[THSt07] *Lars Arne Turczyk, Oliver Heckmann, Ralf Steinmetz;* **Simulation of Information Lifecycle Management;** Information Resources Management Association 18th IRMA International Conference 2007, Vancouver, Mai 2007.

Simulation of Information Lifecycle Management

Lars Arne Turczyk
TU Darmstadt
KOM Multimedia
Communications Lab
Merckstr. 25
64283 Darmstadt, Germany
Phone: +49 69 797 2778
lars.turczyk@siemens.com

Oliver Heckmann
TU Darmstadt
KOM Multimedia
Communications Lab
Merckstr. 25
64283 Darmstadt, Germany
Phone: +49 6151 16 5188
heckmann@kom.tu-darmstadt.de

Ralf Steinmetz
TU Darmstadt
KOM Multimedia
Communications Lab
Merckstr. 25
64283 Darmstadt, Germany
Phone: +49 6151 16 6150
ralf.steinmetz@kom.tu-darmstadt.de

ABSTRACT
In this paper we analyze the effects of the number of storage hierarchies in an ILM system. We describe the model for our simulator used to run the simulations. Afterwards the results are compared and recommendations are made.

Keywords
Information Lifecycle Management, ILM scenarios, storage hierarchies

1. INTRODUCTION
Information Lifecycle Management (ILM) is a strategic concept for storage of information and documents in which the value of the stored objects depends on the underlying business model and processes. Documents are assigned to a storage medium automatically so that the existing storage capacities can be used optimally and more cost efficiently.

For realizing cost potentials it is necessary to obtain a wider knowledge about ILM procedures and scenarios but experience reports do not exist in sufficient form and experimenting and searching in real systems is too expensive. Therefore the aim of this paper is to generate results and experiences by simulation of ILM scenarios.

First we work out the aims of a simulation and then create the corresponding simulation model. The model allows a strategy-orientated analysis. Based on this model a simulator was implemented as an examination tool for the behaviour of ILM scenarios. The simulation model uses results of our study conducted in 2006 [1] which analyzed the access behaviour on documents of a company database. The study provided a statistical description of the access patterns which is used in the model for the automatic migration of files.

The scientific use of the work consists of providing a model for simulation which generates generally utilizable results concerning ILM behaviour and cost optimization.

The results focus on the optimal number of storage hierarchies in an ILM system

The paper starts by listing the objectives of ILM simulations and describing the simulation model. Then the scenarios to be simulated are defined. In section 5 simulation results are presented and interpreted. The paper ends with an outlook on further ILM simulations.

2. RELATED WORK
Strange examined the long-term access behaviour on files in an UNIX system [2]. His aim was to identify regularities and patterns which can be applied to automated migration strategies for Hierarchical Storage Management (HSM). To verify hypotheses on migration algorithms a simulator was also designed and implemented. His examination is different to our approach on implementation. The simulator developed by Strange served as a tool merely for checking migration algorithms which were verified using observed access behaviour. A stochastic simulation of the access behaviour was renounced. Instead, the user behaviour was generated deterministically from the access protocols. Since only the effect of the migration rules was analyzed, he could restrict the number of feigned storage hierarchies to two. In addition, only very simple migration algorithms were used.

Further work deals mainly with algorithms which can be used for ILM or other storage strategies. Some examinations focus on the analysis of the access behaviour and the development of migration strategies.

The file migration protocol listing of a supercomputer was analyzed in a study by Katz and Miller. Migration methods were developed for a corresponding system [3].

Schmitz has also analyzed the access behaviour on files in a supercomputer to be able to derive an optimal migration strategy [4].

Miller and Gibson examined the access behaviour in further studies in UNIX environments and designed a "file aging algorithm" as a migration rule [5].

This paper is influenced by the analysis of the long-term access behaviour of Turczyk et.al. [1]. In contrast to other work, they analyzed Microsoft office files of a company database. In addition, they derived complex statistical rules for the migration of office files and documents.

3. SIMULATION MODEL

Our objectives below list how to implement a simulator as an examination tool for ILM scenarios. The primary objectives are the analysis of fundamental questions of ILM:

- Integral analysis of ILM scenarios (end-to-end)
- Identification of the necessary number of storage hierarchies

The simulator takes into consideration the integral lifecycle, i.e. from the initial situation designed for a company to the point where a stable state is reached. The simulator should offer transferable results concerning the questions mentioned above.

Some aspects of ILM have not been the subject of focus with this simulator and must therefore be considered separately. These secondary objectives are, in particular,:

- Identification of the necessary number of migration rules
- Optimization of the wording of migration rules
- Analysis of the dynamic behaviour of ILM scenarios
- Realization of potential for cost reduction

Here we list the assumptions which are to be made for implementation under consideration of the objectives. For the simulation of ILM the following assumptions and simplifications are met:

1. The ILM concept is independent from a special technical implementation
2. ILM works automatically
3. Analysis of Microsoft office files (doc, xls, ppt, etc.)
4. A cycle of the simulation corresponds to a working day.
5. Service control criteria have an effect on the user satisfaction and the acceptance of an ILM system but not on its effect.
6. For simulation a cycle-based approach is used. The use of a cycle-based simulation is founded on assumptions that automated migrations to a lower storage hierarchy are always carried out regularly at night. Migrations to a higher level are carried out within the corresponding cycle.

The application of these assumptions leads to the following the simulation model. The simulator uses predefined migration rules and scenarios to simulate the migration behaviour of an ILM system. The simulator uses some modules to reflect the objectives. Figure 1 shows the simulation model with its structural layout:

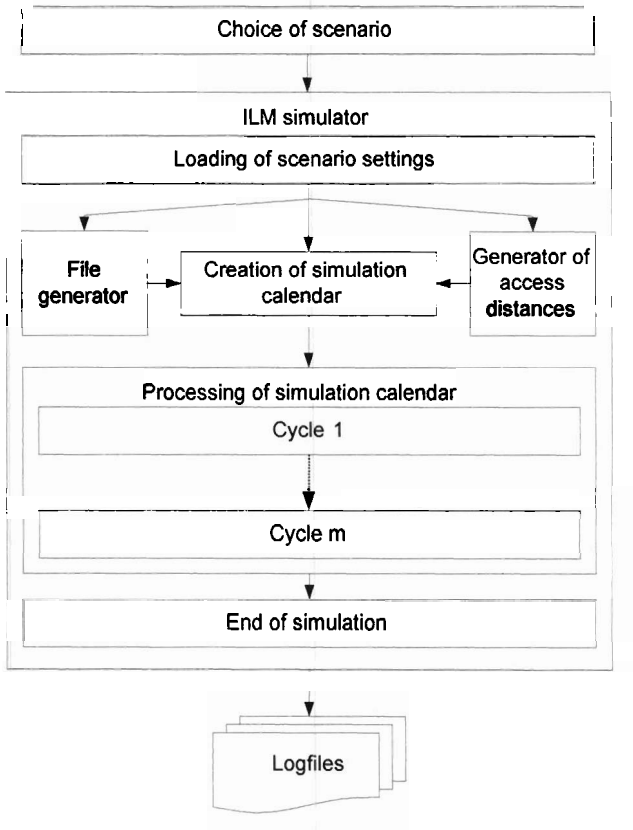

Figure 1: Simulation model

The main component of this plan is the ILM simulator. A scenario is loaded and simulations are executed. As a result the simulator generates logfiles. Any evaluation and interpretation of the results is done externally. How the simulator works is shown in the next chapter.

4. SIMULATED ILM SCENARIOS

The simulated scenarios are intended to reproduce different ILM concepts. The advantage is that, for example, companies thinking of implementing ILM obtain an insight into the behaviour of ILM-systems without purchasing components.

The cost effects are evaluated without changing the current storage environment.

4.1 General definition of ILM scenarios

A scenario describes all necessary elements of an initial situation for ILM. It provides a hypothesis on the circumstances under which ILM shall be employed in an organisation.

Definition 1 (Scenario):

A scenario is defined by:

1. Number of storage hierarchies
2. Starting situation
3. Number of files
4. Number of simulation cycles
5. Choice of a set of migration rules

For distinguishing different scenarios each scenario is labeled "H-T-I-D-R" according to table 1.

Table 1: Scenario label

Abbreviation	Description
H	Number of Hierarchies
T	Type of starting situation
I	Initial number of files
D	Duration (number of cycles)
R	Migration rules

For example, "H5-T2-I100-D4000-R5" is a scenario.

The number of storage hierarchies is a measure of the granularity of the simulated ILM system. It is the initial parameter and subject of examination of our work. With the help of a sensitivity analysis the adequate number of hierarchies is derived later in this paper.

The number of simulated files and the number of simulation cycles determine the parameters of the simulation. The set of migration rules determines, on the one hand, which rules are used and, on the other hand, to which groups of files these rules are applied.

When loading a scenario, the simulator receives the information listed in points 1 to 5. Next the starting situation is entered to define which type of ILM situation shall be simulated.

Definition 2 (Starting Situation):

The starting situation defines the level of filling of the ILM system. It is either already filled at the beginning or it is empty. In addition it is determined if there is an increase in the amount of stored files. Third, the starting situation defines whether, in a filled system, the files are presorted.

The following table shows all possible types of starting situations:

Table 1: Starting situations

	Files are concentrated on storage hierarchy 1	Files are distributed according to their values over all storage hierarchies
filled ILM system	Type 1	Type 4
Unfilled ILM system with data growth	Type 2	Type 5
filled ILM system with data growth	Type 3	Type 6

Remark:

1. The option "Files are concentrated on storage hierarchy 1" defines that all files are put on storage hierarchy 1 at the beginning of the simulation.
2. The option "Files are distributed according to their values over all storage hierarchies" defines that all files are already distributed over the different storage hierarchies at the beginning of the simulation.
3. The option "filled ILM System" means that the observed files are already in existence at the beginning of the simulation.
4. The option "unfilled ILM System with data growth" means that the ILM system is empty at the beginning of the simulation and the files will be added during the simulation.
5. The option "filled ILM system with data growth" combines a filled system with the adding of files during the simulation.
6. In case of types with data growth (types 2, 3, 5 and 6) the data growth has to be specified in addition.
7. The simulation of types 4, 5 and 6 requires the use of metadata.

Distribution of the files on different hierarchies at the beginning of an ILM system turns out to be problematic. To be able to assign the files to their storage hierarchies they have to be classified according to their value. From the data offered by file systems about files the value cannot be extracted. For the correct valuation metadata about the files is needed. In general these metadata are also not available. Therefore a correct valuation could be made if at the implementation of an ILM system the values of the files were assigned manually by the user. As most companies will avoid this, starting situations of types 4-6 are not simulated here.

4.2 Definite simulated ILM scenarios

In the simulation of the integral behaviour we observe scenarios with a duration of 4,000 days (H5-T2-I500-D4000-R5).

To examine the effect of the number of hierarchies four simulation runs are carried out. To make circumstances as realistic as possible, type 3 scenarios are simulated. The adopted data growth is about 20% per annum. The simulator starts the simulation with a data stock of 500 files. The simulation duration is 2,000 days (Hx-T3-I500-D2000-R5).

5. SIMULATIONS

For simulations we first look at single files and derive the meaning of jitter. When considering what happens on the storage hierarchy jitter is a reasonable measure for considering what happens with the files. There are two levels of observation the micro-level focussing on single files and the macro-level focussing on entire storage hierarchies.

5.1 Micro-level analysis

For the integral examination of the behaviour of ILM scenarios the complete lifecycle has to be observed. In accordance with the definition of a lifecycle this starts with the generation of a file and ends with its "final use" [6].

Since the simulator is able to observe the value of a file over the complete time interval between first storage and deletion of the file, the simulator can be considered integral. In the simulations we observe the integral behaviour over duration of 4,000 days.

The micro-level looks at the single file with regard to:

- Access frequency
- Migration activity.

5.2 Access frequency

The access frequency of a file is quantity of access within one day. By observing the access activity of a single file we gain an insight into its lifecycle. Figure 2 shows the access frequency of a single file graphically.

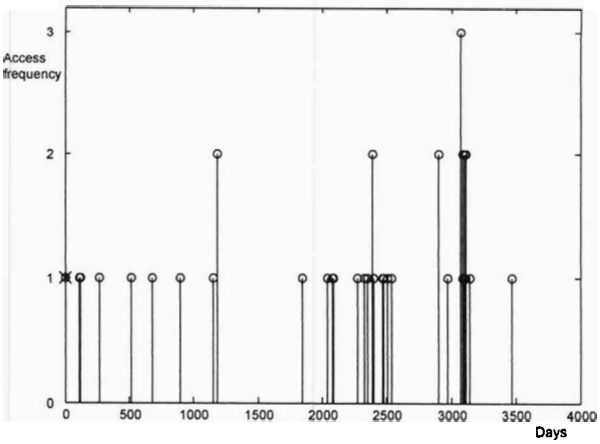

Figure 2: Access frequency of a single file

The access frequency is used to calculate the probability of further accesses. This is possible because the distribution functions of the accesses on file are known [1].

Hence the access frequency influences mainly the migration activity which is described in the next section.

5.3 Migration activity

The migration activity describes the appearance of migrations of a file between the storage hierarchies.
Figure 3 shows the migration activity of the same file considered in figure 2.

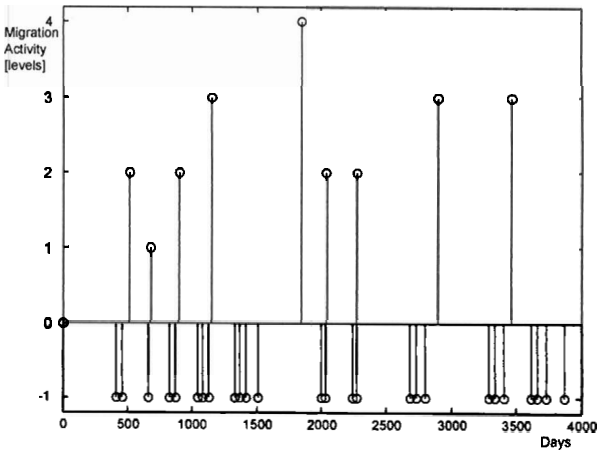

Figure 3: Migration activity

The negative amplitudes mark migrations to lower hierarchies as a consequence of lesser information value. The positive amplitudes stand for re-migrations of the file to a higher hierarchy, i.e. the file was already migrated to a more cost efficient hierarchy before it is put back to a higher level again. Figure 4 shows how the storage conditions of the file change during its lifecycle and offers an integral observation of a single file in the ILM system.

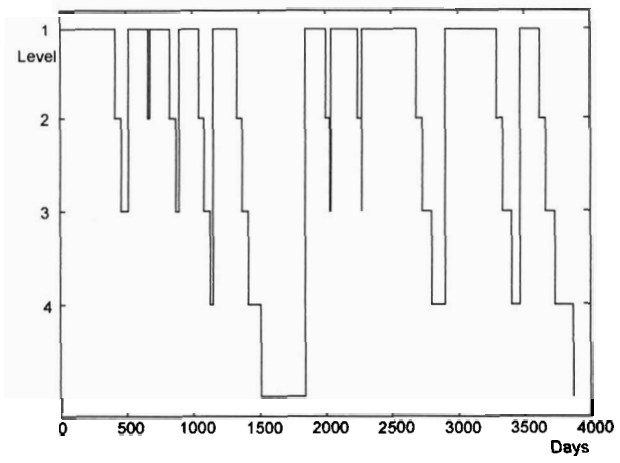

Figure 4: Integral observation of a single file

In figure 4 you can see that between day 1,300 and day 1,500 after the generation of the file the access probability sinks considerably. Therefore the file is moved to the lowest storage hierarchy in several steps. After approximately 1,800 days the file is accessed so that it must be shifted to the highest storage level again.
After about 3,800 days the information is again at the lowest level. The lifecycle follows the typical course of a file with a periodical access sample.
Since the file trembles between the storage hierarchies, this is a good example for the demonstration of jitter. Until the file was definitely to be stored on level 5, it was migrated back to a higher hierarchy nine times in 4,000 days. Therefore it has a jitter of J(4000)=9. For the simultaneous analysis of several files the graphic evaluation does not make sense, therefore the jitter is an important measure to generate results over all files. This happens when examining the number of levels at the macro-level.

5.4 Macro-level analysis

For the examination of the effect of the number of storage hierarchies four simulation runs are carried out. To make circumstances as realistic as possible, type 3 scenarios are simulated. The assumed data growth is about 20% per annum. The simulator starts the simulation with a data stock of 500 files. The simulation duration is 2,000 days. Ten simulation runs are averaged to one simulation to reduce fluctuations of measurements.

The migration rules used in the simulations are based on our study [1]. As distribution function either the Weibull-distribution ($W(\alpha;\beta)$) or Gamma-distribution ($G(\alpha;\beta)$) is used (see table 3).

Table 3: Applied distribution functions

Number of accesses	1-6	7-14	15-∞
File type			
doc	W(0,35;3,5)	G(0,32;183)	W(0,35;3,5)
xls	W(0,25;1,1)	W(0,25;1,1)	W(0,25;1,1)
ppt	W(0,38;14,3)	W(0,38;14,3)	W(0,38;14,3)
pdf	W(0,35;3,5)	G(0,32;183)	W(0,35;3,5)
other	W(0,46;27,7)	G(0,29;181)	W(0,46;27,7)

The number of storage hierarchies is the initial variable of the simulations. At every simulation the first level has a threshold probability of p1 = 10%. The distances of the threshold probabilities dpi of the other levels i are equidistant, i.e. the remaining 10% are split up equally (see figure 5).

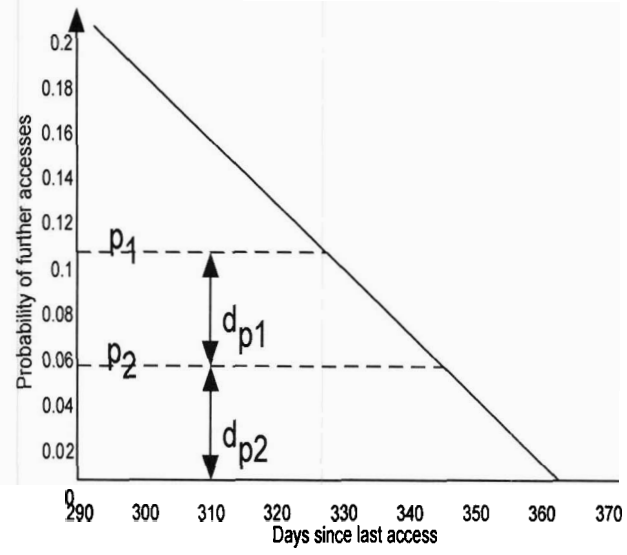

Figure 5: Equidistant threshold probabilities in case of two hierarchies with p1=10% and p2=5%

The threshold probability is described as the probability of further accesses on the file, where the file is assigned to a new hierarchy. In the example shown the probabilities of the first and second level are p1 = 10% or rather p2 = 5%. When the probability of further accesses on a file stored on hierarchy 1 falls below the threshold probability of 5%, it is migrated to hierarchy 2.

When adding a new storage hierarchy the threshold probabilities are adapted correspondingly so that they are of equal distance to each other again.

The influence on the number of hierarchies is observed by means of the relative capacity-need. In addition the jitter serves as a measure to look at the reliability of the system.

Now the individual simulation-runs and the accompanying results are explained. At the beginning a simulation is carried out with two hierarchies. The number of storage

hierarchies is increased by one per each run up to a maximum number of five hierarchies.

At the first simulation there is only one threshold probability of p1 = 10% which lies between level 1 and 2. Figure 6 shows the result of the simulation.

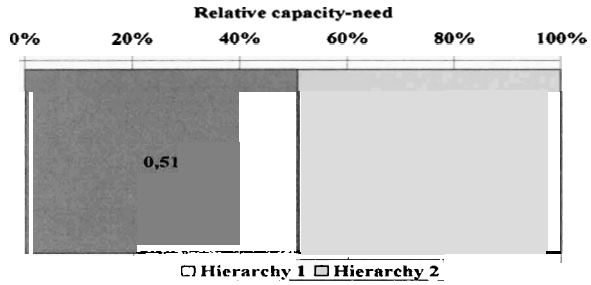

Figure 6: Mean relative capacity-needs for two hierarchies

In simulation 1 the relation between hierarchies 1 and 2 is approximately 1:1, i.e. almost half of the complete data stock is stored on the second, more economical hierarchy level. The average jitter is J(1000) = 2.136.

In simulation 2 three hierarchies are available for the storage of the files. The related value probabilities are p1 = 10% and p2 = 5 %.

Figure 7 represents the result of simulation graphically by means of the relative capacity need.

Again approximately 50% of the data are on the first storage hierarchy. On the second hierarchy nearly a sixth of the complete stock is kept and on the third hierarchy nearly a third of the files is stored.

A mean jitter of J(1000) = 2.093 was measured.

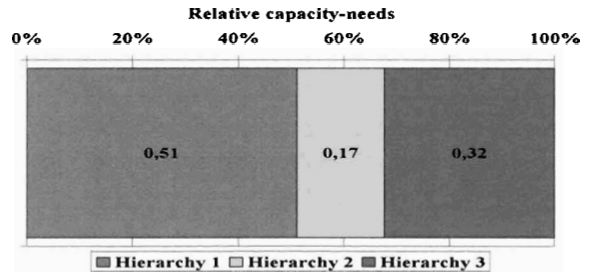

Figure 7: Mean relative capacity-needs for three hierarchies

In simulation 3 the probability values are p1 = 10%, p2 = 6.66% and p3 = 3.33%. The mean capacity values arising from the simulation are represented in figure 8.

Hierarchy 1 keeps 51% and hierarchy 2 keeps 10%. Hierarchies 3 and 4 keep 14% and. 25% respectively.

The average jitter is J(1000) = 2.16.

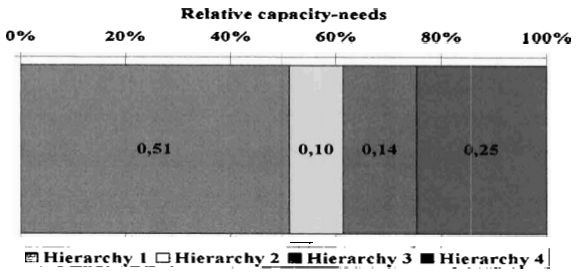

Figure 8: Mean relative capacity-needs for four hierarchies

In simulation 4 a scenario with 5 hierarchies is simulated. The related probability values are p1 = 10%, p2 = 7.5%, p3 = 5% and p4 = 2.5%.

The results are shown in figure 9. Hierarchy 1 keeps 51 % and hierarchy 2 keeps 7.4%. Hierarchies 3 and 4 keep 9.2% and 12.3% respectively. Hierarchy 5 keeps 20.9%.

The measured jitter was J(1000) = 2.17.

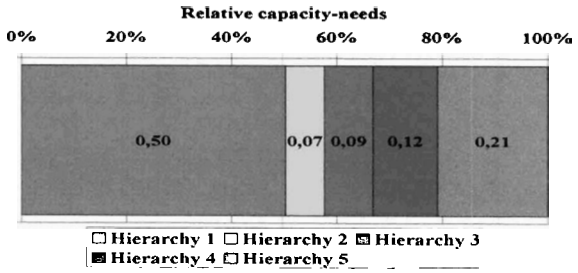

Figure 9: Mean relative capacity-needs for five hierarchies

5.5 Results and Interpretation

The observed jitter values of the four simulations vary less than 5%. It can be assumed that the reliability of an ILM system is independent of the number of hierarchies.

Figure 10 gives an overview of the capacity-need of the different scenarios.

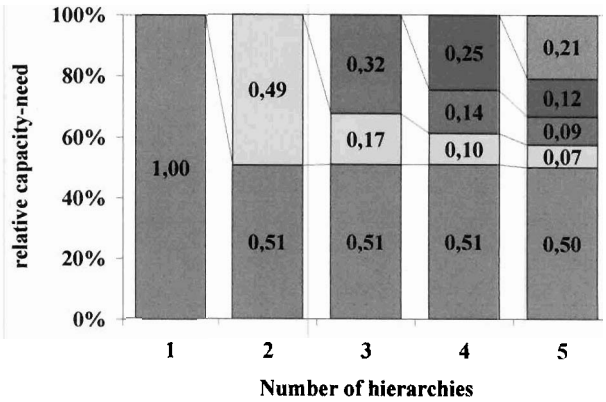

Figure 10: Overview of the mean capacity-needs

By applying the sensitivity analysis we examine the effects of a variation in the number of hierarchies.

The storage demand of the first hierarchy remains constant irrespective of changes in the number of storage hierarchies.

The capacity need of the second level is reduced with an increasing number of storage hierarchies.

The reason for this behaviour of the ILM system is the specification of the threshold probabilities.

With any added hierarchy the distances of the values change.

If there are only two hierarchies, the second hierarchy stores all files with an access probability of less than 10%.

If three hierarchies are used, the second only keeps files with an access probability of between 5% and 10%. Therefore the relative capacity need of the second level becomes smaller.

Generally speaking from all hierarchies lower than hierarchy 1, the lowest hierarchy always keeps the largest part of the files. Altogether, the greatest share of the information is stored on the top and the bottom hierarchies. This result coincides with the observations in real IT systems and is an essential driver for ILM. [7].

5.6 Application of the results

The aim of the simulation of ILM in this paper was to determine a reasonable number of hierarchies. Since the optimal number of hierarchies is determined by different factors, particularly costs and QoS requirements, this task can only be solved with additional assumptions.

At first the administrative costs are assumed to increase with each additional hierarchy. The gain resulting from an added hierarchy has to cover at least the additional cost. This is achieved by a sufficient degree of utilization.

It can therefore be concluded that an additional storage hierarchy is only profitable if it stores at least 20% of the overall storage demand.

Of course the optimal number of hierarchies depends on the business processes, but in general two to a maximum of three hierarchies seem to be quite adequate in the case of regular ILM deployment.

In a 2-hierarchy-design where the first hierarchy is hard-disk based (e.g. FC or iSCSI) and the second hierarchy is realized by slow visual storage components (e.g. MO) it is recommended that the threshold probability be set to p1 = 5%.

In this case hierarchy 1 would store approximately 68% and the second hierarchy would store 32% of total.

When extending the design to a 3-hierarchy-design where the first hierarchy is hard-disk based (e.g. FC or iSCSI), the second is for example S-ATA and the third hierarchy is realized by slow MO it is recommended to put the threshold probability to p1 = 10% and p2=5%.

In this case hierarchy 1 would keep 50%, hierarchy 2 18% and hierarchy 3 32%, i.e. hierarchy 1 would be exonerated by 18%

6. SUMMARY AND OUTLOOK

We presented simulation results for Information Lifecycle Management. The objective was focused on the optimal number of storage hierarchies. Although the number depends on the definite business process, the range of numbers of hierarchies could be isolated. In the next step further ILM scenarios will be simulated and compared. The focus will lie on the secondary objectives listed in chapter 3.

REFERENCES

[1] Turczyk, Lars Arne; Groepl, Marcel; Heckmann, Oliver and Steinmetz, Ralf: Analyse von Datei-Zugriffen zur Potentialermittlung fuer Information Lifecycle Management, TU Darmstadt KOM Technical Report 01/2006

[2] Strange, Stephen: Analysis of Long-Term UNIX File Access Patterns for Application to Automatic File Migration Strategies. Technical Report UCB/CSD-92-700, EECS Department, University of California, Berkeley, 1992.

[3] Miller, Ethan L. and Katz, Randy H.: An Analysis of File Migration in a UNIX Supercomputing Environment. In: USENIX Winter, pages 421–434, 1993.

[4] Schmitz, Carolin: Entwicklung einer optimalen Migrationsstrategie für ein hierarchisches Datenmanagement System. Technischer Bericht, Forschungszentrum Jülich GmbH, 2004.

[5] Gibson, T. and Miller E.: An Improved Long-Term File-Usage Prediction Algorithm, 1999.

[6] Turczyk, Lars Arne; Heckmann Oliver; Berbner, Rainer and Steinmetz, Ralf: A Formal Approach to Information Lifecycle Management. In Proceedings of IRMA '05, Washington DC, May 2005

[7] Gostner, Roswitha; Turczyk, Lars; Heckmann, Oliver and Steinmetz, Ralf: Analyse von Datei-Zugriffen zur Potentialermittlung fuer Information Lifecycle Management, TU Darmstadt KOM Technical Report 01/2005