



---

# TECHNICAL REPORT

---

## **NB PILOT: MIGRATION EXPERIMENTS AND RECOMMENDATIONS**



TECHNICAL REPORT: 2009-0913

REVISION No 1

DET NORSKE VERITAS



# DET NORSKE VERITAS



## FINAL REPORT

Date of first issue: <b>08 June 2009</b>	Project No: <b>91303021</b>	DET NORSKE VERITAS AS Research and Innovation
Approved by: <b>Erik Stensrud</b>	Organisational unit: <b>DNV Research and Innovation</b>	C3 1322 Høvik Norway
Author: <b>Thomas Mestl, Feng Luan</b>	Client ref.: <b>DNV</b>	Tel: +47 67579522 Fax: <a href="http://www.dnv.com">http://www.dnv.com</a> NO 945 748 931 MVA
<p>Summary:</p> <p>This report summarizes the results from the migration experiments performed at the National Library of Norway. The objective with the NB pilot was to explore, test and gain as much information about issues related to migration. From the experiments the following conclusions might be drawn:</p> <p><b>MB:</b> The limiting factor in pure file migration is determined by the slowest speed, either slow read speed from the source disc or slow write speed to the target disc. For large data volumes it appears that the practically achievable migration speed is actually given by the worst read/write speed - and NOT the average or highest possible read/write speed.</p> <ul style="list-style-type: none"> <li>• When planning to replace old discs, select new discs that avoid worst write speed.</li> <li>• The lower bound formula for migration time estimation gives reasonable results when using the worst write speed parameter.</li> <li>• It is no point of using multi core processors in situations with plain migration - unless a number of additional target discs are added.</li> </ul> <p><b>MB-V:</b> The time required to perform verification increases considerable for increasing data volume, i.e. 55% and 70%.</p> <ul style="list-style-type: none"> <li>• A reduction in total migration time is obtained by using 2 CPUs – one that is entirely dedicated to file verification. Adding additional CPUs may counteract the observed increase in verification time.</li> <li>• Checksum information and verification can be added without negatively affecting the total migration performance when multiple CPUs are used.</li> <li>• In the experiments, the main time limiting factor was the bad read and write parameters of the target disc.</li> <li>• The lower bound formula gives very good approximations for the expected total migration time when using the worst case read and write parameter for the target disc.</li> </ul> <p>When planning to replace old discs, select new discs that avoid the worst read and write speeds.</p> <p><b>MB-P:</b> In situations where the file conversion is the time dominant factor, adding more CPUs will reduce the required conversion time. The achieved time reduction by parallel multi core processors follows <math>1/n</math>, until <math>n = \# \text{CPUs} - 1</math> (2).</p> <ul style="list-style-type: none"> <li>• The number of the parallel tasks should not exceed <math>\# \text{CPUs} - 1</math> (2).</li> </ul>		

Report No: <b>2009-0913</b>	Subject Group:	Date of this revision: <b>3. June 2009</b>	Revision No: <b>1</b>	Number of pages: <b>16</b>
Report title: <b>NB-Pilot: Experiments and Recommendations (Longrec)</b>				
<p>© 2002 Det Norske Veritas AS All rights reserved. This publication or parts thereof may not be reproduced or transmitted in any form or by any means, including photocopying or recording, without the prior written consent of Det Norske Veritas AS.</p>				



***Table of Contents*** ***Page***

- 1 BACKGROUND..... 1
- 2 PILOT OBJECTIVES ..... 2
- 3 MIGRATION SETUP ..... 3
- 4 THEORETICAL CONSIDERATIONS..... 4
- 5 MIGRATION EXPERIMENTS ..... 5
  - 5.1 Basic Migration (MB): ..... 5
  - 5.2 Migration with Verification (MB-V): ..... 6
  - 5.3 Migration and File Conversion (MB-P): ..... 8
- 6 PILOT RESULTS AND IMPLICATIONS FOR NB ..... 9



# 1 Background

The National Library (NB=Nasjonalbiblioteket<sup>1</sup>) of Norway is considered as the nation’s memory. They store information about Norway’s cultural heritage through a variety of media. NB has approximately 19.028.000 records of books, hand writings, maps, magazines, newspaper, photographs, posters, films and TV recordings, broadcastings, music, web documents etc., adding up to about 42km of shelf space (stand 2008<sup>2</sup>).

In order to make these recordings easier accessible to the public the NB has started an ambitious project of converting (hopefully all) physical records into digital ones. At the end of 2008 they have digitized nearly 1 millions records, or about 5%, and hope to be finished by 2018.

One of their main challenges is given by the sheer amount of the resulting data volume (in addition each record gives another 2 copies for backup). In 2008, they had ca 1000 TB of genuine digital copies and expect an annual increase of 750 TB. As the storage hardware becomes obsolete or the support agreements with storage vendors expire, every 3-4 years the entire data volume has to be migrated over to new storage discs.

If they digitalize all the records, the expected amount of genuine digital data will likely be in the range of 37PB (Peta Bytes!) with 564 million expected number of files. It can easily be envisioned that the previous migration is still not finished when the new migration has to start.

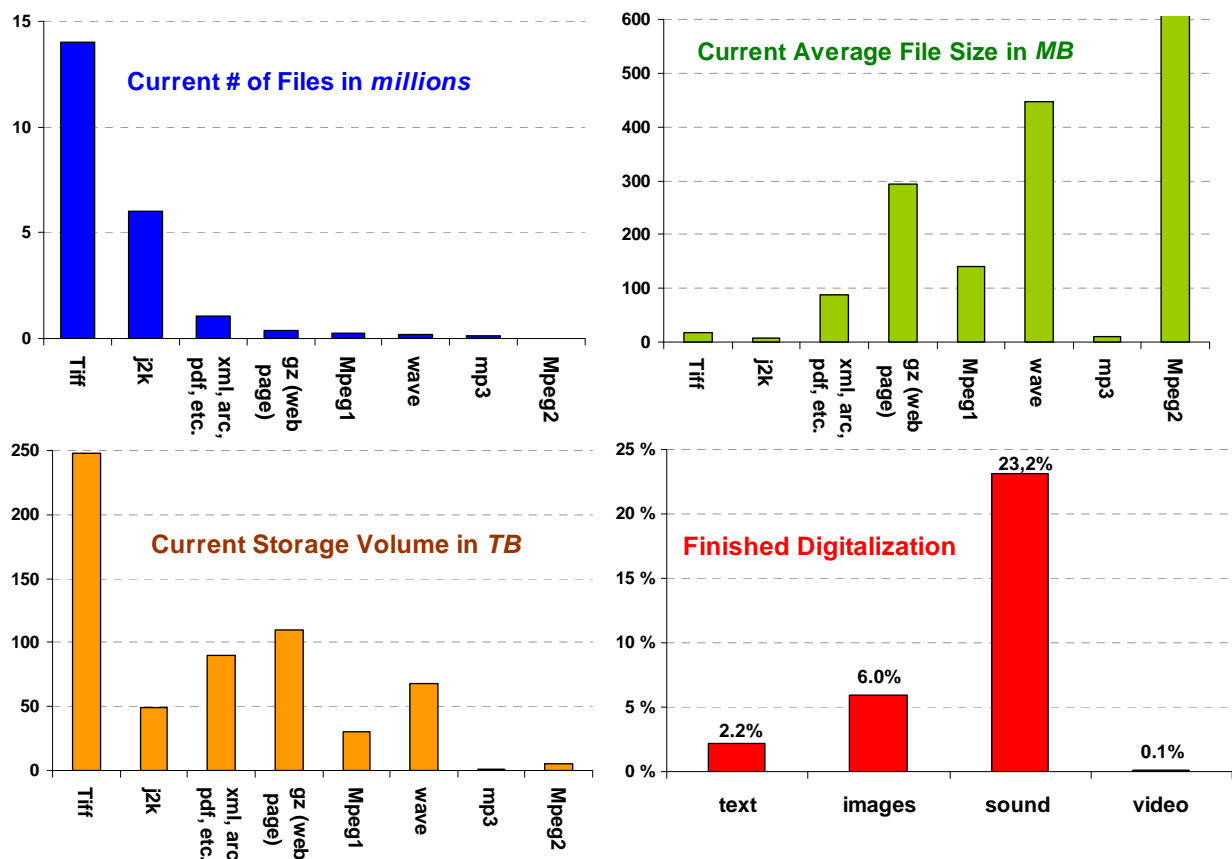


Figure 1: Information about the number of electronic files, their average size and the required storage space of NB’s digital records. In 2008 about 5% of the total physical records were digitized.

<sup>1</sup> <http://www.nb.no/>

<sup>2</sup> From a presentation of Trond Teigen to SUN Microsystems 2008

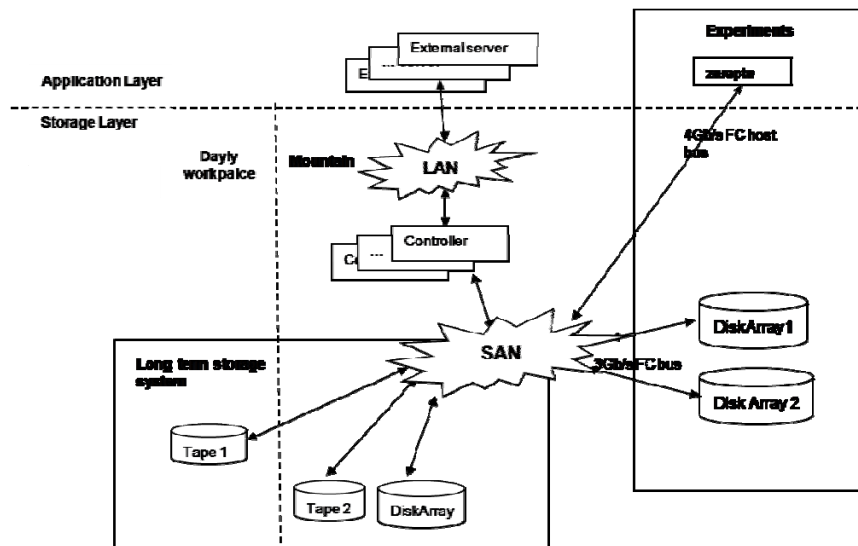


Figure 2: Storage system in the National Library of Norway

Figure 2 visualizes NB's storage system consisting of two tape arrays and one hard-disk array. Each tape array can store up to 100 tapes, whereas the hard disk array uses RAID 5 to achieve high enough data redundancy and better I/O performance. All storage media are connected and managed under a storage area network (SAN).

## 2 Pilot Objectives

The objective with the NB pilot was to explore, test and gain as much information about issues related to migration such as:

- Investigate the feasibility and reliability of the setup and migration
- Describe the process with focus on automation and migration verification
- Verification both at process level and file level
- Identify different types of tests to secure data integrity
- Consider the different types of tests and the suggested testplan and make a new testplan

The practical and resource constraints the migration tests were performed by LongRec's PhD student Feng Luan from the NTNU.

### 3 Migration Setup

The migration experiments were carried out in the SAN environment (see box named *experiments* in Figure 2) which included one server, i.e. Zarepta, and two folders in two different hard disk arrays, i.e. DiskArray1 and DiskArray2. The task of Zarepta is to manage those two hard disk arrays. Zarepta has 4 CPUs and use Redhat Enterprise Linux with kernel version 2.6.18-128.e15 as its operating system. It directly connects to the SAN via a fibre bus with 4GB/s transfer bandwidth.

Each hard disk array is a Sun StorageTek 6140 Array<sup>3</sup> were several 1TB hard disks with 7200 rpm were added to obtain a RAID5. They can also connect to SAN via a 3GB/s fibre bus.

In order to be able to compare various migration setups benchmark data were collected for each hard disk array by Bonnie++1.95<sup>4</sup>. Bonnie++1.95 was run 10 times on each hard disk array, the benchmark results are shown in Table 1.

**Table 1. Benchmark of read and write speed to and from the storage media**

	Read in MB/s			Write in MB/s		
	Best	Worst	Average	Best	Worst	Average
<b>DiskArray1</b>	300.8	220.73	264.32	131.78	89.88	112.05
<b>DiskArray2</b>	301.67	209.06	239.9	128.2	96.196	116.44

The migration was done from DiskArray1 to DiskArray2. Three different migration schemes were set up.

- **Basic Migration (MB):** 12 folders each containing 100 video files á 432MB shall be migrated. In the experiments different volumes of data will be copied by using 1 and 2 CPUs respectively.
- **Migration with verification (MB-V):** A verification process was evoked after the copy process has finished. The verification program that was used was a Linux process named md5sum that uses MD5 to verify a specified file's integrity.
- **Migration with file processing (MB-P):** 200 video files (same mp4-file format, same file size of 432MB) underwent a file conversion with a program called ffmpeg<sup>5</sup>. File conversion speed was approximately 1.4MB/s.

<sup>3</sup> [http://www.sun.com/storage/disk\\_systems/midrange/6140/specs.xml](http://www.sun.com/storage/disk_systems/midrange/6140/specs.xml)

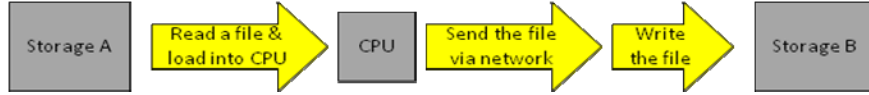
<sup>4</sup> <http://www.coker.com.au/bonnie++/>

<sup>5</sup> <http://www.ffmpeg.org/>

## 4 Theoretical Considerations

Feng & Mestl derived a framework for estimating the migration time based on a detailed process description from which the dominating migration time parameters could be derived. This framework gives mathematical expressions for lower and upper bounds of the expected file migration time, for more details see <sup>6</sup>.

For a simple file migration (MB) the process is shown in the figure below



which would give the following expressions for lower and upper bounds for the expected migration time respectively:

$$T_{MB}^L = \max\left(\frac{1}{B_R^A}, \frac{1}{B_N^B}, \frac{1}{B_W^B}\right) \cdot S \quad \text{and} \quad T_{MB}^U = \left(\frac{1}{B_R^A} + \frac{1}{B_N^B} + \frac{1}{B_W^B}\right) \cdot S$$

where e.g.  $B_R^A$  is the read speed from DiscArray 1. Subscripts  $R$ ,  $N$  and  $W$  indicates read, networking or write, whereas superscripts  $A$  and  $B$  indicates DiscArray 1 or 2 respectively.  $S$  is total data volume that is read from the source disc and  $S'$  is the data volume written to the target disc (except in case of file conversion  $S=S'$ ).

Similar expressions can be derived for migration with verification:

$$T_{MB-V}^L = \max\left(\frac{1}{B_R^A}, \frac{1}{B_N^B}, \left(\frac{1}{B_W^B} + \frac{1}{B_R^B}\right), \frac{1}{B_V^B}\right) \cdot S \quad T_{MB-V}^U = \left(\frac{1}{B_R^A} + \frac{1}{B_N^B} + \frac{1}{B_W^B} + \frac{1}{B_R^B} + \frac{1}{B_V^B}\right) \cdot S$$

and migration with file conversion:

$$T_{MB-P}^L = \max\left(\frac{1}{B_R^A}, \frac{1}{B_P^A}\right) \cdot S \text{ or } \max\left(\frac{1}{B_N^B}, \frac{1}{B_W^B}\right) \cdot S' \quad T_{MB-P}^U = \left(\frac{1}{B_R^A} + \frac{1}{B_P^A}\right) \cdot S + \left(\frac{1}{B_N^B} + \frac{1}{B_W^B}\right) \cdot S'$$

<sup>6</sup> Feng Luan, Thomas Mestl. A Mathematical Framework for Modeling and Analyzing File Migration Time. 2009, to be submitted.

## 5 Migration Experiments

Three migration experiments were done at the National Library.

### 5.1 Basic Migration (MB):

In this test case, a series of folders each containing 100 video files á 432MB each, was migrated from DiscArray 1 to DiscArray 2. The migration time from experiments and from the formula are shown in Table 2. For computation of the lower / upper bounds the average values for the read / write speed for the DiscArrays 1 / 2 were chosen.

**Table 2: Measured and estimated migration time for basic migration (MB). Time and error estimation valid for 1 CPU situation.**

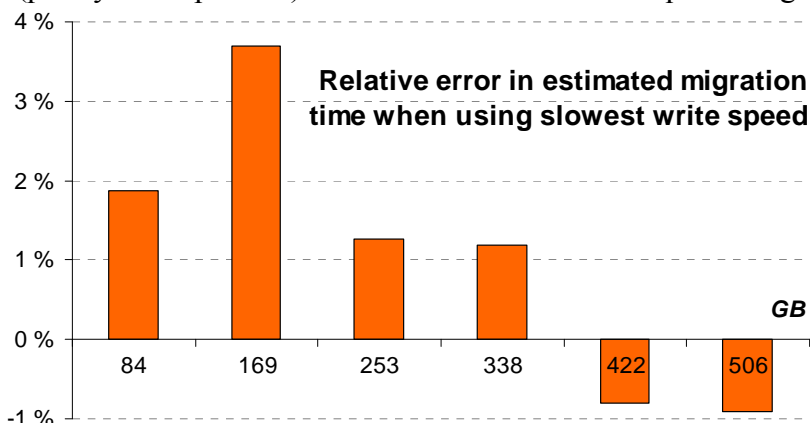
# folders	# files	Total data volume in MB	1 CPU: measured in s	2 CPU: measured in s	Upper bound in s	Rel error in %	Lower bound in s	Rel error in %
2	200	86400	<b>881.7</b>	890.3	<b>1090</b>	+23.6%	<b>742</b>	-15.8%
4	400	172800	<b>1732.3</b>	1580.7	<b>2180</b>	+25.8%	<b>1484</b>	-14.3%
6	600	259200	<b>2661</b>	2679.3	<b>3270</b>	+22.9%	<b>2226</b>	-16.4%
8	800	345600	<b>3550.7</b>	3652.7	<b>4360</b>	+22.8%	<b>2968</b>	-16.4%
10	1000	432000	<b>4527.3</b>	4530.7	<b>5450</b>	+20.4%	<b>3710</b>	-18.0%
12	1200	518400	<b>5438.3</b>	5338.7	<b>6540</b>	+20.3%	<b>4452</b>	-18.1%

As expected, the lower bound underestimates the migration time whereas the upper bound overestimates the required migration time. Interestingly, the total migration speed is NOT significantly reduced by using an additional CPU.

For calculation of the lower bound in Table 2, the average write speed of 116.44MB/s was used resulting in a relative error between 14% and 18%. When using the worst write speed from the benchmark measurements, i.e. 96.196MB/s, the relative error becomes much lower, i.e. between -1% and 4%, Figure 3. As the error is both  $\pm$  it can not be used as a lower bound unless for large data volumes. In fact it gives an indication that DiscArray 2 actually works near its worst performance. It may therefore be concluded that the limiting factor in the overall migration speed was mainly determined by the (worst) write speed of DiscArray 2.

In a real situation, DiscArray 2 would correspond to the new storage media. In order to plan for a low overall migration time, then a discarray should not be selected on the basis of its best or average write speed, but focus should be on avoiding the worst write speed.

It may be worth mentioning that by using the best read & write speeds the upper bound (purely serial process) would underestimate the required migration time.



**Figure 3: Relative error in the 'Lower bound' for the migration time when using the worst write speed of DiscArray 2 (target storage).**

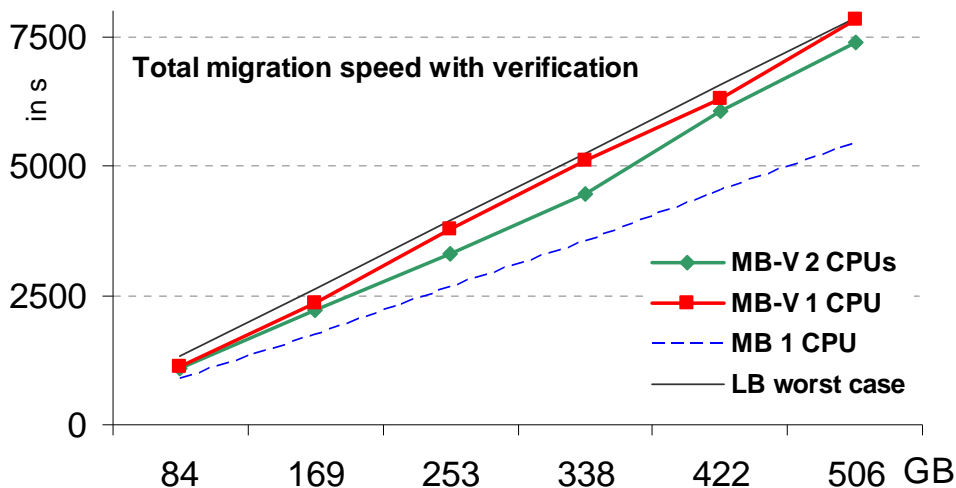
## 5.2 Migration with Verification (MB-V):

In the next experiment, the file migration was combined with a verification process, i.e. after the copy process of a specific file has finished its verification starts. The verification program that we use is a Linux process named md5sum, which use MD5 to verify a specified file's integrity. The experimental results are listed in Table 3.

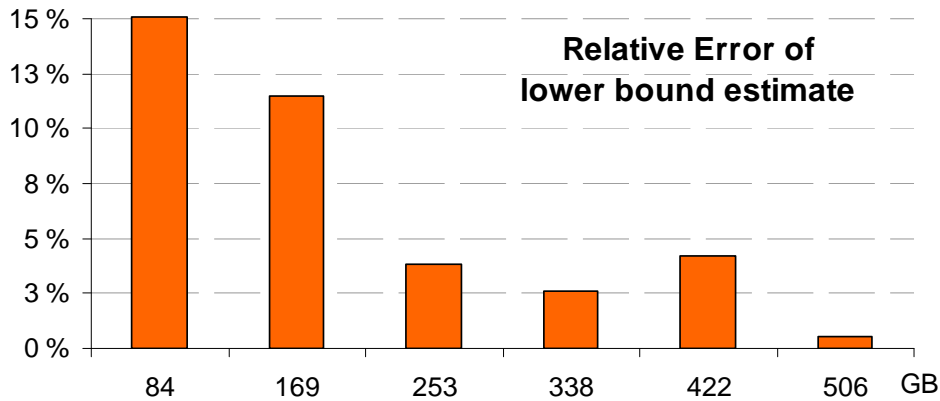
**Table 3: Measured and estimated total migration time when verification is included. Time and error estimation valid for 1 CPU situation.**

# folders	# files	Total data volume in MB	1 CPU: measured in s	2 CPUs: measured in s	Upper bound in s	Rel error in %	Lower bound in s	Rel error in %
2	200	86400	<b>1139.3</b>	1092.7	<b>2325</b>	+104.1%	<b>1102</b>	-3.7%
4	400	172800	<b>2352.7</b>	2198.0	<b>4650</b>	+97.6%	<b>2204</b>	-6.3%
6	600	259200	<b>3789</b>	3309.3	<b>6975</b>	+84.1%	<b>3306</b>	-12.7%
8	800	345600	<b>5114.3</b>	4469.3	<b>9300</b>	+81.8%	<b>4409</b>	-13.8%
10	1000	432000	<b>6291.3</b>	6066.7	<b>11624</b>	+84.8%	<b>5511</b>	-12.4%
12	1200	518400	<b>7825.3</b>	7394.0	<b>13949</b>	+78.3%	<b>6613</b>	-15.5%

The upper bound overestimates the expected total migration time due to the assumption that all processes are serial. The lower bound (based on average speeds) is quite close to the actual total migration time for small data volumes, but it increasingly underestimates the required time for larger data volumes. Figure 5 depicts the total migration time (red and green line with 1 and 2 CPUs respectively) together with the lower bound with the worst read and write speed for DiscArray 2 (grey line). Again, a closer examination of the lower bound reveals that a very good approximation for the expected total migration time is achieved when using the worst case read and write parameter for DiscArray 2, Figure 5.



**Figure 4: Total migration time with verification (MB-V) in dependence on 1 CPU (red line) and 2 CPUs (green line). For comparison, basic migration MB 1 CPU is given by the blue dotted line. The difference between the blue and red line gives the time required for verification. The lower bound with worst read and write speed for DiscArray 2 gives a good approximation when 1 CPU is used.**

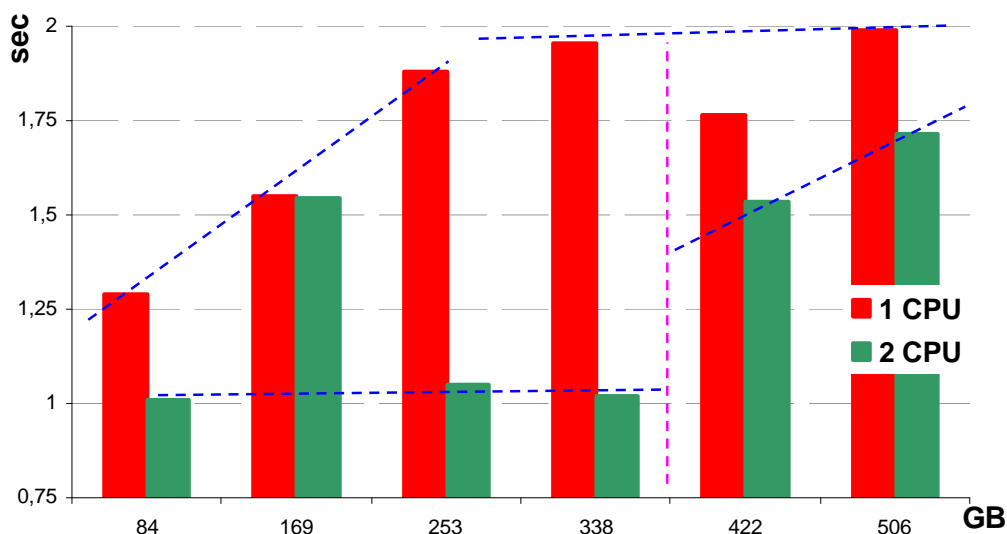


**Figure 5: The relative error of the lower bound when using the worst read and write performance of DiskArray 2. The file is first written to DiskArray 2 (target storage) and then read from it in order to be verified.**

The Zarepta has 4 parallel processors, consequently when invoking one copy process, the copy process and the md5sum verification process will not interrupt each other. However, if two parallel copy processes are started, then at least two md5sum processes will be invoked resulting in a load of 4 parallel tasks in Zarepta. It seems that with this load Zarepta reaches its limit leading to an increase in total migration time.

This becomes obvious when subtracting the pure migration times (from the MB experiments) from the MB-V time data resulting in an approximation for the required verification time (in dependence on the number of CPUs). Figure 6 shows that the average verification time per file increases quite considerably with growing data volumes, i.e. 55% and 70% for one or two CPUs respectively. It appears that it levels off around 2 s/file (blue dotted lines in the figure). When using 2 CPUs this increase does not happen gradually, but occurs first after a certain load (pink dotted line) which actually marks the saturation load when just using one CPU.

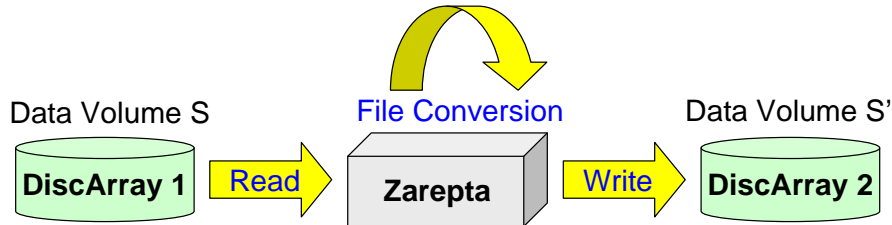
The spike at 169 GB load in the 2 CPU situation equals that to the 1 CPU situation. This might be explainable if some other background process effectively blocks the second CPU from performing the verification task, effectively reducing the 2 CPU experiment to one with 1 CPU. Unfortunately, no information was collected about other ongoing background processes in Zarepta.



**Figure 6: The average verification time per file. As the data volume increases so does the required verification time/file (55% and 70% increase for 1 or 2 CPUs respectively). It appears that the verification time levels off at around 2 s/file at high data volume.**

### 5.3 Migration and File Conversion (MB-P):

200 video files (same *mp4*-file format, same file size of *432MB*) underwent a file conversion with a program called *ffmpeg*<sup>7</sup> before migrating them to their new storage site (DiscArray 2). It is difficult to benchmark the conversion program as software performance depends on both hardware configuration and the software algorithm. It was measured to be in the range of *1.4MB/s*.



**Figure 7:** Files were read from DiscArray 1 into Zarepta which performed a conversion into a new file format (resulting in the reduction of the total volume of data from  $S \rightarrow S'$ ) which is then written to DiscArray 2.

As the conversion reduces the file sizes the experimental results cannot be compared to the measurements from the MB experiment. The results from the file conversion experiment are given in Table 4.

**Table 4. Migration and conversion time (MB-P) in seconds.**

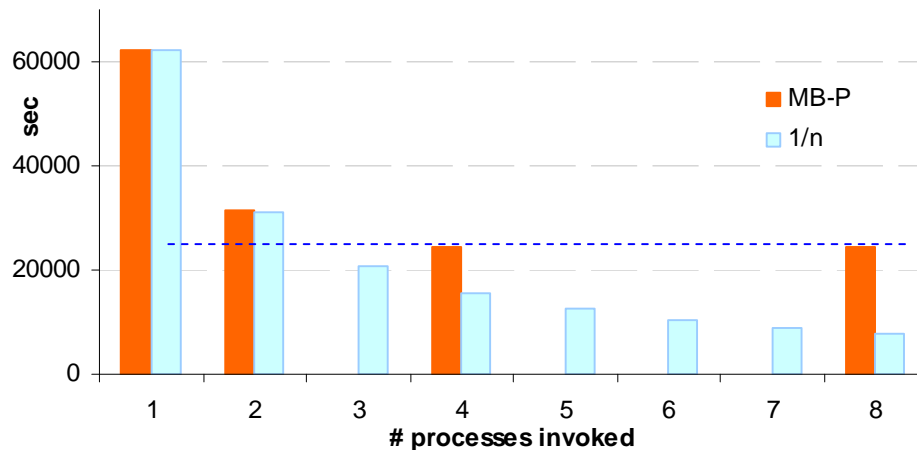
# files	Volume before conversion in MB	Volume after conversion = before migration in MB	# of conversion + migration tasks invoked	Total migration time in s	Upper bound in s	rel error in %	Lower bound in s	rel error in %
200	86400	39000	1	<b>62299.5</b>	62386	0.1%	61714	-0.9%
			2	<b>31518.6</b>				
			4	<b>24379.4</b>				
			8	<b>24554.9</b>				

The first noticeable thing in Table 4 is the very good approximation for the total migration time given by both the lower and upper bound formula. The reason for that is that the very slow conversion speed of *1.4Mb/s* completely dominates all the processes involved, i.e. the read, write or transfer speed do not play a significant role in this setup (conversion time  $\approx 310s$ , read  $\approx 1.6s$ , write time  $\approx 1.7s$  per file).

Observe also, when invoking more migration tasks (= conversion + migration) then the total migration time goes down until it levels off. This phenomena is easily explainable by the fact that Zarepta's 4 CPUs are increasingly involved with (mainly) file conversion. However, as the number of invoked tasks comes close to the number of CPUs the effect of other background processes will steal considerable processing power, see Figure 8. Invoking more tasks will therefore not lead to a further reduction in conversion time.

It might therefore be deduced that the gain from parallelization by multi core processors follows  $1/n$  (light blue columns in Figure 8), until  $n = \text{number of CPUs} - 1$  (or 2) when background tasks level performance off (blue dotted horizontal line).

<sup>7</sup> <http://www.ffmpeg.org/>



**Figure 8: Migration time (mainly file conversion) as a function of number of conversion processes invoked. Already with 3 parallel conversion processes the limits of Zarepta's 4CPUs are reached (blue dotted line) as background processes steal considerable processing power.**

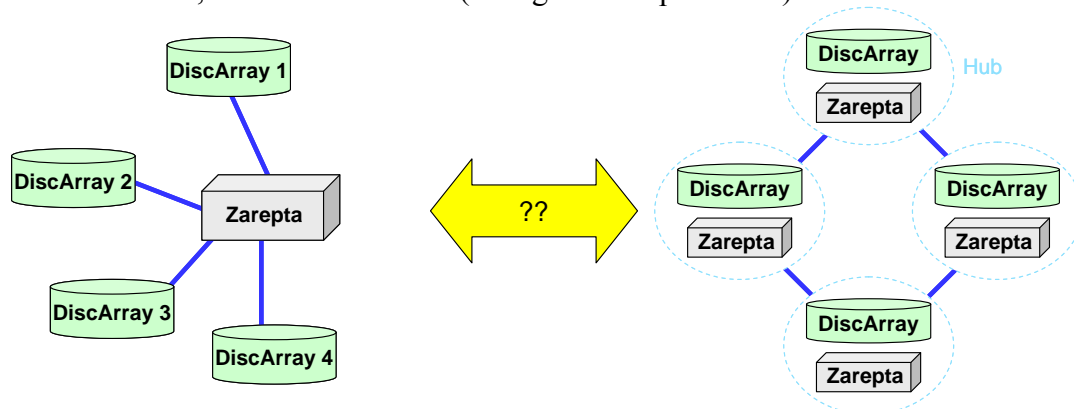
## 6 Pilot Results and Implications for NB

From the experiments the following conclusions might be drawn:

- MB:** The limiting factor in pure file migration is determined by the slowest speed, either slow read speed from the source disc or slow write speed to the target disc. For large data volumes it appears that the practically achievable migration speed is actually given by the worst read/write speed - and NOT the average or highest possible read/write speed.
- ⇒ When planning to replace old discs, select new discs that avoid worst write speed.
  - The lower bound formula for migration time estimation gives reasonable results when using the worst write speed parameter.
  - It is no point of using multi core processors in situations with plain migration - unless a number of additional target discs are added.
- MB-V:** The time required to perform verification increases considerable for increasing data volume, i.e. 55% and 70%.
- A reduction in total migration time is obtained by using 2 CPUs – one that is entirely dedicated to file verification. Adding additional CPUs may counteract the observed increase in verification time.
  - ⇒ Checksum information and verification can be added without negatively affecting the total migration performance when multiple CPUs are used.
  - In the experiments, the main time limiting factor was the bad read and write parameters of the target disc.
  - The lower bound formula gives very good approximations for the expected total migration time when using the worst case read and write parameter for the target disc.
  - ⇒ When planning to replace old discs, select new discs that avoid the worst read and write speeds.
- MB-P:** In situations where the file conversion is the time dominant factor, adding more CPUs will reduce the required conversion time. The achieved time reduction by parallel multi core processors follows  $1/n$ , until  $n = \# \text{CPUs} - 1$  (2).
- The number of the parallel tasks should not exceed  $\# \text{CPUs} - 1$  (2).

### Hubs vs. central multi-core processor:

A central issue for NB is the question of storage topology. Should a central multi-core processor serving a series of storage arrays be the preferred solution, or shall rather a solution with interconnected, decentralized hubs (storage + own processor) be chosen.



**Figure 9: Storage topology: centralized processor (left) or autonomous, de-centralized hubs (right). A multi core, centralized processor can effectively transform the system into a parallel system equivalent to a de-centralized hub.**

Based on the migration experiments the following implications for the NB may be drawn:

- As the write speed is the time limiting factor in migration, a single central CPU could easily serve multiple target storage discs (equipped with buffers) to decrease migration time by parallelization of read / write tasks. The network bandwidth with 3-4GB/s is about 24-40 times faster than the write speed. This means that one CPU could potentially serve  $(24-40)/2 = 12-20$  discarrays before the CPU becomes over loaded. This is however only valid if no other background processes interferes with the file migration, i.e. they are handled by a different CPU.
- Migration time will not increase if verification is done by hashing MD5 checking on a separate CPU.
- No migrated files becoming corrupted or displaying bit errors have been encountered so far, even though they have been migrated several times. On the other hand, no systematic search has been done and only a small fraction of the migrated volume has been accessed so far.
- In the experiments the discarrays had approximately equal performance. In a real situation, DiscArray 2 would correspond to the new (target) storage discs. As the old storage (DiscArray 1) is given, the new discarray should be selected based on its write performance (just migration), or based on both the write and read performance (migration + verification). It is not the best or average write speed that counts but its worst read / write performance.
- If file conversion is slow, additional computation power may be added to achieve an effective file conversion speed - preferable in the same range as the read/write speed. (this would however correspond to adding about 180 additional CPUs for MB-P).
- NB is currently logically migrating about 12 mill files (ca 700-800 TB) from tiff-format to the new jpg2000 format with an inclusion of colour correction processing (required processing time is around 5 s/file), and then physically migrating them to a new disc store. If the storage discs are similar to DiscArray 2, then the migration time alone would be between 90 and 101 days. File conversion and processing would require additional 695 days. Using at least 9 parallel or multi-core processors (for conversion and processing) would reduce the overall time to approximately 80 days, i.e. the whole project could be finished within 0.25 years.