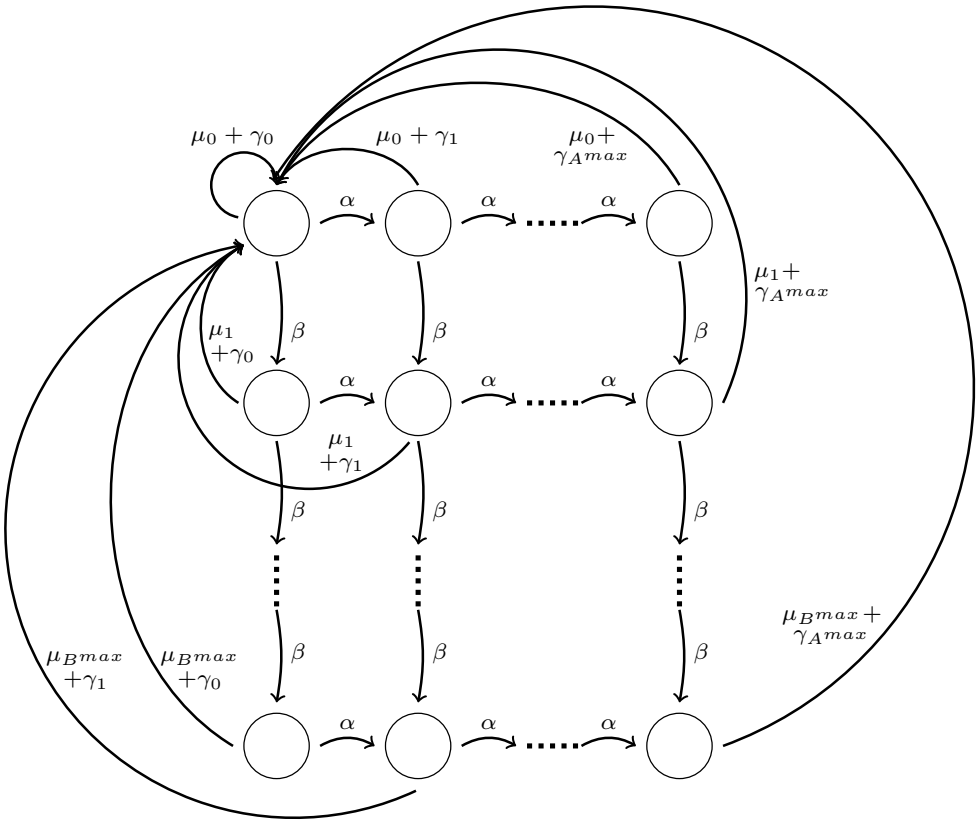
Stochastic Modelling for Backup Scheduling

Maaike Vollebergh

July 12, 2018

Bachelor thesis Mathematics

Supervisors: prof. dr. Michel Mandjes, Brendan Patch MSc.



Korteweg-de Vries Institute for Mathematics
Faculty of Sciences
University of Amsterdam

Abstract

More and more data is created annually and important data should be stored safely. To not lose files by a system failure, we need to make sure that backups are made regularly. However, more backups create more network traffic and therefore more network costs. In this thesis, this tradeoff between frequent backups and reducing network traffic is addressed. A stochastic model for this optimisation problem is designed and described, using backup and age classes. Users switch between backup and age classes and in each backup class and in each age class there are different rates at which backups are made and at which system failures occur, respectively. In addition, methods for performance evaluation are described and used to find the optimal solution for this model, where the expected quantity of lost files is minimized under the condition that no more than a certain maximum of bandwidth is used.

Title: Stochastic Modelling for Backup Scheduling
Authors: Maaike Vollebergh, maaike.vollebergh@student.uva.nl, 10721207
Supervisors: prof. dr. Michel Mandjes, Brendan Patch MSc.
Second grader: prof. dr. Sindo Nunez Queija,
End date: July 12, 2018

Korteweg-de Vries Institute for Mathematics
University of Amsterdam
Science Park 904, 1098 XH Amsterdam
`http://www.kdvi.uva.nl`

Contents

1. Introduction

The data that we create and copy annually will reach 44 zettabytes, or 44 trillion gigabytes, by 2020 [1]. In this world with more and more data we need to make sure that the important data is stored safely. Files on a single system can get lost in case of a failure of the system, theft, accidental deletion, et cetera. Because of the interruption on processes and the costs that losing files entails, it is essential to use central servers to regularly make backups of this data. On the other hand; data backups create a lot of network traffic, which makes the network costs higher. The question that arises here is: how can we schedule backups in a way that the least possible files get lost, but the network capacity is not exceeded?

To the best of our knowledge, there are three other studies that address this question. Van de Ven, Zhang & Schörgendorfer [2] introduced a model for distributed backup scheduling using backup probabilities. They had a decentralized approach, where the decision of when to initiate a backup is based on local information only. They focused on frequent backups and reducing network peak load. Claeys, Dorsman, Saxena, Walraevens & Bruneel [3] studied a threshold-based exhaustive policy for data backup scheduling: as soon as the backlog size reaches or exceeds some limit, a backup is initiated and an old backup is finished when all data is backed up. Xia, Machida & Trivedi [4] used a Markov decision process approach. They assume that after a failure the system is down for some time, because of the failure and the subsequent recovery. They minimize this system downtime while satisfying required levels of protection.

In essence, we use the same setting as in [2]. However, our contribution is that we also take system failures, where files get lost, into account and that we model bandwidth and system failures more explicitly. We use a decentralized approach, to avoid high communication costs. In this approach the system of each user keeps track of the times since the last backup or system failure and the amount of files that they produce. The time since the last backup or failure says something about the age of the system and the need to backup. The older a system is, the higher the chance of a system failure and the longer since a last backup or failure, the more files you have, so more need to backup. The rates at which backups are made are chosen by the system manager and depend on information about these times.

The remainder of this paper is organized as follows. In Section 2 we describe the model, Section 3 contains the performance evaluation method and in Section 4 the results are presented. Finally we will do some conclusions and the outlook in Section 5.

2. Model Description

We consider N users in a data network that utilise a common backup server. We will describe this model by introducing the concept of backup and age classes, so as to distinguish users by the age of their systems and their need to backup. We will define four processes: $M(\cdot)$ for the quantity of locally stored files per user, $L(\cdot)$ for the total quantity of lost files, $(A(\cdot), B(\cdot))$ for the backup and age classes users are in and $H(\cdot)$ for the bandwidth used.

Users in the data network produce files according to independent Poisson processes. We denote the rates at which user i produces files by λ_i. Users store their files locally between backup times and after a backup the quantity of locally stored files goes to zero. The time between backups follows a Coxian distribution with parameters chosen by the system manager. In addition, after backup we assume that users may experience a system failure, also after a Coxian distributed amount of time. The Coxian distribution is a phase-type distribution with different phases in sequence, where after each phase there is a probability of transitioning to the absorbing state [5]. In this model there are different phases with their own exponential distribution of backing up or experiencing a system failure, the backup and age classes. After each phase we can go to the absorbing state, where the quantity of locally stored files is zero. The parameters of the Coxian distribution for the time between system failures are dependent on uncontrollable system features. If a failure occurs before a backup, than the affected user loses all their files and the quantity of locally stored files also goes to zero. The evolution of the quantity of locally stored files of the users is tracked by the stochastic process $(M(t))_{t\geq 0}$ (with $M(t) \in \mathbb{N}_0^N$). We now introduce the notation needed to describe the evolution of $M(\cdot)$.

We allow the rate at which an individual user backs up their files at a particular time to depend on the time since their last backup or failure as follows. We suppose that as the time since the last backup or system failure gets longer, the need for the user to backup becomes greater. The reason for this is firstly that there are probably more locally stored files that can get lost, and secondly that the user is more likely to experience a system failure since the system of the user is older. We therefore suppose each user is in one of the backup classes $\{0, 1, \ldots, B^{\max}\}$ at any time. Users who have just backed up are placed in class 0. Except for class $B^{\max}$ users, the time spent in class j before advancing to class $j+1$ is $\text{Exp}(\beta)$ distributed, with $\beta \geq 0$. The backup rate classification of user n at time t is tracked in the nth coordinate of $B(t)$, $B(t) \in \{0, 1, \ldots, B^{\max}\}^N$. A user in class j backs up at rate $\mu_j \geq 0$, $j \in \{0, 1, \ldots, B^{\max}\}$, $\mu_0 \leq \mu_1 \leq \ldots \leq \mu_{B^{\max}}$. This backup policy clearly leads to a Coxian distributed amount of time between backups.

Similarly, the rate at which users experience a system failure at a particular time depends on the time since the last backup or system failure. If a user doesn't backup or doesn't experience a system failure, the system of the user is gradually getting older.

We suppose each user is in one of the age classes $\{0, 1, \ldots, A^{\max}\}$ at any time. The time spent in class j before advancing to class $j+1$, except for class $A^{\max}$ users, is $\text{Exp}(\alpha)$ distributed, with $\alpha \geq 0$. The age of the systems of user n at time t is tracked in the nth coordinate of $A(t)$, $A(t) \in \{0, 1, \ldots, A^{\max}\}^N$. A user in class j backs up at rate $\gamma_j \geq 0$, $j \in \{0, 1, \ldots, A^{\max}\}$, $\gamma_0 \leq \gamma_1 \leq \ldots \leq \gamma_{A^{\max}}$. This aging policy clearly leads to a Coxian distributed amount of time between system failures.

The total quantity of lost files for all users during $[0, t]$ is $L(t)$, with $L(t) \in \mathbb{N}_0$. When user n experiences a system failure, which happens at a rate that depends on the background process $(A(\cdot), B(\cdot))$, the quantity of locally stored files in the nth coordinate of $M(\cdot)$ is added by $L(\cdot)$.

The state space of this process is

$$\{(m, l, a, b) : m \in \mathbb{N}_0^N, l \in \mathbb{N}_0, a \in \{0, 1, \ldots, A^{\max}\}^N, b \in \{0, 1, \ldots, B^{\max}\}^N\},$$

with the different transitions and corresponding rates for one user as given in Table 2.1, where m_i, a_i and b_i are defined as the ith coordinate of m, a and b, respectively. A representation of this model for one user can be seen in Figure 2.1.

Transition	Rate	States
$(m_i, l, a_i, b_i) \to (m_i + 1, l, a_i, b_i)$	λ_i	
$(m_i, l, a_i, b_i) \to (0, l + m_i, 0, 0)$	γ_j	$a_i = j$, $j \leq A^{\max}$
$(m_i, l, a_i, b_i) \to (0, l, 0, 0)$	μ_k	$b_i = k$, $k \leq B^{\max}$
$(m_i, l, a_i, b_i) \to (m_i, l, a_i + 1, b_i)$	α	$a_i = 0, 1, \ldots, A^{\max} - 1$
$(m_i, l, a_i, b_i) \to (m_i, l, a_i, b_i + 1)$	β	$b_i = 0, 1, \ldots, B^{\max} - 1$

Table 2.1.: Transition rates for user i.

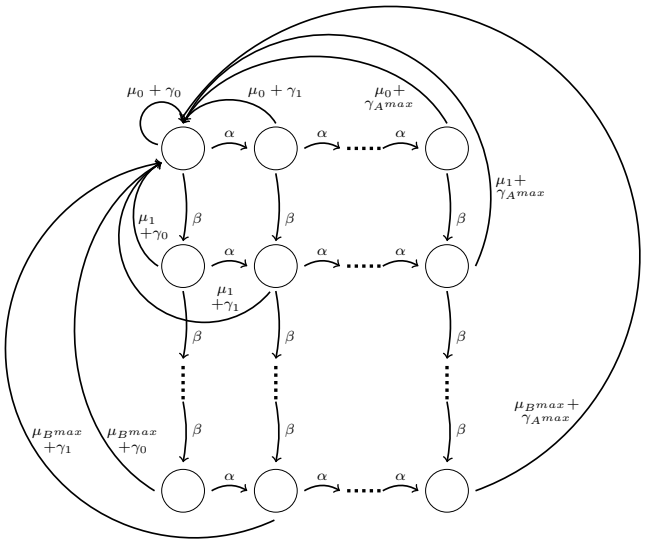


Figure 2.1.: Representation of the model for one user.

Transition	Rate	States
$x \to x + e_{(jA^{\max}+k+2)} - e_{(jA^{\max}+k+1)}$	$X_{j,k}(t)\alpha$	$j \leq B^{\max}$; $k \leq A^{\max}$
$x \to x + e_{((j+1)A^{\max}+k+1)} - e_{(jA^{\max}+k+1)}$	$X_{j,k}(t)\beta$	$j \leq B^{\max}$; $k \leq A^{\max}$
$x \to x + e_1 - e_{(jA^{\max}+k+1)}$	$X_{j,k}(t)(\mu_j + \gamma_k)$	$j \leq B^{\max}$; $k \leq A^{\max}$

Table 2.2.: Transition rates of the background chain at time t.

We introduced the backup and age classes tracked by $(A(\cdot), B(\cdot))$. The backup classes that users are in influence the bandwidth used at any particular time: making more backups means that there is more bandwidth used. To keep track of the bandwidth used at time t, we use the variable $H(t)$. Let H be the stationary distribution of the process $H(\cdot)$, with $h_0 > 0$ as the restriction on the bandwidth. The stationary distribution H is determined by the process $(X(t))_{t\geq 0}$ (with $X(t) \in \mathbb{N}_0^{(B^{\max}+1)(A^{\max}+1)}$). User i has background process $(X^i(t))_{t\geq 0}$ (with $X^i(t) \in \mathcal{J} := \{(0,0),(0,1),\ldots,(0,A^{\max}),(1,0),\ldots,(B^{\max},A^{\max})\}$, that keeps track of the backup and age classes that user i is in. There are $(B^{\max}+1)\cdot(A^{\max}+1)$ different states a user can be in, based on different combinations of backup and age classes. The process $X(\cdot)$ keeps track of the quantity of users in each state, since the quantity of users in each state is a measure for the bandwidth used. Let, for $j = 0,1,\ldots,B^{\max}$ and $k = 0,1,\ldots,A^{\max}$, $X_{jk}(t) := \sum_{i=1}^{N} \mathbb{1}_{X^i(t)=(j,k)}$ at time t, with $\sum_{j=0}^{B^{\max}} \sum_{k=0}^{A^{\max}} X_{jk}(t) = N$ for every $t \in \mathbb{R}_+$. Then the process $X(\cdot)$ is denoted by:

$$x = \begin{pmatrix} X_{00}(t) \\ X_{01}(t) \\ \vdots \\ X_{0A^{\max}}(t) \\ X_{10}(t) \\ \vdots \\ X_{A^{\max}B^{\max}}(t) \end{pmatrix}.$$

The transition rates of the background chain $X(\cdot)$ can be found in Table 2.2, where e_i is defined as the ith unit vector in $\mathbb{N}_0^{(B^{\max}+1)(A^{\max}+1)}$.

The used bandwidth, summarised in the long term by H, and the total quantity of lost files, tracked by $L(\cdot)$, are two important substitutable quantities in this model. To find the optimal trade-off between those two, we define an optimisation problem. The objective function is:

$$r(\beta, \boldsymbol{\mu}) = \lim_{t\to\infty} \frac{\mathbb{E}L(t)}{t},$$

with $\boldsymbol{\mu} = (\mu_j)_{j=1}^{B^{\max}}$. We are using $\frac{\mathbb{E}L(t)}{t}$ in the objective function, because we are interested in how $\mathbb{E}L(t)$ evolves in the long run.

To influence this objective function we can choose the parameters β and $\boldsymbol{\mu}$. The optimisation problem then is:

$$\begin{aligned} &\min_{\beta,\boldsymbol{\mu}} \; r(\beta, \boldsymbol{\mu}) \\ &\quad \text{s.t. } \mathbb{P}(H > h_0) < \epsilon. \end{aligned}$$

It would be possible to have each file that gets lost associated with a random cost to the system manager, say $\theta \in \mathbb{R}_+$. Using Walds equation [7], where we have independent and indentically distributed random variables (the costs per file) and one random variable (the quantity of lost files) that is independent from the other variables, the new objective function can be defined.

There are two different approaches of this model: the decentralized and the centralized approach. The difference between those two is that in the decentralized approach we choose the parameters β and $\boldsymbol{\mu}$ at time 0, while in the centralized approach we periodically choose parameters β and $\boldsymbol{\mu}$. We will use the decentralized approach.

3. Performance Evaluation Method

In this chapter we will describe the method for the performance evaluation. For the performance evaluation of this model we will use $B^{\max} = 1$, $A^{\max} = 0$. The background process $X^i(\cdot)$ of one user then has $(B^{\max} + 1) \cdot (A^{\max} + 1) = 2$ states, one if the rate is μ_0 (say $X^i(t) = 1$) and one if the rate is μ_1 (say $X^i(t) = 2$).

We will first evaluate a system with one user. Because we work with expectations and expectations are linear, we can multiply the outcomes of a one-user-system and get outcomes for multiple users. A representation of the one-user-system can be seen in Figure 3.1.

3.1. Encompassing Performance Evaluation Model

Fiems, Mandjes and Patch address in [6] network models where, instead of clients joining and leaving nodes one by one, the full population at an individual node may also move around the network. These transitions are called *multiplicative transitions*, by which the network population vector, say m, is multiplied by a transition matrix A. After the transition the network population becomes Am. We are using the method that Fiems et al. describe in [6].

In our one user model the network population vector $m \in \mathbb{N}_0^2$ contains two values: the quantity of locally stored files ($M(\cdot)$) in the first coordinate and the quantity of lost files ($L(\cdot)$) in the second coordinate. We define $\mathcal{K}_{ij}$ as the number of transitions from state i to state j, so that $\mathcal{K}_{11} = \{1, 2\}$, $\mathcal{K}_{12} = \mathcal{K}_{21} = \{1\}$. For each $k \in \mathcal{K}_{ij}$ we define a transition matrix, $A_{ij}^{(k)}$, a (2×2)-matrix with entries in $\mathbb{N}_0$. The rate $\alpha_{ij}^{(k)} \geq 0$ is the rate at which the multiplicative transition from m to $A_{ij}^{(k)} m$ takes place. The rates and corresponding matrices are:

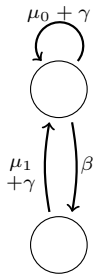


Figure 3.1.: Representation of the model with $B^{\max} = 2$, $A^{\max} = 1$.

- The first rate is
$$\alpha_{11}^{(1)} = \mu_0.$$
Here files are backed up, so the locally stored files leave the system. The corresponding matrix is
$$A_{11}^{(1)} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

- The second rate is
$$\alpha_{11}^{(2)} = \gamma.$$
Here the user experiences a system failure, so the locally stored files move to the lost files. The corresponding matrix is
$$A_{11}^{(2)} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}.$$

- The next rate is
$$\alpha_{12}^{(1)} = \beta.$$
Here the user moves to the next backup class, so nothing happens to their files. The corresponding matrix is
$$A_{12}^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

- The next rate is
$$\alpha_{21}^{(1)} = \mu_1.$$
Here files are backed up again, so the locally stored files leave the system. The corresponding matrix is
$$A_{21}^{(1)} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

- The last rate is
$$\alpha_{21}^{(2)} = \gamma.$$
Here the user experiences a system failure again, so the locally stored files move to the lost files. The corresponding matrix is
$$A_{21}^{(2)} = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}.$$

The objective function of our optimisation problem is $r(\beta, \boldsymbol{\mu}) = \lim_{t\to\infty} \frac{\mathbb{E}L(t)}{t}$. To compute $\mathbb{E}L(t)$ for one user, following the method in [6], we define
$$\bar{E} = \begin{bmatrix} \mathbb{E}(M(t)I_1) \\ \mathbb{E}(L(t)I_1) \\ \mathbb{E}(M(t)I_2) \\ \mathbb{E}(L(t)I_2) \end{bmatrix},$$

with $I_i(t) = \mathbb{1}_{X^1(t)=i}$. Evaluation of $\langle \rho, \bar{E}(T) \rangle$, where $\rho = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}^T$ will then give us $\mathbb{E}L(t)$ for one user. To compute $\bar{E}(T)$ we first define the following rates:

$\lambda_n^{(i)}$ external arrival rate at queue n when $X^1(t) = i$;

$\mu_{nn'}^{(i)}$ departure rate for a single file from queue n to queue n' when $X^1(t) = i$.

In our model these rates are, for every $i \in \{1, 2\}$,

$$\lambda_n^{(i)} = \begin{cases} \lambda & n = 1 \\ 0 & n = 2 \end{cases}$$
$$\mu_{nn'}^{(i)} = 0.$$

Then to compute $\bar{E}(T)$ we define the following terms:

- The matrix $\mathcal{M}$ that depends on the rates $\mu_{nn'}^{(i)}$. Since all the rates $\mu_{nn'}^{(i)}$ are zero in our case, the matrix $\mathcal{M}$ is also zero.
- The vector
$$\mathcal{L}_i := [\lambda_n^{(i)}]_{n=1}^2 = \begin{bmatrix} \lambda \\ 0 \end{bmatrix}$$
with the external arrival rates in queue 1 and queue 2 when $X(\cdot) = 1, 2$;
- For i,$j \in \{1, 2\}$, $i \neq j$, the matrices
$$\mathcal{A}_{ij} := \sum_{k=1}^{K_{ij}} \alpha_{ij}^{(k)} A_{ij}^{(k)}, \quad \mathcal{A}_{ii} := \sum_{k=1}^{K_{ii}} \alpha_{ii}^{(k)} A_{ii}^{(k)} - \bar{\alpha}_i \mathbb{I}_2,$$
where $\mathbb{I}_2$ is the (2×2)-dimensional identity matrix, and with
$$\bar{\alpha}_i := \sum_{j=1}^{2} \sum_{k=1}^{K_{ij}} \alpha_{ij}^{(k)}.$$

The terms $\mathcal{L}_i$, $\mathcal{A}_{ij}$ and $\mathcal{A}_{ii}$ are dependent on the state of the background process. We are combining these terms in the following matrices

$$\mathcal{L} := \operatorname{diag}([\mathcal{L}_i]_{i=1}^2) = \begin{bmatrix} \lambda & 0 \\ 0 & 0 \\ 0 & \lambda \\ 0 & 0 \end{bmatrix};$$
$$\mathcal{A} := [\mathcal{A}_{ji}]_{i,j=1}^2,$$
$$= \begin{bmatrix} -\mu_0 - \gamma - \beta & 0 & 0 & 0 \\ \gamma & -\beta & \gamma & \mu_1 + \gamma \\ \beta & 0 & -\mu_1 - \gamma & 0 \\ 0 & \beta & 0 & -\mu_1 - \gamma \end{bmatrix}.$$

We also define the generator matrix $\bar{\mathcal{A}} := [\bar{\alpha}_{ij}]_{i,j=1}^{2}$, with for $i,j \in \{1,2\}$, $i \neq j$,

$$\bar{\alpha}_{ij} = \sum_{k=1}^{K_{ij}} \alpha_{ij}^{(k)}, \quad \bar{\alpha}_{ii} = -\sum_{i' \neq i} \bar{\alpha}_{ii'}.$$

In our case this is

$$\bar{\mathcal{A}} = \begin{bmatrix} -\beta & \beta \\ \mu_1 + \gamma & -(\mu_1 + \gamma) \end{bmatrix}$$

Then, finally, we define

$$\begin{aligned} \mathcal{C} :&= \begin{bmatrix} \mathcal{M} + \mathcal{A} & \mathcal{L} \\ \mathbb{O}_{2,4} & \bar{\mathcal{A}}^T \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{A} & \mathcal{L} \\ \mathbb{O}_{2,4} & \bar{\mathcal{A}}^T \end{bmatrix}. \end{aligned}$$

with $\mathbb{O}_{I,J}$ defined as an all-zeros matrix of dimension $I \times J$.

From Lemma 1 from [6] we have the following expression for $\bar{E}(T)$:

$$\bar{E}(T) = e^{(\mathcal{M}+\mathcal{A})T}\bar{E}(0) + [\mathbb{I}_J, \mathbb{O}_{J,I}] \cdot e^{\mathcal{C}T} \cdot \begin{bmatrix} \mathbb{O}_{J,I} \\ \mathbb{I}_I \end{bmatrix} \pi(0),$$

with I the number of background states, $J := I \cdot N$ and $\pi(t)$ the transient distribution vector of the background process. This means $\pi(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Then in our case $\bar{E}(T)$ is:

$$\bar{E}(T) = e^{\mathcal{A}T}\bar{E}(0) + [\mathbb{I}_4, \mathbb{O}_{4,2}] \cdot e^{\mathcal{C}T} \cdot \begin{bmatrix} \mathbb{O}_{4,2} \\ \mathbb{I}_2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Evaluation of $\langle \rho, \bar{E}(T) \rangle$ then gives us the expected quantity of lost files for one user, $\mathbb{E}L(t)$. If we multiply this quantity with the quantity of users, say N, we get the expected quantity of lost files for N users.

Recall that our optimisation problem is

$$\begin{aligned} &\min_{\beta, \boldsymbol{\mu}} \; r(\beta, \boldsymbol{\mu}) \\ &\quad \text{s.t. } \mathbb{P}(H > h_0) < \epsilon. \end{aligned}$$

Now that we defined the encompassing performance evaluation, we can use it to evaluate the objective function and the constraint of our optimisation problem.

3.2. Evaluation of the Objective Function

To find an expression for the objective function $r(\beta, \boldsymbol{\mu})$ we look at the expression for $\bar{E}(T)$,

$$\bar{E}(T) = e^{\mathcal{A}T}\bar{E}(0) + [\mathbb{I}_4, \mathbb{O}_{4,2}] \cdot e^{\mathcal{C}T} \cdot \begin{bmatrix} \mathbb{O}_{4,2} \\ \mathbb{I}_2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

By [6] the second term reads $\bar{B}(T) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, such that

$$\begin{aligned} [\mathbb{I}_4, \mathbb{O}_{4,2}] \cdot e^{\mathcal{C}T} \cdot \begin{bmatrix} \mathbb{O}_{4,2} \\ \mathbb{I}_2 \end{bmatrix} &= \int_0^T e^{\mathcal{A}(T-s)} \cdot \mathcal{L} \cdot e^{\bar{\mathcal{A}}^T s} ds \\ &:= \bar{B}(T). \end{aligned}$$

Now consider the spectral decompositions of the matrices $\mathcal{A}$ and $\bar{\mathcal{A}}^T$, for diagonal matrices D_1 and D_2,

$$\mathcal{A} = W D_1 W^{-1} \qquad \bar{\mathcal{A}}^T = V D_2 V^{-1}.$$

Then, because the matrices W and V are invertible,

$$\begin{aligned} \bar{B}(T) &= \int_0^T e^{\mathcal{A}(T-s)} \cdot \mathcal{L} \cdot e^{\bar{\mathcal{A}}^T s} ds \\ &= \int_0^T W e^{D_1(T-s)} W^{-1} \cdot \mathcal{L} \cdot V e^{D_2 s} V^{-1} ds. \end{aligned}$$

Now let $\psi_{lnj} = [W]_{ln}[W]^{-1}_{nj}$, $\phi_{imk} = [V]_{im}[V]^{-1}_{mk}$ and $d_{in} = [D_i]_{nn}$. Then we have for $\bar{B}(T)$, using the row-column rule for multiplication, with $S = \{1, \ldots, 4\} \times \{1, \ldots, 2\}$,

$$\begin{aligned} &[\bar{B}(t)]_{ij} = \\ &\sum_{(k,l)\in S} \Big[[\mathcal{L}]_{kl} \sum_{(m,n)\in S} \phi_{lnj}\psi_{imk} \Big\{ \mathbb{1}_{\{d_{1m}=0, d_{2n}=0\}} t \\ &+ \mathbb{1}_{\{d_{1m}\neq 0, d_{2n}\neq 0\}} (e^{d_{2n}t} - e^{d_{1m}t})(d_{2n} - d_{1m})^{-1} \\ &+ \mathbb{1}_{\{d_{1m}=0, d_{2n}\neq 0\}} (e^{d_{2n}t} d_{2n}^{-1} - d_{2n}^{-1}) \\ &+ \mathbb{1}_{\{d_{1m}\neq 0, d_{2n}=0\}} (e^{d_{1m}t} d_{1m}^{-1} - d_{1m}^{-1}) \Big\} \Big]. \end{aligned} \tag{3.1}$$

Since the matrix $\mathcal{A}$ has non-positive column sums, the eigenvalues of $\mathcal{A}^T$, and so those of $\mathcal{A}$, either equal zero or have a negative real part. The matrix $\bar{\mathcal{A}}$ also has non-positive row sums, so its eigenvalues, and so those of $\bar{\mathcal{A}}^T$, also either equal zero or have a negative real part. Since the eigenvalues of both matrices $\mathcal{A}$ and $\bar{\mathcal{A}}^T$ either equal zero or have a negative real part we can split Equation 3.1 in parts that go to zero as time goes to infinity and parts that does not, so that we can identify numbers η and ζ such that, as $t \to \infty$,

$$\mathbb{E}L(t) = \eta t + \zeta + o(1),$$

with $o(1) \to 0$ as $t \to \infty$. Here η is determined only from eigenvalues corresponding to zero eigenvalues. In this way it does not depend on the initial position of the system. This means that we found the following expression for our objective function

$$\begin{aligned} r(\beta, \boldsymbol{\mu}) &= \lim_{t\to\infty} \frac{\mathbb{E}L(t)}{t} \\ &= \lim_{t\to\infty} \frac{\eta t + \zeta + o(1)}{t} \\ &= \eta. \end{aligned}$$

3.3. Evaluation of the Constraint

To evaluate the constraint of our optimisation problem, which is

$$\mathbb{P}(H > h_0) < \epsilon,$$

with h_0 the limit of data, we evaluate the background process. In this system, as in Figure 3.1, the background process X is distributed as follows:

$$X \sim \text{Bin}(N, \frac{\mu_1 + \gamma}{\mu_1 + \gamma + \beta}). \tag{3.2}$$

The bandwidth then is

$$H = X\mu_0 + (N - X)\mu_1 + \delta,$$

with δ the background rate. The expression for the constraint in our optimisation problem then is

$$\begin{aligned} \mathbb{P}(H > h_0) &= 1 - \mathbb{P}(H \leq h_0) \\ &= 1 - \mathbb{P}(X\mu_0 + (N - X)\mu_1 + \delta \leq h_0) \\ &= 1 - \mathbb{P}(X(\mu_0 - \mu_1) \leq h_0 - \delta - N\mu_1) \\ &= 1 - \mathbb{P}(X > \frac{h_0 - \delta - N\mu_1}{\mu_0 - \mu_1}), \text{ because } \mu_0 - \mu_1 \leq 0 \\ &= \mathbb{P}(X \leq \frac{h_0 - \delta - N\mu_1}{\mu_0 - \mu_1}) \\ &= \sum_{i=0}^{\lfloor \frac{h_0 - \delta - N\mu_1}{\mu_0 - \mu_1} \rfloor} \binom{N}{i} \Big(\frac{\mu_1 + \gamma}{\mu_1 + \gamma + \beta}\Big)^i \Big(1 - \frac{\mu_1 + \gamma}{\mu_1 + \gamma + \beta}\Big)^{N-i} \\ &= \sum_{i=0}^{\lfloor \frac{h_0 - \delta - N\mu_1}{\mu_0 - \mu_1} \rfloor} \binom{N}{i} \Big(\frac{\mu_1 + \gamma}{\mu_1 + \gamma + \beta}\Big)^i \Big(\frac{\beta}{\mu_1 + \gamma + \beta}\Big)^{N-i} \\ &< \epsilon. \end{aligned}$$

4. Results

In this section we will investigate the sensitivity of the optimisation problem in terms of the different parameters that influence the objective function: β, the rate at which users advance to the next backup class; μ_0, the rate at which backups are made when a user is in backup class 0; and μ_1, the rate at which backups are made when a user is in backup class 1. With the expressions found in Sections 3.2 and 3.3 we can visualise this optimisation problem. From now on we will give our model the properties as indicated in Table 4.1.

First we will have a look at the expectation of the quantity of lost files over time. In Figure 4.1 we see the expectation of the quantity of lost files with different initial conditions. The initial condition tells us how many locally stored files there are at time 0, $M(0)$. It can be seen that asymptotically the different functions have the same constant derivative, which is the same as the derivative of the linear approximations: the objective function of our optimisation problem.

We will now investigate the effect of the parameters β, μ_0 and μ_1 on this derivative and the used bandwidth. In Figures 4.2a, 4.3a and 4.4a the probability that the bandwidth is higher than the allowed level is shown as a function of the different parameters. This probability should to be smaller than ϵ according to the constraint of our optimisation problem. This maximum and the corresponding maximums of the parameter are also indicated in Figures 4.2a, 4.3a and 4.4a. As the parameters increase, the more frequently backups are made and the more bandwidth is used. That is why we would expect that, as in Figure 4.2a, the function of the probability that the bandwidth is higher than the allowed level would be increasing. That this is not the case for Figures 4.3a and 4.4a

Property	Symbol	Value
Number of users	N	100
Arrival rate	λ	5
Failure rate	γ	0.5
Background rate	δ	1
Maximum of bandwidth	h_0	150
Constraint for the probability that the bandwidth is higher than the maximum	ϵ	0.05

Table 4.1.: Properties of the system.

can be explained by the function itself,

$$\mathbb{P}(H > h_0) = \sum_{i=0}^{\lfloor \frac{h_0-\delta-N\mu_1}{\mu_0-\mu_1} \rfloor} \binom{N}{i} \Big(\frac{\mu_1+\gamma}{\mu_1+\gamma+\beta}\Big)^i \Big(\frac{\beta}{\mu_1+\gamma+\beta}\Big)^{N-i}.$$

If the backup rate μ_0 increases a little bit, $\lfloor \frac{h_0-\delta-N\mu_1}{\mu_0-\mu_1} \rfloor$ remains the same integer and the outcome does not change. That is why in Figure 4.3a the graph is constant on some intervals. If the backup rate μ_1 increases a little bit, $\lfloor \frac{h_0-\delta-N\mu_1}{\mu_0-\mu_1} \rfloor$ also does not change, but $\frac{\mu_1+\gamma}{\mu_1+\gamma+\beta}$ increases. This is the second parameter of the Binomial distribution defined in Definition 3.2. If this parameter increases, the probability to be smaller than some value will decrease. That is why in Figure 4.3a the graph is decreasing on some intervals.

In Figures 4.2b, 4.3b and 4.4b the derivative of the expected quantity of losses is shown, also as a function of the different parameters. Because increasing the parameters increases the amount of backups made, the graphs of the derivatives of the expected quantity of losses are decreasing. The maximums of the parameters β, μ_0 or μ_1 from Figures 4.2a, 4.3a and 4.4a, respectively, are also indicated in Figures 4.2b, 4.3b and 4.4b. Because the graphs of the derivatives of the expected quantity of losses are decreasing and we are minimizing this derivative, these maximums are the optimal values of that parameter in each setting, respectively.

By renormalizing time such that $\beta = 1$, we only have the parameters μ_0 and μ_1 to optimize on. In Figure 4.5 the probability that the bandwidth is higher than the allowed level, as a function of the backup rates μ_0 and μ_1, can be seen. We see that this probability is increasing as the backup rates increase. In this way the acceptable region can be seen, namely where the probability is less then ϵ. In Figure 4.6 we see the expected quantity of lost files decreasing as μ_0 and μ_1 are increasing. This shows us again that the optimal point should be where the probability is as close as possible to ϵ, at the boundary of the acceptable region that we know from Figure 4.5. Running the optimisation program gives the following optimal values:

$$\mu_0 = 1.1453;$$
$$\mu_1 = 2.1301.$$

This optimal point is indicated in Figures 4.5 and 4.6 and this is where we expected it to be.

The optimal point does not only depend on the parameters that we can influence, but also on the properties that we gave the system. If we, for example, change the failure rate γ, the optimal values of the backup rates μ_0 and μ_1 also change. This can be seen in Figure 4.7. We see especially that the backup rate μ_1 is increasing, to compensate for the higher failure rate.

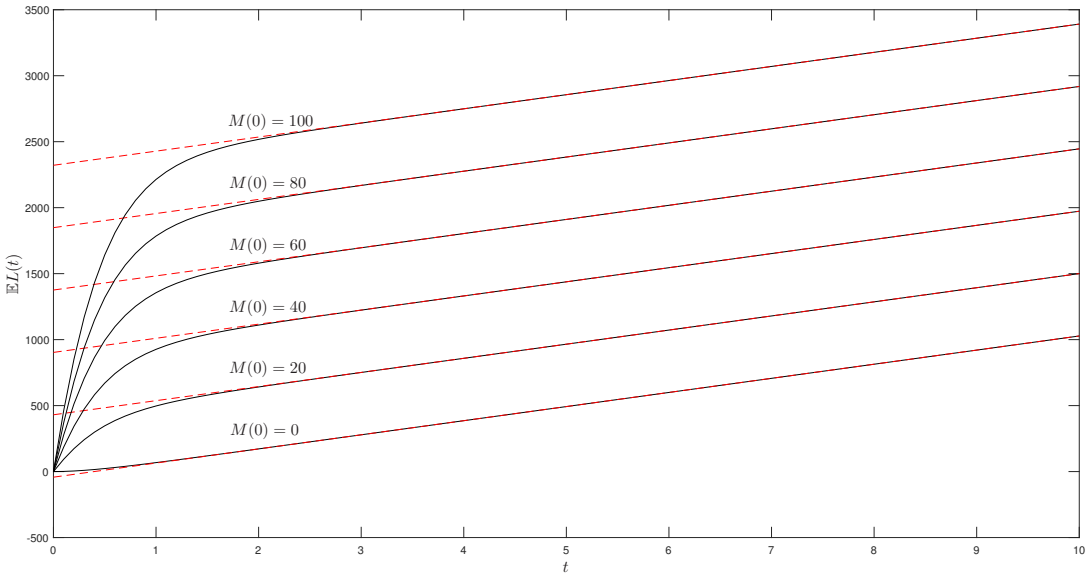
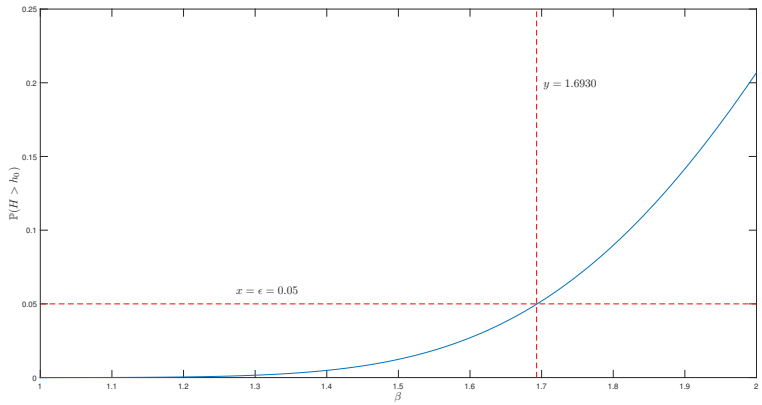
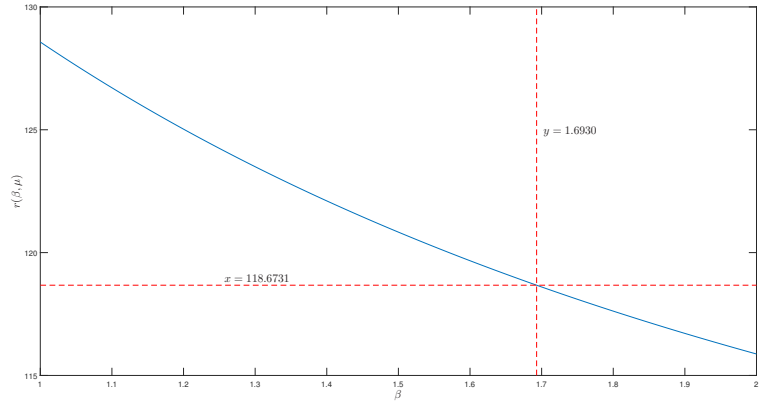


Figure 4.1.: The expectation of the quantity of lost files over time with different initial conditions of the quantity of locally stored files and the linear approximations, with $\beta = 4$, $\mu_0 = 1$ and $\mu_1 = 2$.

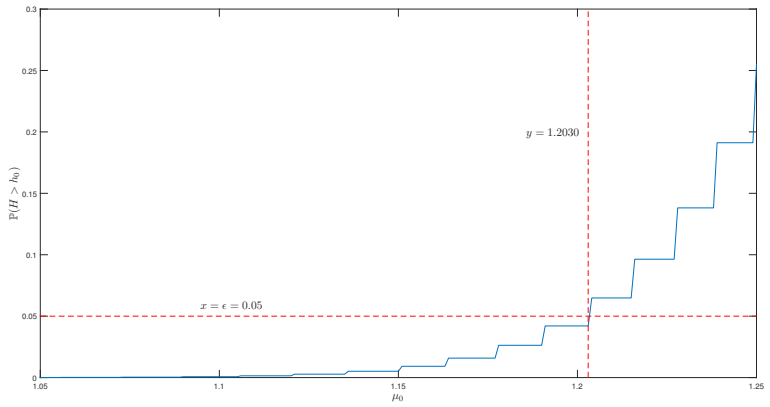


(a)

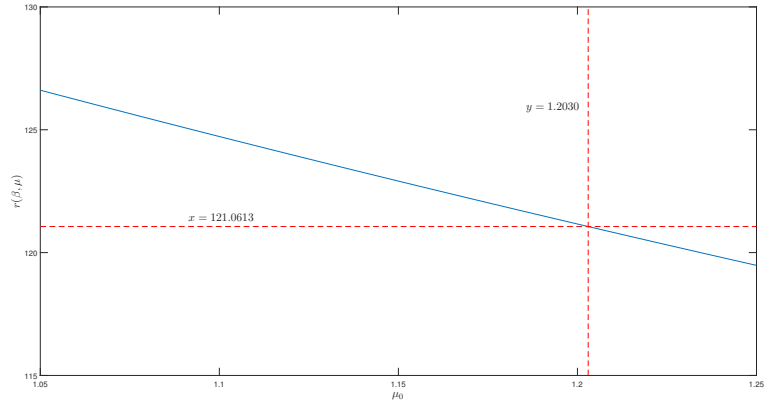


(b)

Figure 4.2.: The probability that the bandwidth is higher than the allowed level (a) and the derivative of the expected quantity of losses (b), both as a function of the rate β at which users advance to the next backup class, with $\mu_0 = 1$ and $\mu_1 = 2$.

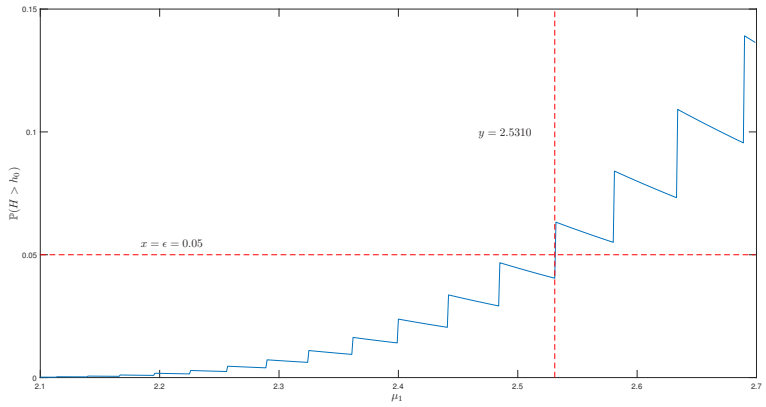


(a)

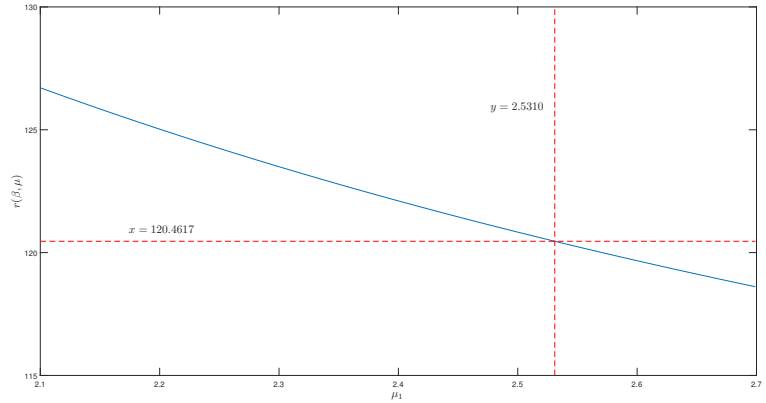


(b)

Figure 4.3.: The probability that the bandwidth is higher than the allowed level (a) and the derivative of the expected quantity of losses (b), both as a function of the rate μ_0 at which backups are made when a user is in backup class 0, with $\beta = 1$ and $\mu_1 = 2$.



(a)



(b)

Figure 4.4.: The probability that the bandwidth is higher than the allowed level (a) and the derivative of the expected quantity of losses (b), both as a function of the rate μ_1 at which backups are made when a user is in backup class 1, with $\beta = 1 and \mu_0 = 1$.

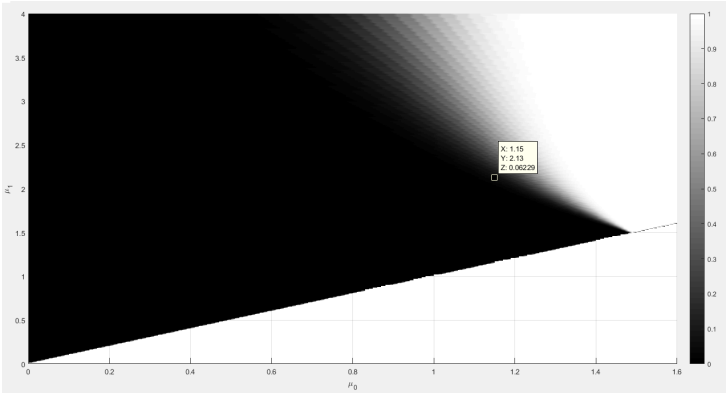


Figure 4.5.: Heatmap of the probability that the bandwidth is higher than the allowed level as a function of the backup rates μ_0 and μ_1.

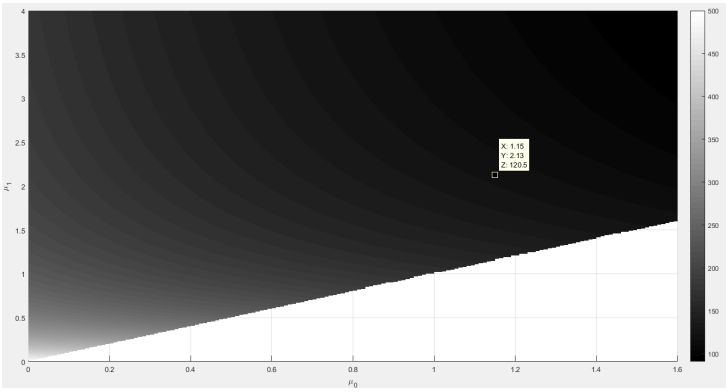


Figure 4.6.: Heatmap of the objective function as a function of the backup rates μ_0 and μ_1.

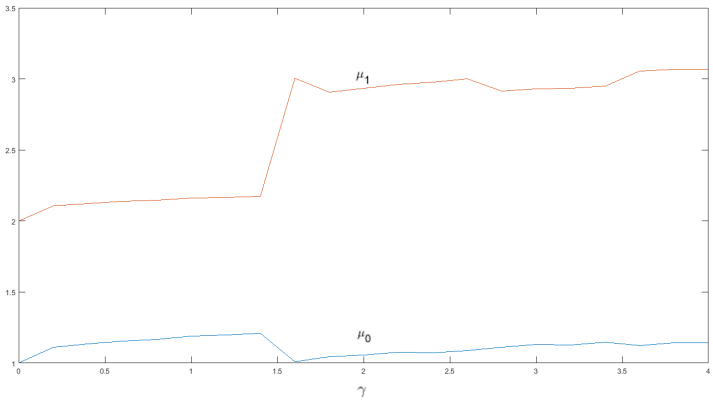


Figure 4.7.: Optimal values of the backup parameters μ_0 and μ_1 as a function of the failure rate γ.

5. Conclusion

In this thesis a model is designed and described for a system where many users in a data network utilise a common backup server. Using the method in [6] a performance evaluation method for this model is developed. The sensitivity of the optimisation problem in terms of the different parameters that influence the objective function is investigated.

There are three main aspects of this thesis that could be expanded on in future research. First, we assumed that the time between system failures follow a Coxian distribution. Relaxing this assumption could be interesting for future research. Second, this thesis only analysed the model with only two different states using the Binomial distribution. In future research, the performance evaluation method could be extended using the Multinomial distribution in order to analyse a version of the model with more states. This would allow for a more accurate representation of failure and more flexibility in backups. Third, this thesis presented a decentralized model, the tractable performance measures of which do not capture the effect of the system state at the moment at which the decision of backing up is made. In order to design a centralized model for this system, a new performance evaluation method should be defined. Defining such a centralized model and comparing it to the decentralized one to see which one works better could be an interesting topic for future research.

Popular Summary

We live in a world with a growing number of computers, laptops, phones, etcetera that create more and more data. A lot of files that we create are important and need to be stored safely; this is what backups are used for. When a backup is made, files are copied to a central server through an internet connection. This happens, for example, when you upload files to Dropbox. The system that we discuss in this thesis contains a very large number of users and one backup server. For this system we are designing a backup scheduling model and evaluating its performance.

In this system two things can happen: a single user can make a backup or can experience a system failure. If a user makes a backup, the internet connection is used. One of the properties of the internet connection is the bandwidth. This says something about how much information the internet connection can transfer at a time. The more files are backed up simultaneously, the more bandwidth is needed. The bandwidth that is necessary depends on the peak load of information. In the backup scheduling model, we therefore want to minimize the peak load. In this thesis we define a maximum of bandwidth and we look at the probability that the bandwidth exceeds this maximum amount.

Apart from backups, a user can experience a system failure. A 'system failure' is the general term for something that causes the user to lose its locally stored files. This is something that we want to avoid, since losing files can entail costs. In this thesis we are looking at how many files are expected to get lost, because this is something that we can compute.

This system involves an optimisation problem. An optimisation problem contains an objective function, which tells us what we want to minimize, and a constraint, which tells us what criterias the system should meet. Our optimisation problem is to minimize the expected quantity of lost files (the objective function), while keeping the probability that we exceed the maximum of bandwidth as low as possible (the constraint).

To model this optimisation problem we keep track of the quantity of locally stored files for every user and the total quantity of lost files. When a user makes a backup, the quantity of locally stored files of that user becomes zero. When a user experiences a system failure, the quantity of locally stored files of that user is added to the total quantity of lost files and the quantity of locally stored files becomes zero.

We define backup and age classes and we suppose that each user is in one of the backup and one of the age classes at any time. As time since the last backup or system failure passes, users will advance to the next backup or age class. The higher the age class, the older the system and, thus, the higher the rate at which a system failure occurs. These rates and the frequency at which users move to the next age class cannot be influenced. The higher the backup class, the more locally stored files there will probably

be and, thus, the higher the rate at which a user makes a backup. These rates and the frequency at which users move to the next backup class can be influenced. These are exactly the parameters that we can choose to influence our objective function. Using these parameters we minimize the expected quantity of lost files, while keeping the probability that we exceed the maximum of bandwidth as low as possible. In that way we can get an optimal way to schedule our backups.

This thesis describes the aforementioned model more mathematically and presents a method to evaluate the model's performance. This method is used to write an optimisation program that visualizes the results of this backup scheduling model.

Bibliography

[1] EMC Digital Universe with Research & Analysis by ICD. (2014). The digital universe of opportunities: rich data and the increasing value of the internet of things. http://www.emc.com/leadership/ digital-universe/2014iview/executive-summary.htm. Accessed 1 May 2018.

[2] van de Ven, P. M., Zhang, B., & Schorgendorfer, A. (2014, April). Distributed backup scheduling: Modeling and optimization. In INFOCOM, 2014 Proceedings IEEE (pp. 1644-1652). IEEE.

[3] Claeys, D., Dorsman, J. P., Saxena, A., Walraevens, J., & Bruneel, H. (2017, July). A queueing-theoretic analysis of the threshold-based exhaustive data-backup scheduling policy. In AIP Conference Proceedings (Vol. 1863, No. 1, p. 200002). AIP Publishing.

[4] Xia, R., Machida, F., & Trivedi, K. (2014, June). A Markov decision process approach for optimal data backup scheduling. In Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on (pp. 660-665). IEEE.

[5] Buchholz, P., Kriege, J., & Felko, I. (2014). Input modeling with phase-type distributions and Markov models: theory and applications. Springer.

[6] Fiems, D., Mandjes, M., & Patch, B. (2018). Networks of infinite-server queues with multiplicative transitions. Performance Evaluation.

[7] Blom, G., Holst, L., & Sandell, D. (2012). Problems and Snapshots from the World of Probability. Springer Science & Business Media.

[8] Kelly, F., & Yudovina, E. (2014). Stochastic networks (Vol. 2). Cambridge University Press.

A. Mean Queue Length of an Infinite Server Queue

A.1. Basic Model with Classic Approach

In this section we consider an infinite server queue $(M(t), t \geq 0)$ with the transition rates as given in Table A.1. We will show that

$$\mathbb{E}[M(t)|M(0) = m_0] = m_0 e^{-\mu t} + \frac{\lambda}{\mu}(1 - e^{-\mu t}).$$

The Kolmogorov forward equations of this system are

$$\begin{aligned} p_0'(t) &= -\lambda p_0(t) + \mu p_1(t), \\ p_j'(t) &= \lambda p_{j-1}(t) + \mu(j+1)p_{j+1}(t) - (\lambda + \mu j)p_j(t),\ j > 0. \end{aligned}$$

Multiplying $p_j'(t)$ by e^{vj} gives

$$\begin{aligned} e^0 p_0'(t) &= -\lambda p_0(t)e^0 + \mu p_1(t)e^0 \\ e^{vj} p_j'(t) &= \lambda p_{j-1}(t)e^{vj} + \mu(j+1)p_{j+1}(t)e^{vj} - (\lambda + \mu j)p_j(t)e^{vj},\ j > 0 \\ &= -\lambda p_j(t)e^{vj} + \mu(j+1)p_{j+1}(t)e^{vj} + \lambda p_{j-1}(t)e^{vj} - \mu j p_j(t)e^{vj},\ j > 0. \end{aligned}$$

Transition	Rate
$M(t) \to M(t)+1$	λ
$M(t) \to M(t)-1$	$\mu M(t)$

Table A.1.: Transition rates for infinite server queue $(M(t), t \geq 0)$.

So that summing over j leads to

$$\sum_{j=0}^{\infty} p_j'(t)e^{vj} = -\lambda\sum_{j=0}^{\infty} p_j(t)e^{vj} + \mu\sum_{j=0}^{\infty}(j+1)p_{j+1}(t)e^{vj} + \lambda\sum_{j=1}^{\infty} p_{j-1}(t)e^{vj} - \mu\sum_{j=1}^{\infty} jp_j(t)e^{vj} \tag{A.1}$$

$$\begin{aligned}
&= -\lambda\sum_{j=0}^{\infty} p_j(t)e^{vj} + \mu\sum_{j=0}^{\infty}(j+1)p_{j+1}(t)e^{vj} + \lambda\sum_{j=0}^{\infty} p_j(t)e^{v(j+1)} \\
&\quad - \mu\sum_{j=0}^{\infty}(j+1)p_{j+1}(t)e^{v(j+1)} \\
&=\lambda(e^v-1)\sum_{j=0}^{\infty} p_j(t)e^{vj} - \mu(1-e^{-v})\sum_{j=0}^{\infty}(j+1)p_{j+1}(t)e^{v(j+1)}.
\end{aligned}$$

Define $\psi(v,t) = \mathbb{E}e^{vM(t)}$. Thus

$$\frac{\partial\psi(v,t)}{\partial t} = \lambda(e^v-1)\psi(v,t) - \mu(1-e^{-v})\frac{\partial\psi(v,t)}{\partial v}.$$

Hence

$$\frac{\partial^2\psi(v,t)}{\partial t\partial v} = \lambda(e^v)\psi(v,t) + \lambda(e^v-1)\frac{\partial\psi(v,t)}{\partial v} - \mu(e^{-v})\frac{\partial\psi(v,t)}{\partial v} - \mu(1-e^{-v})\frac{\partial^2\psi(v,t)}{\partial v^2}.$$

Leading to

$$\begin{aligned}
\frac{\partial^2\psi(v,t)}{\partial t\partial v}|_{v=0} =&\lambda(e^0)\psi(0,t) + \lambda(e^0-1)\frac{\partial\psi(v,t)}{\partial v}|_{v=0} \\
&- \mu(e^{-0})\frac{\partial\psi(v,t)}{\partial v}|_{v=0} - \mu(1-e^{-0})\frac{\partial^2\psi(v,t)}{\partial v^2}|_{v=0} \\
=&\lambda\psi(0,t) - \mu\frac{\partial\psi(v,t)}{\partial v}|_{v=0} \\
=&\lambda - \mu\frac{\partial\psi(v,t)}{\partial v}|_{v=0}.
\end{aligned}$$

Since $\mathbb{E}[M(t)|M(0)=m_0] = \frac{\partial\psi(v,t)}{\partial v}|_{v=0}$, we now have the following expression

$$\frac{\partial}{\partial t}\mathbb{E}[M(t)|M(0)=m_0] = \lambda - \mu\mathbb{E}[M(t)|M(0)=m_0].$$

Solving this ODE gives us

$$\mathbb{E}[M(t)|M(0)=m_0] = \frac{\lambda}{\mu} + Ce^{-\mu t}.$$

Using m_0 we are able to find an expression for C

$$\begin{aligned}
m_0 &= \frac{\lambda}{\mu} + C \\
C &= m_0 - \frac{\lambda}{\mu}.
\end{aligned}$$

Transition	Rate
$M(t) \to M(t)+1$	λ
$M(t) \to M(t)-1$	$\mu M(t)$
$M(t) \to aM(t)$	ν

Table A.2.: Transition rates for infinite server queue $(M(t), t \geq 0)$.

This gives us

$$\begin{aligned}\mathbb{E}[M(t)|M(0)=m_0] &= \frac{\lambda}{\mu} + (m_0 - \frac{\lambda}{\mu})e^{-\mu t} \\ &= m_0 e^{-\mu t} + \frac{\lambda}{\mu}(1-e^{-\mu t}).\end{aligned}$$

A.2. More Advanced Model with Alternative Approach

In this section we consider an infinite server queue $(M(t), t \geq 0)$ with multiplicative transitions. The transition rates are given in Table A.2. Due to the multiplicative transitions the approach of the previous section gets problematic if the method in Equation A.1 is applied. We therefore use an alternative related approach to show that

$$\mathbb{E}[M(t)|M(0)=m_0] = m_0 e^{-\mu t} + \frac{\lambda + a\nu}{\mu}(1-e^{-\mu t}).$$

We know that

$$\begin{aligned}\mathbb{E}e^{vM(t+\Delta t)} &= e^{v}\mathbb{E}[e^{vM(t)}\lambda\Delta t] + e^{-v}\mathbb{E}[e^{vM(t)}M(t)\mu\Delta t] + e^{av}\mathbb{E}[e^{avM(t)}\nu\Delta t] \\ &\quad + \mathbb{E}[e^{vM(t)}(1-\lambda\Delta t - M(t)\mu\Delta t)] - \mathbb{E}[e^{avM(t)}\nu\Delta t].\end{aligned}$$

Upon taking the derivative with respect to t we obtain

$$\begin{aligned}\frac{\partial}{\partial t}\mathbb{E}e^{vM(t+\Delta t)} &= \lambda e^{v}\mathbb{E}[e^{vM(t)}] + \mu e^{-v}\frac{\partial}{\partial v}\mathbb{E}[e^{vM(t)}] + \nu e^{av}\mathbb{E}[e^{avM(t)}] \\ &\quad - \lambda\mathbb{E}[e^{vM(t)}] - \mu\frac{\partial}{\partial v}\mathbb{E}[e^{vM(t)}] - \nu\mathbb{E}[e^{avM(t)}] \\ &= \lambda(e^{v}-1)\mathbb{E}[e^{vM(t)}] + \mu(e^{-v}-1)\frac{\partial}{\partial v}\mathbb{E}[e^{vM(t)}] + \nu(e^{av}-1)\mathbb{E}[e^{avM(t)}],\end{aligned}$$

so that upon taking the derivative with respect to v we obtain

$$\begin{aligned}\frac{\partial}{\partial v}\frac{\partial}{\partial t}\mathbb{E}e^{vM(t+\Delta t)} &= \lambda e^{v}\mathbb{E}[e^{vM(t)}] + \lambda(e^{v}-1)\frac{\partial}{\partial v}\mathbb{E}[e^{vM(t)}] \\ &\quad - \mu e^{-v}\frac{\partial}{\partial v}\mathbb{E}[e^{vM(t)}] + \mu(e^{-v}-1)\frac{\partial^2}{\partial^2 v}\mathbb{E}[e^{vM(t)}] \\ &\quad + a\nu e^{av}\mathbb{E}[e^{avM(t)}] + \nu(e^{av}-1)\frac{\partial}{\partial v}\mathbb{E}[e^{avM(t)}].\end{aligned}$$

Since $\mathbb{E}[M(t)|M(0)=m_0] = \frac{\partial}{\partial v}\mathbb{E}e^{vM(t+\Delta t)}|_{v=0}$, we obtain

$$\frac{\partial}{\partial t}\mathbb{E}[M(t)|M(0)=m_0] = \lambda - \mu\mathbb{E}[M(t)|M(0)=m_0] + a\nu.$$

Our goal is to solve this for $\mathbb{E}[M(t)|M(0)=m_0]$

$$\begin{aligned}
\frac{d\mathbb{E}[M(t)|M(0)=m_0]}{dt} &= \lambda + a\nu - \mu\mathbb{E}[M(t)|M(0)=m_0] \\
\frac{1}{\lambda + a\nu - \mu\mathbb{E}[M(t)|M(0)=m_0]} d\mathbb{E}[M(t)|M(0)=m_0] &= dt \\
\int \frac{1}{\lambda + a\nu - \mu\mathbb{E}[M(t)|M(0)=m_0]} d\mathbb{E}[M(t)|M(0)=m_0] &= \int dt \\
-\frac{\log(\lambda + a\nu - \mu\mathbb{E}[M(t)|M(0)=m_0])}{\mu} &= t + C \\
\log(\lambda + a\nu - \mu\mathbb{E}[M(t)|M(0)=m_0]) &= -(t+C)\mu \\
\lambda + a\nu - \mu\mathbb{E}[M(t)|M(0)=m_0] &= e^{-(t+C)\mu} \\
\mathbb{E}[M(t)|M(0)=m_0] &= \frac{\lambda + a\nu - e^{-(t+C)\mu}}{\mu}.
\end{aligned}$$

We use the initial condition m_0 to find an expression for C

$$\begin{aligned}
m_0 &= \frac{\lambda + a\nu - e^{-C\mu}}{\mu} \\
e^{-C\mu} &= \lambda + a\nu - \mu m_0 \\
-C\mu &= \log(\lambda + a\nu - \mu m_0) \\
C &= -\frac{\log(\lambda + a\nu - \mu m_0)}{\mu}.
\end{aligned}$$

This gives us the result

$$\begin{aligned}
\mathbb{E}[M(t)|M(0)=m_0] &= \frac{\lambda + a\nu - e^{-(t - \frac{\log(\lambda + a\nu - \mu m_0)}{\mu})\mu}}{\mu} \\
&= \frac{\lambda + a\nu - e^{-(\mu t - \log(\lambda + a\nu - \mu m_0))}}{\mu} \\
&= \frac{\lambda + a\nu - e^{-\mu t}(\lambda + a\nu - \mu m_0)}{\mu} \\
&= m_0 e^{-\mu t} + \frac{\lambda + a\nu}{\mu}(1 - e^{-\mu t}).
\end{aligned}$$