poisson random variable with parameter \( \Lambda \) and \( D \), then \( \sum_{n=0}^{\infty} (n + 1)E_{n+2,\Lambda}(D) = \frac{(AD)^2}{2} \). Therefore, \( E[B_i] = \frac{D\rho}{2} \), and from (6)

\[
E[F_i] = \gamma_d D \left( t_r + \frac{D\rho}{2} \right).
\]

(22)

Note that, in equation (22), \( \gamma_d D \) is the average number of failures in the available interval, \( t_r \) is the time spent in each roll back operation (assumed to be constant) and the last term is the expected time spent with recoveries per failure. The final expression for the availability obtained from (20):

\[
A = \frac{D}{e^{\gamma_c C^*} - 1 + D + \gamma_d D \left( t_r + \frac{D\rho}{2} \right)}.
\]

(23)

Similar expression to (22) was obtained in [8] when it is assumed that the recovery time is proportional to the elapsed time from the end of the last checkpointing operation to the present failure. Here no such assumption is made and the development takes into account a queueing model of the transaction processing.

From (23) it is easy to calculate the optimum value of \( D (D_{opt}) \) to obtain the maximum availability, by taking the derivative of (23) with respect to \( D \).

For the second model model, \( \beta_i = 1 \) for \( i = 0 \) and zero otherwise, since the system always start from \( s_0 \) after a recovery or checkpointing. The availability is directly obtained from equations (1) and (18). Details are omitted for conciseness.

## 4 Examples

In this section we present an example to illustrate the influence of the model variables on the system availability, using the upper bound model. The resource contention model used in the example is a simple \( M/M/K \) queue with finite buffers (20 buffers). However, clearly more complex models could be used, and the impact on the solution cost depends on the size of the contention model. Note that the availability given by equation (23) depends on the contention model only through the long term fraction of time the system is busy and is given in a closed form since there is a closed form solution for the \( M/M/K \) queue. The time to perform a checkpointing operation if no failures occur during this period (\( C^* \)) is chosen equal to 2 hours, and the recovery time (\( t_r \)) is assumed to be negligible. The rate of failures during both the checkpointing operation as well as the available period are assumed to be identical.

Figure 2 shows the effect of the available interval length on the system availability. In that figure, the processing rate is assumed to be 4,000 transactions per hour and the transaction arrival rate varies from 1,000 to 3,500 transactions per hour. The failure rate is assumed to be 1 in 1,000 hours. The figure shows that the availability sharply decreases when \( D \) approaches 20–30 hours, independently of the system load. In this range the duration of the checkpointing operation
has a significant effect in the availability. Clearly there is an optimum value \( D \) that maximizes the availability, and checkpoints should be performed often if the system is heavily utilized. (For instance, if the transactions arrival rate is 3,500 then the optimum value of \( D \) should be around 65 hours, in contrast to \( D = 130 \) hours when the arrival rate is 1,000.) Furthermore, the sensitivity of the availability with respect to \( D \) decreases with the arrival rate of transactions, i.e. the smaller the arrival rate the smaller is the sensitivity of \( A \) with respect to \( D \) for values of \( D \) around the optimum value.

In Figure 3 we plot the optimum value of \( D \) versus the system load defined as the ratio of the transaction arrival rate to the service rate, for two values of the failure rate. It is interesting to notice that, in the two cases plotted, the optimum value sharply decreases when the load increases above a certain level. In both cases for a load above 0.3, the checkpointing has to be performed with much more frequency than for very small load values and, after this level, the changes in the optimum \( D \) value with respect to the load are less significant.

5 Summary

We have obtained simple expressions for evaluating the system availability for models of transaction oriented database systems subjected to checkpointing and recovery operations. Unlike previous work, our approach allows the use of a detailed model of the contention for the resources in the system. We do not need to assume that the recovery time when a failure occurs is proportional to the elapsed time since the last checkpointing operation. In our model, the recovery time is exactly the time spent processing transactions in that interval.

From the same basic development we obtain upper and lower bounds for the system availability.
Figure 3: Optimum value of $D$ versus the system load.

In one of the models, no transactions are allowed to arrive in the system when a recovery or checkpointing operations are performed. In the second model the effect of arrivals during the non-available periods are captured by assuming the system starts in an overload state after these operations.

Examples for the first model illustrate the use of the approach. Due to space limitation we do not show the curves for the lower bound model. However, we mention that the bounds are very tight in the example of Figure 2 and the results vary in the third decimal place. In future work we will present other examples to compare extensively the effect of arrivals during the non-available periods on the overall system availability.

References


