AMCIS 2007 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2007

Modeling Information Lifecycle Management

Lars Turczyk
TU Darmstadt

Nicolas Liebau

Recommended Citation

Turczyk, Lars and Liebau, Nicolas, "Modeling Information Lifecycle Management" (2007). *AMCIS 2007 Proceedings*. 39.
http://aisel.aisnet.org/amcis2007/39

Modeling Information Lifecycle Management

Lars Arne Turczyk
Technische Universität Darmstadt
KOM – Multimedia Communications Lab
Merckstr. 25
D-64283 Darmstadt
Germany
Lars.Turczyk@siemens.com

Nicolas Liebau
Technische Universität Darmstadt
KOM – Multimedia Communications Lab
Merckstr. 25
D-64283 Darmstadt
Germany
Nicolas.Liebau@kom.tu-darmstadt.de

Ralf Steinmetz
Technische Universität Darmstadt
KOM – Multimedia Communications Lab
Merckstr. 25
D-64283 Darmstadt
Germany
Ralf.Steinmetz@kom.tu-darmstadt.de

ABSTRACT

Information Lifecycle Management (ILM) is a strategic concept for storage of information and documents. ILM is based on the idea that in an enterprise different information have different values. Valuable information is stored on systems with a high quality of service (QoS). The value changes over time and therefore migration of information is required to cheaper storage systems with a lower QoS. Automated migration makes ILM dynamic. Such automation requires storage systems to understand what files are important at what time so that right policies can be applied. In this point ILM nowadays lacks methods and tools.

In this paper we describe the modeling of a simulator for Information Lifecycle Management (ILM). The objectives are determined verbally and the related assumptions are specified. The model is implemented into a simulator, which allows dynamic ILM considerations. 90% of decision makers consider implementing ILM (Linden 2006) but there are too few experience reports and experimenting and researching in real systems are too expensive. This paper addresses this issue and contributes to supporting and assisting IT managers in their decision-making process. The simulative approach has two advantages. First, it offers predictions about the long-term dynamic behaviour of an ILM scenario. Second, it dispenses with real storage hardware which allows comparing architectural design alternatives.

The first simulation results focus on a reasonable number of hierarchies in ILM scenarios. The results raise the awareness of the required number of hierarchies and the choice of storage technologies. They help to avoid mismanagement and poor investments in advance. This shows that the simulator is a useful tool for design of ILM solutions. Together with other tools like TCO-calculators it supports the decision process of IT managers.

Keywords

Information Lifecycle Management, Model, Simulator, File Migration, Decision Support.

INTRODUCTION

Information Lifecycle Management (ILM) is a strategic concept for storage of information and documents (Peterson 2004). In 2005 META forecasted the annual growth rates of tiered infrastructure as follows: 20% to 25% percent for enterprise (monolithic), 50% to 55% for midrange (modular), and 80% to 85% for capacity-based, yielding aggregate storage growth of about 45%. Like-for-like price/capacity will improve 35% per year. The implication is that the use of lower cost storage alternatives will be a key facet of storage growth over the next few years (Meta 2005)

ILM is based on the idea that in an enterprise different information have different values. Valuable information is stored on systems with a high quality of service (QoS). The value changes over time and therefore migration of information is required to cheaper storage systems with a lower QoS. Automated migration makes ILM dynamic. Such automation requires storage systems to understand what files are important at what time so that right policies can be applied. In this point ILM nowadays lacks methods and tools. Furthermore the steadily growing storage demand requires tools to consider the dynamic behaviour of file migration. ILM offers significant potential cost savings by tiering storage and 90%of decision makers consider implementing ILM (Linden 2006). Nonetheless, there are too few experience reports on this subject and experimenting and researching in real systems are too expensive.

A simulative approach could support this but there are no ILM simulators available. A simulator would have two advantages. First, it offers predictions about the long-term dynamic behaviour of an ILM scenario. Second, it dispenses with real storage hardware which allows comparing architectural design alternatives. The research question is “How has an ILM-simulator to be modelled?”. Our model intends to support the design and decision process on ILM implementations.

To be truly successful with the design and implementation of an ILM solution, it is important to remember that only about 25% of the challenge relates to the selection of products (Goepel 2005), i.e. 75% of the effort is taken up by the pre-purchase phase with the solution design. We focus on these 75% and derive a model that leads to a simulation tool. The simulator offers the possibility to check the design under dynamic circumstances without implementing real systems. It shows the distribution of the storage demand over the storage hierarchies. This offers the chance to avoid misplanning and misinvest in advance.

First we discuss the modeling of ILM. Then the feasibility of the model is proven by observing a single file (micro observation). Afterwards simulations of several scenarios are performed. The focus of the simulations lies in the adequate number of storage hierarchies (macro observation). Determining the number of hierarchies is the most important question when designing ILM architectures. The paper ends with the application of the simulation results to ILM design.

The essence of this paper is as follows:

1. We present a detailed model for ILM in an enterprise.
2. We use the model for dynamic ILM simulations.
3. We show how the simulations lead to design guidelines concerning the number of hierarchies in ILM environments.

RELATED WORK

Migration of files from more expensive storage to less expensive storage has been studied as far back as early1980s. These studies (Smith 1982 and Lawrie et. al. 1982) concluded that a file selection algorithm based on file age and size results in a minimum amount of file recall occurrences and in optimal storage utilization. Strange (1992) examined the long-term access behaviour on files in an UNIX system. 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 has great differences to our approach for 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 (1993). Migration methods were developed for a corresponding system. Schmitz (2004) has also analyzed the access behaviour on files of a supercomputer to be able to derive an optimal migration strategy. Miller and Gibson (1999) examined the access behaviour in further studies in UNIX environments and designed a "file aging algorithm" as a migration rule.
In the area of web caching similar investigations were done. The well known Least Recently Used (LRU) algoritm and its derivatives aim in the same direction of optimization like ILM (Robinson and Devarakonda 1990; Jelenkovic and Radovanovic 2004). LRU discards the least recently used items first. This algorithm requires keeping track of what was used when, which is expensive if one wants to make sure the algorithm always discards the least recently used item. In opposite to web caching in ILM there is no discarding of files.

Today ILM is a strict focus of research. The main results are found in the field of “How” ILM works, i.e. most research was done in the field of procedures and policies. Vendors presented their understanding of ILM (Reiner et. al. 2004). Turczyk et.

al. (2006) gave a formal definition usable for ILM abstraction. Beigi et. al. (2005) and Tanaka et. al. (2005) offered proposals for policy description of ILM. Last but not least, Chen (2005) focused on the valuation of files which can be used as a part of an ILM model.

SIMULATION MODEL

This section focuses on the "How" can ILM be modeled. It gives an overview about the programming structure, too. First a simulation plan is developed and afterwards the simulation model is presented.

Model variables and definitions

The ILM model consists of a simulation plan, the model itself and the simulation cycle. These three issues are described sequently.

The simulation plan gives an overview on the environment in which the simulator works and other working packages connected to it. 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. The simulation plan looks as follows:

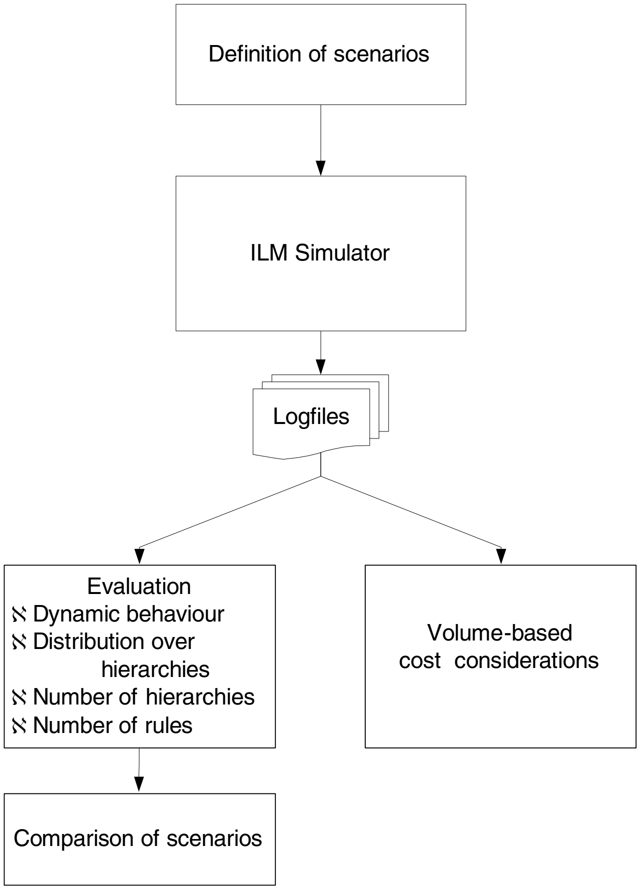

Figure 1. Simulation Plan

Inside the simulator uses predefined migration rules and scenarios to simulate the migration behaviour of an ILM system. The simulator uses two modules, a file generator and a generator for access distances. They are described subsequently. First we show the simulation model with its structural layout (see figure 2).

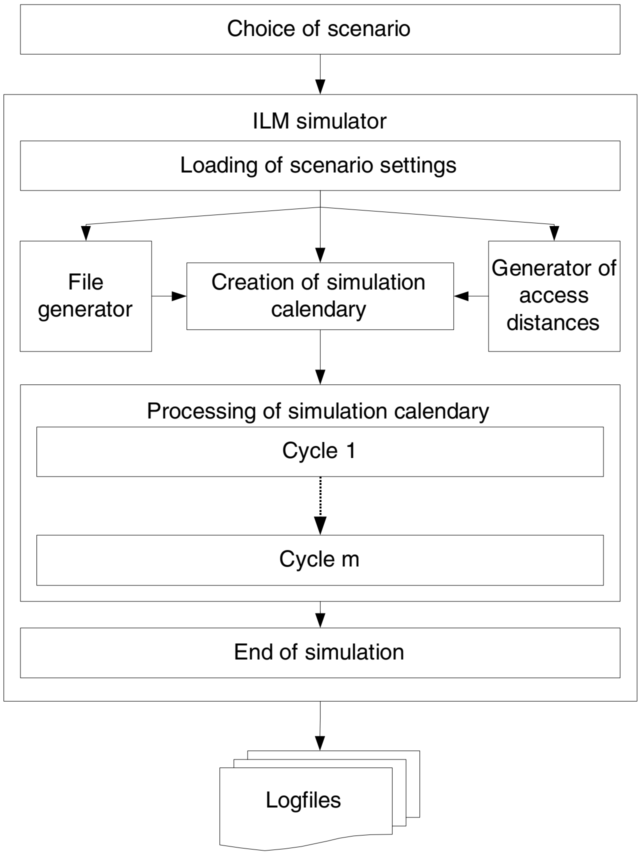

Figure 2. Simulation Model

Now we define what is meant by “scenario” which is used by the modules “file generator” and “generator of access distances”

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

When loading a scenario, the simulator receives the information listed in points 1 to 5.

For distinguishing different scenarios each scenario is labelled “H-T-I-D-R” according to table 1.

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

Table 1. : Scenario Label

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

With these settings the required number of files is generated. This happens by a special module, the file generator.

Modeling objectives and assumptions

The primary objectives are the analysis of fundamental matters of ILM:

1. Integral analysis of ILM scenarios (i.e. the whole lifecycle is observed)
2. Identification of the necessary number of storage hierarchies
3. Observing the relative capacity needs per hierarchy
4. Observing the migrations and re-migrations (jitter)

The model 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 implemented model in a simulator shall offer transferable results concerning the matters mentioned above.

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

Assumption 1 is based on Turczyk (2005) who stated independence of long-term ILM conception from short-term storage product lifecycles. Assumption 2 is a general characteristic of all ILM-solutions (Peterson 2004).

Implementation details

To create a realistic model we conducted two case studies in advance. In 2005 we analysed a knowledge database of a consulting department within a large German enterprise (Gostner et. al. 2005). The results showed on the one hand that there is demand in industry for ILM and on the other hand that Microsoft office application formats .doc, .xls and .ppt dominate the file population. Next in 2006 a second case study focussed on the access behaviour on the MS-files (Groepl et. al. 2006). The access behaviour is an important issue in ILM simulations. The results of the second case study are used in the simulator.

Module File Generator

The module file generator is needed to generate the input for the simulations. The module produces stochastically calendar items for the generation of files. A file is characterized by the attributes “file type”, “file size” and “time of the generation”.

The process of file generation is uniquely indicated by distribution functions of the random variables for the attributes:

- File type: weighted uniform distribution
- File size: Normal distribution
- Time of the generation or rather distance between two file generations: Exponential distribution

Module Generator for Access Distances

This module generates and assigns the accesses of each single file. This is done using a mathematical approach. The generator creates stochastically distances between two file accesses. The generator is based on our case study where we derived distribution functions for file accesses. This happens in conjunction with the file type and access history. As distribution function either the Weibull-distribution (W(α;β)) or Gamma-distribution (G(α;β)) is used (see table 2).

Number of accesses File type	1-6	7-14	15-∞
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)

Number of accesses File type	1-6	7-14	15-∞
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)

Table 2.: Applied Distribution Functions

The distribution functions make the calculation of the probability of further accesses possible. Calculating the access probabilities of a single file leads to migration rules which use percent-values. For example, one of the later used migration rules is: “If the access probability of a file falls below the threshold probability of 10%, this file is migrated to the next lower storage hierarchy.”

Simulation Cycle

Each simulation cycle has the same basic structure. A cycle corresponds to one working day. Therefore the number of cycles defines the simulated lifecycle of the file within the ILM system. For simulation purposes the number of cycles can be set to infinity. This option is used to observe if the ILM system reaches a stable state. For simulation of realistic ILM scenarios the number of cycles is limited in order to see how long it takes to achieve effects. Figure 4 shows the structure of a cycle.

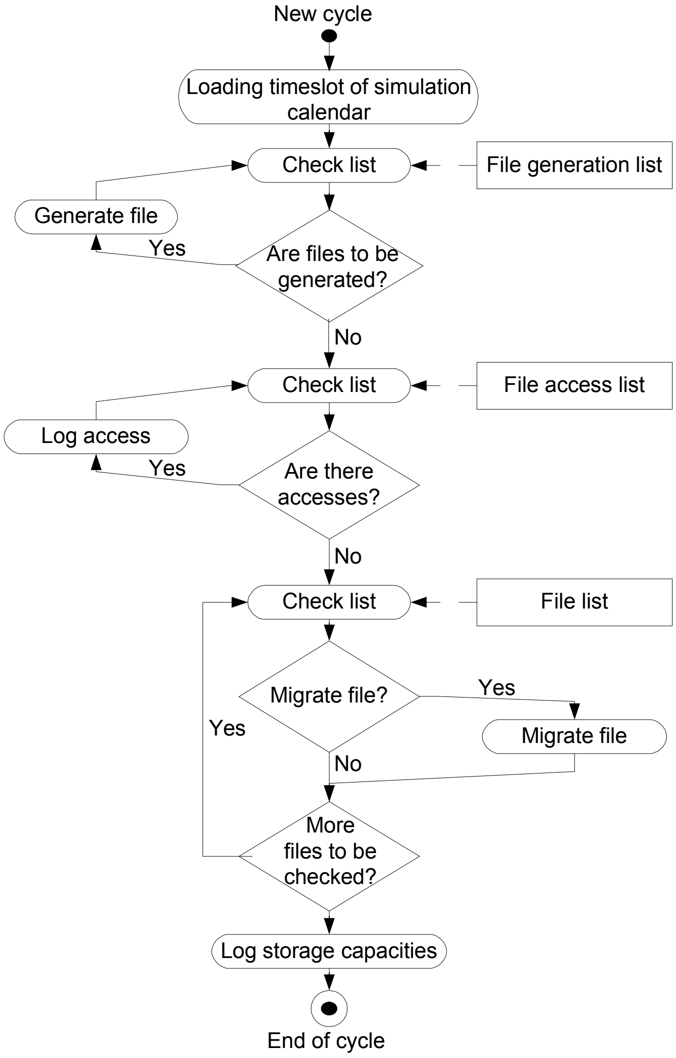

Figure 4. Simulation Cycle

Now the model is ready to be implemented. In the next section we check the feasibility of the model. Afterwards the definite simulations are done.

PROOF OF CONCEPT

The main task of this section is to investigate if the model works. Therefore we analyze the behavior of a single file (micro observation) and the effects on each storage hierarchy (macro observation). A realistic simulator must track each single file and aggregate the files over all hierachies, i.e. the micro level is the basis for the macro level. The macro level is point of interest, because here the storage demand per hierarchy is observed.

The simulator neglects costs of migration and focusses on cost of storage. Nonetheless the microlevel offers the input for calculating migration cost.

Micro-level analysis

For the integral examination of the behavior 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" (Peterson 2004).

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 behavior over duration of 4,000 days.

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

- Access frequency
- Migration activity.

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.

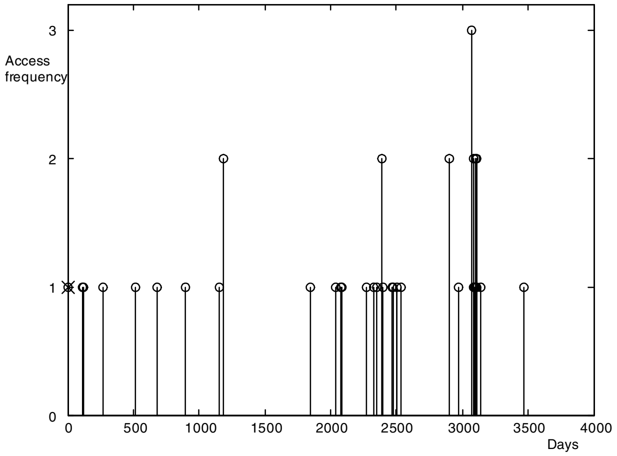

Figure 5: Access frequency of a single file

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 (see table 2).

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

Migration activity

The migration activity describes the appearance of migrations of a file between the storage hierarchies.
Figure 6 shows the migration activity of the same file considered in figure 5.

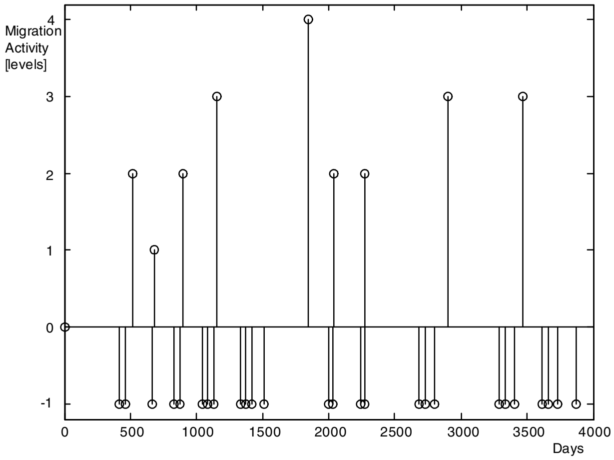

Figure 6: Migration activity

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 7 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.

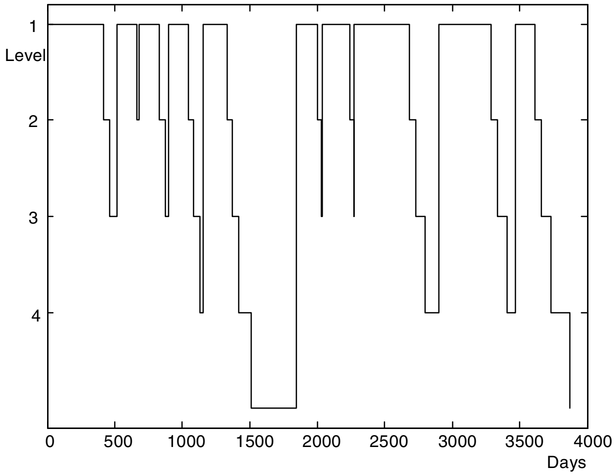

Figure 7: Integral observation of a single file

In figure 7 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.

Macro-level analysis

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.

Figure 8 shows the results of a simulation with 3 hierarchies. As migration rules were used:

Rule 1: Migrate the file if access probability falls below 10% (p1=10%).

Rule 2: Migrate the file if the access probability falls below 5% (p2=5%).

Rule 3: Move the file to level 1, if the file is accessed.

The annual growth rate 3 was 20%.

After 3200 simulation cycles the relative capacity need per hierarchy was:

Hierarchy 1: 47%

Hierarchy 2: 17%

Hierarchy 1: 36%

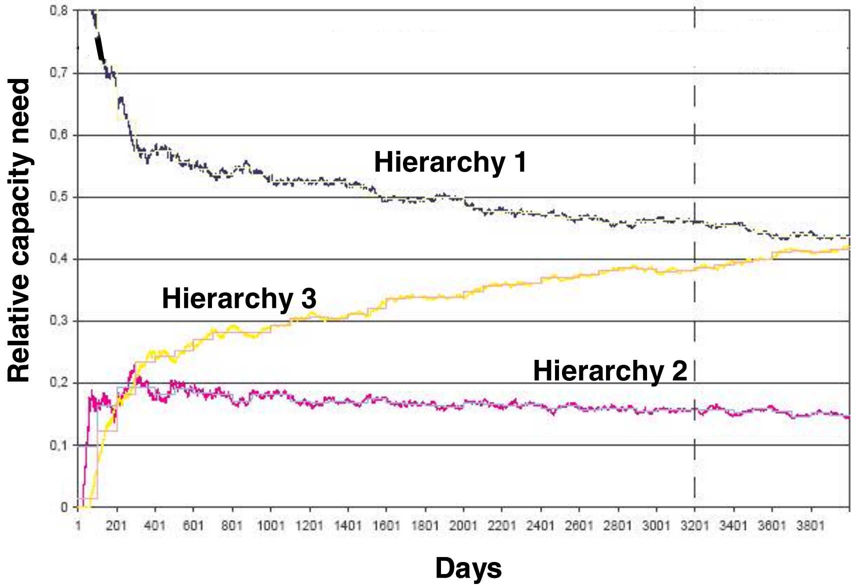

Figure 8: Type 1 Scenario

Now the simulator is fit for service. In the next section we start the definite simulations for answering the question of the adequate number of hierarchies in ILM design.

DEFINITE SIMULATIONS

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 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. This happens in order to detect the stability in the system (Eickhoff 2002)

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.

Simulation 1: At the first simulation there is only one threshold probability of p1 = 10% which lies between level 1 and 2.

Relative capacity need hierarchy 1: 51%

Relative capacity need hierarchy 2: 49%

The average jitter is J(1000) = 2.136.

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.

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

Relative capacity need hierarchy 1: 51%

Relative capacity need hierarchy 2: 17%

Relative capacity need hierarchy 3: 32%

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

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.

Simulation 3: In simulation 3 the probability values are p1 = 10%, p2 = 6.66% and p3 = 3.33%.

Relative capacity need hierarchy 1: 51%

Relative capacity need hierarchy 2: 10%

Relative capacity need hierarchy 3: 14%

Relative capacity need hierarchy 4: 25%

The average jitter is J(1000) = 2.16.

Simulation 4: 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%.

Relative capacity need hierarchy 1: 50%

Relative capacity need hierarchy 2: 7%

Relative capacity need hierarchy 3: 9%

Relative capacity need hierarchy 4: 12%

Relative capacity need hierarchy 5: 21%

The measured jitter was J(1000) = 2.17.

Results and Interpretation

Result 1: *The observed jitter values of the four simulations are almost constant.*

This proves that the reliability of an ILM system is independent of the number of hierarchies.

Result 2: *From all hierarchies lower than hierarchy 1, the lowest hierarchy always keeps the largest part of the capacity-need.*

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.

Figure 9 gives a summary of the capacity-need of the different scenarios.

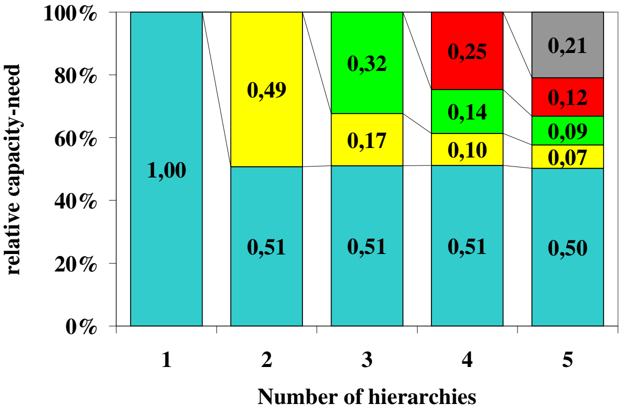

Figure 9: Overview of the mean capacity-needs

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 (Merrin and Harnetty, 2005).

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.

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.

Example:

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) the simulations show that the threshold probability can 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 the simulations show that the threshold probabilities can be set 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%

SUMMARY AND OUTLOOK

90% of decision makers consider implementing ILM (Linden 2006) but there are too few experience reports and experimenting and researching in real systems are too expensive. This paper addresses this issue and contributes to supporting and assisting IT managers in their decision-making process. We presented a simulation model for Information Lifecycle Management in an enterprise. The simulative approach has two advantages. First, it offers predictions about the long-term dynamic behaviour of an ILM scenario. Second, it dispenses with real storage hardware which allows comparing architectural design alternatives. The objective focused on the optimal number of storage hierarchies.

We showed how the simulations lead to design guidelines concerning the number of hierarchies in ILM environments. Although the number depends on the definite business process, the range of numbers of hierarchies could be isolated. 75 % of the effort for ILM decision making is taken up by the pre-purchase phase with the solution design. Here the simulator supports IT decision makers by accelerating the process and offering concrete arguments.

The limitation of the model is that migration cost are not considered and the focus lies on storage capacity cost. In order to be able to make a better decision IT managers need to know the migration cost, too. With the observation of the lifecycle for each single file, the simulator offers the information required for calculating the migration cost. Here an extension for the simulator will be implemented.

In the next steps the strategy of the simulative approach will be followed closely. Further ILM scenarios will be simulated and compared. The focus will lie on migration strategies, i.e. the migration policies and their optimization.

REFERENCES

1. Beigi, M., Devarakonda, M., Jain, R., Kaplan, M., Pease, D., Rubas, J., Sharma, U., Verma, A., (2005) “Policy-Based Information Lifecycle Management in a Large-Scale File System”. In Proceedings of the Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY’05) pp. 139-148
2. Chen, Y., (2005) “Information valuation for Information Lifecycle Management”. In: Proceedings of the Second International Conference on Autonomic Computing (ICAC'05), pages 135-146.
3. Eickhoff, M.(2002) „Statistische Auswertung und Erkennung der stationaeren Phase in der Simulation zustands-diskreter Systeme. Diplomarbeit, Universität Dortmund, 2002.
4. Gibson, T. and Miller E., (1999) “An Improved Long-Term File-Usage Prediction Algorithm”, 1999.
5. Goepel, T., (2005) “Information Lifecycle Management”, HP Americas StorageWorks Conference 2005, Las Vegas, 2005
6. Gostner, R., Turczyk, L., Berbner, R., Heckmann, O. and Steinmetz, R. (2005): „Analyse von Datei-Zugriffen zur Potentialermittlung fuer Information Lifecycle Management“, TU Darmstadt KOM Technical Report 01/2005
7. Groepl, M., Turczyk, L., Heckmann, O. and Steinmetz, R.: „Statistische Datenanalyse von langfristigem Dateizugriffsverhalten in einer Unternehmensdatenbank fuer ILM“, TU Darmstadt KOM Technical Report 01/2006
8. Jelenkovic P. R. and Radovanovic A., (2004) “Least-Recently-Used Caching with Dependent Requests”, *Theoretical Computer Science*, Vol. 326, Issues 1-3, pp 293-327, October 2004
9. Lawrie, D. H. Randal, J. M. and. Barton, R. R., (1982) “Experiments with Automatic File Migration,” IEEE Computer 15(7), 1982, pp. 45-55.
10. Merrin, R., Harnetty, D., (2005) “Compliance Confusion Set to Drive Market in 2005”, Engenio Information Technologies, http://www.engenio.com/news/uk/2005/02-23-700.html
11. META Group, (2005) META Trends 2005/2006 — Infrastructure, Trends #2059
12. Miller, E. L. and Katz, R. H., (1993) “An Analysis of File Migration in a UNIX Supercomputing Environment”. In: USENIX Winter, pages 421–434, 1993.
13. Peterson, M., (2004) “ILM Definition and Scope - An ILM Framework, SNIA Data Management Forum”, Version 2.3, July 2004
14. Reiner, D. Press, G., Lenaghan, M., Barta, D., Urmston, R., (2004) “Information Lifecycle Management: The EMC Perspective”. Proceedings of the 20th International Conference on Data Engineering (ICDE’04) 1063-6382/04
15. Robinson J. T. and Devarakonda, N. V., (1990) “Data Cache Management Using Frequency based Replacement”. In the Proceedings of the 1990 ACM SIGMETRICS Conference, pages 134-142, 1990.
16. Schmitz, C., (2004) “Entwicklung einer optimalen Migrationsstrategie für ein hierarchisches Datenmanagement System.“ Technischer Bericht, Forschungszentrum Jülich GmbH, 2004.
17. Smith, A. J., (1982) “Long Term File Migration: Development and Evaluation of Algorithms,” Communications of ACM 24(8), 1982, pp.521-532.
18. Strange, S.: 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.
19. Tanaka, T., Ueda, R., Aizono, T., Ushijima, K., Naitih, I., Komoda, N., (2005) “Proposal and Evaluation of Policy Description for Information Lifecycle Management”. Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’05) 0-7695-2504-0/05
20. Turczyk, L., (2005) “Information Lifecycle Management als Weg aus dem Speicherdilemma“. Information Wissenschaft & Praxis, 56 7/2005,
21. Turczyk, L., Berbner, R., Heckmann, O., Steinmetz, R., (2006) “A formal approach to Information Lifecycle Management”. In Proceedings of 17th Annual IRMA International Conference, Washington D.C.