UNIVERSITEIT VAN AMSTERDAM

System and Network Engineering

Research Project 2

Concept
SAN Health Status Monitor

By:
Yanick de Jong - *yanick.dejong@os3.nl*
Adriaan van der Zee - *adriaan.vanderzee@os3.nl*

Supervisor (KLM IS):
L.H.M. Gommans

Version: 1.0 (Final)

Date:
2009-07-06

Abstract

This project, the result of a four week research project, delivers a concept Fibre Channel SAN health status monitor. Problem indicator from different sources and of different nature are combined into a relational model, and status levels have been defined to represent the health of a redundant storage infrastructure continuously, as well as over longer time periods.

The scope of current SAN monitoring tools has been broadened to include information from hosts that connect to a Fibre Channel fabric. The relational model combines problem indicator subjects from hosts, fabric switches, and storage systems.

Problem indicators and relations between them lead to different health status levels, which can be represented continuously as well as historically.

Next steps towards the realisation of a SAN health status monitor are a proof of concept implementation, and further development of a translation scheme between problem indicators and their relations.

Acknowledgements

During our four weeks at KLM IS we have had formal meetings and casual conversations with many people from a variety of departments. We would like to thank all of them for their extensive cooperation; we have learned a great deal from them about the storage infrastructure and their problems.

Special thanks go to:

- Leon Gommans, for his supervision and extensive feedback on the early versions of this report.
- Bert Koldijk, who helped us with accessing and understanding current SAN monitoring sustems, and who attended our final presentation in Leon's absence.
- Tim Valkenburg, who invited us to the Intensice Care Infrastructure Taskforce meetings, and gave us time for a presentation of our work.
- Erik Breems, our manager, who let us present our work to the SAN people.

Responsible for coordinating all SNE RP2 projects, and publishing this particular one, is Cees de Laat from the UvA.

Table of contents

Glossary

DCB Error	I/O error on IBM AIX operating system
Fabric	Switched Fibre Channel network
Fibre Channel	Network technology, used for remote block level data access
HBA	Host Bus Adapter; Fibre Channel interface card
ISL	Inter Switch Link; connection between two Fibre Channel switches
KLM IS	KLM Information Systems
LUN	Logical Unit Number; remote hard disk volume
LVM	Logical Volume Manager
SAN	Storage Area Network
SDD	Subsystem Device Driver; IBM implementation for multipath storage access

1 Introduction

This report is the result of the final of two four-week research projects that are part of the one-year curriculum of the master study System and Network Engineering from the University of Amsterdam (UvA). This research project has taken place at KLM Information Services, at the Central Systems department, which has several sub-departments that manage different server and storage hardware platforms. The goal of this research was to propose a concept for a SAN health status monitor that has a broader scope than the current monitoring systems in use, and should therefore gives better insight into storage-related problems and their impact.

1.1 Background information

Virtually all IT related products and services that are in use by KLM are being provisioned and operated by the Information Systems (IS) division. Within Central Systems department (part of Operations) different hardware platforms are managed by different groups. The Storage Area Network (SAN) is managed by the SAN group, and the UNIX and Linux groups are each responsible for systems running the respective operating system. All three groups mentioned have been involved with this research project.

Storage provided by the SAN group is being used by a variety of different servers, running different applications on different operating systems. The SAN infrastructure is based on redundant Fibre Channel fabrics that cross-connect to redundant storage servers. This redundant setup should ensure that failures can be isolated to one half of the infrastructure, while the full load can be taken by the non-affected half of the infrastructure.

In order to monitor conditions that may lead to- or indicate a fail-over situation between redundant parts of the SAN infrastructure, the SAN group of KLM IS has asked for a study that will result in a conceptual design of a SAN Health Status Monitor (HSM).

1.2 Problem definition

Although hardware monitoring of the different SAN components is already in place to support day-to-day SAN infrastructure administration, there is no general health status monitor that combines different monitoring input and tracks trends over a longer period of time. Furthermore, SAN failure indicators that are being detected on other systems, managed by for example the UNIX and Linux groups, can lead to better insights in the cause of problems, when this information is combined with failure indicators from within the SAN.

The ultimate goal of the SAN health status monitor is twofold. Firstly, the combined monitoring input should lead to immediate and accurate problem analysis in case of major infrastructure failures, and secondly, trends of minor and/or intermittent problems should be taken into account to define a long-term health status. This long-term health status is meant to help the SAN group take measures to prevent future infrastructure failures. This project discusses the concepts of a system, and will support the potential implementation at a later stage.

1.3 Research questions

How can an alarm system be created that monitors the long term as well as immediate health of a fibre channel fabric?

- What indicators are relevant for the health of the fibre channel fabric, and where can they be found?
- What are the important interrelations between such indicators, and how can they be quantified?
- What kind of health status levels can be defined, and by which indicators and thresholds should they be reached?
- What long term and/or immediate actions should be taken upon the different health status levels, by whom, and what are the priorities?

1.4 Outline of this report

Section 2 is about the different components that are involved in a Fibre Channel based storage infrastructure. For each component group the functions and interactions with other components are explained. This ultimately leads to a relational model that is presented in section 2.2.

In section 3 the different problem conditions are explored, and indicators are determined that can trigger the problem conditions. Different health status levels that have been defined are discussed in section 3.3, and in section 3.4 the problem indicators are being related to each other.

Section 4 explains how the different health status levels can be presented, both in (near) real time and historically.

2 The storage infrastructure

Many of the KLM IS servers – or in storage terms more commonly called hosts – connect to centrally managed storage clusters over a switched Fibre Channel network. Hosts access the storage as if it were local disks. Redundant hosts and storage clusters are split over multiple locations to increase availability, and two independent fabrics (Fibre Channel networks) connect the hosts and storage clusters over the different locations (see figure 1). A host can mirror its data over separate storage clusters, and each copy of the data can be reached over physically separate paths.

This section of the report discusses how the three main components – hosts, switches and storage clusters – function and interact with each other, especially with regard to storage access.

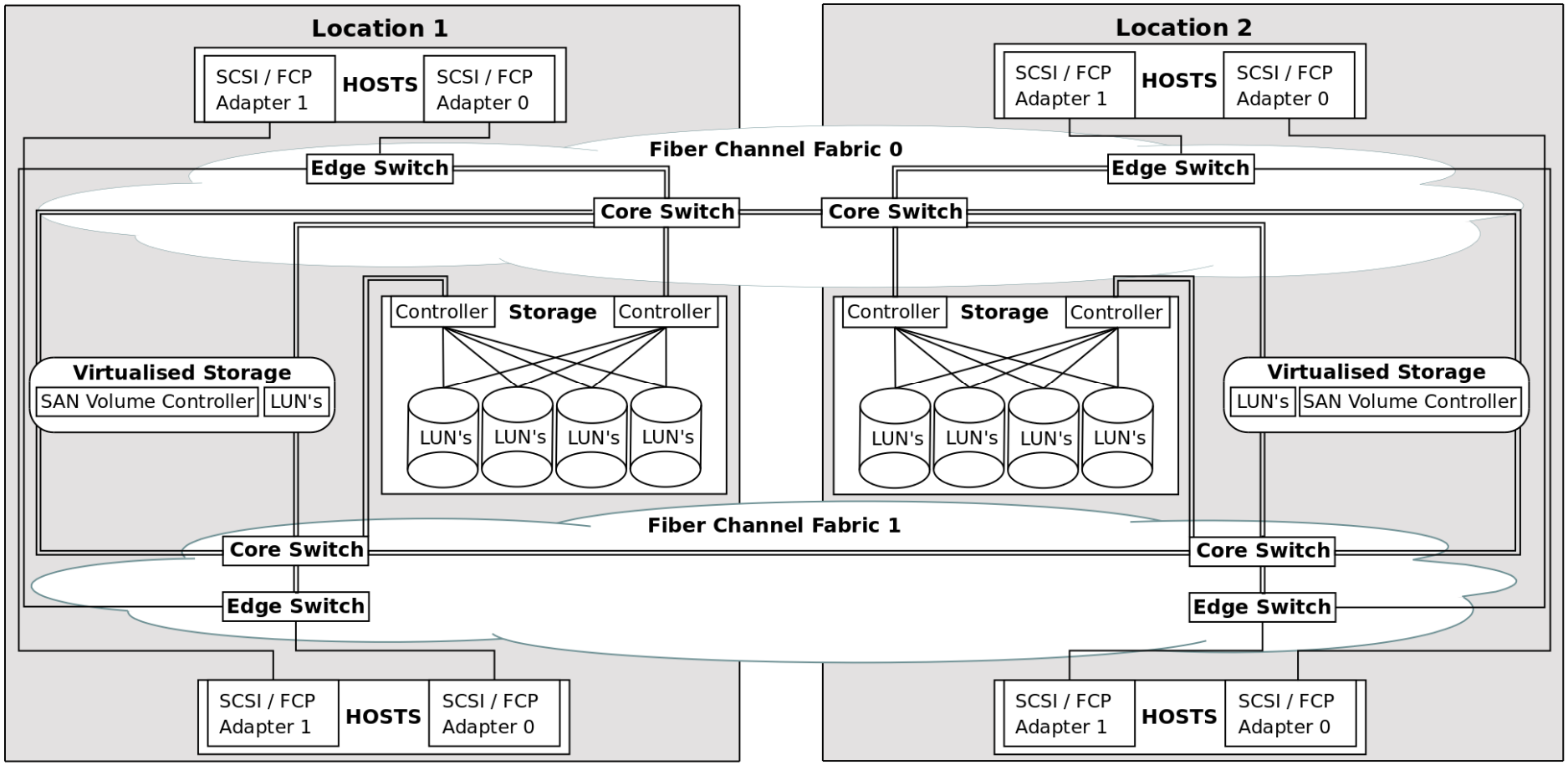


Figure 1 - Physical view of the storage infrastructure

2.1 Components

The ultimate goal of this project is to create a conceptual design of a SAN health status monitor which combines information gathered from different sources involved with storage. In order to relate information sourced from the different components, it is important to first understand how hosts, switches and storage clusters are involved with storage individually, and how they interact.

2.1.1 Hosts

The SAN group is responsible for managing the SAN fabrics and the storage clusters. Hosts, however, have mostly been left out of scope from the SAN group, as they are being managed and monitored by different groups within KLM IS.

Hosts come, within KLM IS, in many different flavours. Different versions of Windows, Linux and UNIX and mainframe platforms are being deployed. There are differences and similarities in the way these different platforms access storage. As only four weeks have been allocated for this project the main focus has been on the similarities between the way UNIX and Linux use storage. Implementation specific details have been researched superficially.

A common way in which hosts are connected to their storage is by means of two separate Host Bus Adaptors (HBAs), each connecting to the separate fabric (see figure 1). HBAs can have multiple ports, in some cases multiple hosts (i.e. in a blade enclosure) may share common multi-port HBAs, and in other cases a single host could have four or more HBAs for increased performance. All these cases can be represented by the model from section 2.2, but the general idea is that a host is connected to two independent fabrics, which enables it to access the storage via separate redundant paths. During normal operation hosts use all available ports in a round robin or load balanced fashion.

As already mentioned storage clusters have also been set up redundantly (see figure 1). Data is to be mirrored over the two storage clusters by the hosts. Hosts access data by means of Logical Unit Numbers (LUNs), which are in fact remote volumes that are being represented to the host's operating system as local disks. By making a LUN available on each of the redundant storage clusters, data can be mirrored by the host, without the storage system being actively aware of it. It is possible that multiple hosts share the same LUN, even though only one host at a time will have write access to a shared LUN.

Interviews have been conducted with both the UNIX and Linux groups to find out the commonalities and specifics of the systems with regard to storage access and monitoring. Although there are differences in implementations, most of the status information is similar. See section 3.4.1 for more information about the status information that can be gathered from the Linux and UNIX hosts. Appendix I provides more details on the Linux and UNIX specific implementations for storage access and management.

2.1.2 Switches

Fibre Channel switches connect the hosts to the storage. As can be seen in figure 1 from section 2, two separate fabrics exist. Each of these fabrics consists of two core switches on different locations, which each connect to a number of edge switches.

Although exceptions exist, generally a host connects to edge switches, and storage to the core switches. The two-tier design, with a pair of connected core switches, means that there is a maximum of four systems between any two systems on the fabric (an edge switch, the two core switches, and another edge switch).

Inter Switch Links (ISLs) are the connections between the switches. As the ISLs need to transport the aggregate data of the host and storage ports, multiple physical ISLs are being used in parallel. Up to eight physical ISLs can be grouped into a trunk that is configured as a single logical link. Multiple trunks may exist in parallel.

In principle hosts and storage clusters connect to both fabrics, creating multiple paths between any host and storage cluster pair. The two fabrics, however, are not interconnected (or a single fabric rather than separate ones would be the result).

2.1.3 Storage clusters

KLM IS has several storage systems connected to the SAN fabric. Different types of disk storage clusters as well as (tape) archives make use of the SAN infrastructure. This project focuses on the disk storage systems, as these are the systems (hosts and their applications) depend on the most.

Two types of disk storage can be seen in figure 1 from section 2: virtual storage, and direct storage. For host access the difference between the two is transparent, but the way they make use of the fabric is different. In both cases a host accesses a storage controller, which interacts with the disks, so the disks are logically located behind the controller. In the case of the non-virtualised storage, this location is also physical, but with the virtual storage the disks are connected to the same fabric switch parallel to the controller. This means that each block of data accessed through the virtual storage controller travels through the fabric switch two times: from the disk to the controller and from the controller to the host (based on a read action). Even though the Fibre Channel switches are supposed to be able to forward traffic on all ports at wire speed, the fact that data passes a switch twice could cause additional impact if performance issues arise.

Something that is common to all disk storage systems is that a LUN (seen by a host as a volume) belongs to one of several storage subsystems. These storage subsystems are connected to the fabrics with multiple ports. In theory a LUN can be made available via all ports of the subsystem, but most commonly this number is limited to just two, one port on each of the fabrics. Due to the nature of Fibre Channel, for each fabric port over which a LUN is available, the host will see a separate path, even if the two ports are connected to the same fabric.

2.2 SAN component relation model

One of the targeted strengths of this project's SAN health status monitor is to relate problem indicators from the different component groups, as discussed in section 3.4.1 through 3.4.3. For this purpose a relational model has been created, depicted in figure 2. This model makes it possible to define the possible influences of a problem detected on one entity, or to verify if a causal relation could exist between two failing entities. More about how possible problems can be related is described in section 3.5.

This model displays subjects of problem indicators, which can be both logical and physical. Another example where logical and physical attributes are combined in a single hierarchical model is the Management Information Base (RFC 2578), which is discussed in more detail in appendix II.

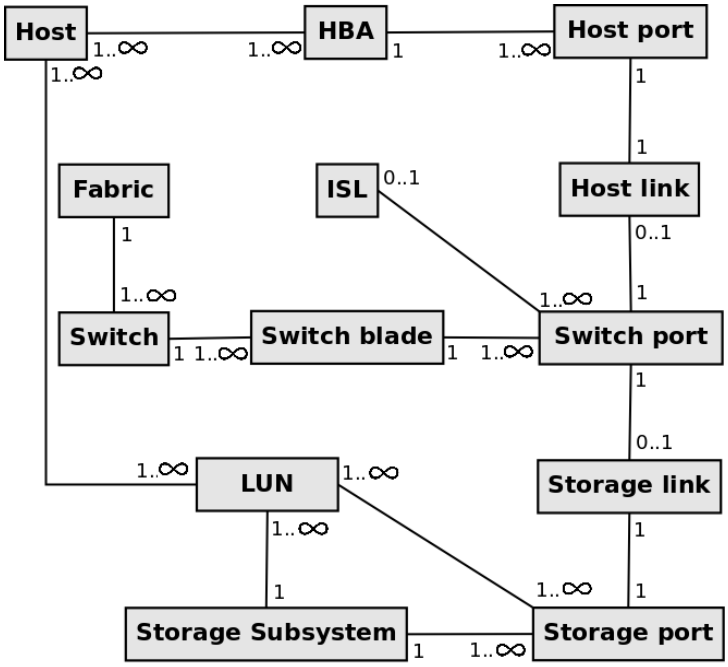


Figure 2 - Storage components logical model

The relations and constraints are as follows:

- One or more hosts can share one or more HBAs, and each HBA can have one or more host ports connected to a switch port. Such a connection is a host link.
- One or more hosts share one or more LUNs.
- A fabric consists of one or more interconnected switches and includes all connected host ports and storage ports as well.
- A switch has one or more switch blades, which each contain one or more switch ports.
- An ISL is a link that connects a switch port to a switch port from another switch, both switches are by definition in the same fabric.
- A storage subsystem contains one or more LUNs which can be made available via one or more storage ports that are connected to a switch port. Such a connection is a storage link.

3 Problem conditions

A problem with the storage infrastructure can be defined as a situation where a component on the path between storage and host has a negative influence on the accessibility of the storage by hosts. There are two possible causes for such a situation: hardware failures, or capacity shortages. The latter can be caused by the former, for example when a failed redundant component causes extra load on the component that took over.

Section 3.1 and 3.2 discuss the causes and effects of hardware failures and capacity shortages respectively.

3.1 Hardware failure

Any piece of hardware can fail, but the impact can differ greatly. For example, a switch contains two Central Processors (CPs), of which one is a hot stand-by. If one fails the other takes over, and as the CP taking over was idle and physically identical to the failing one, it can guarantee to be able to take the same load. If, however, a single host port fails, there is no hard guarantee that the corresponding host port on the other fabric can handle the load that was previously being balanced over the two host ports.

Sound capacity management, planning, and testing should make sure that redundant components are really redundant, but only a real incident will tell whether this was the case or not. Therefore it is important for the SAN health status monitor to register hardware failures, and monitor its effects on related components. In addition, failed hardware, even if it does not impact the performance of storage access by hosts, still reduces the designed redundancy of the system, making it more vulnerable for subsequent problems.

Several levels of redundancy can be seen in the storage infrastructure. System redundancy across sites and fabrics has already been discussed. On a smaller level hardware component redundancy may be in place. For example the CP in a switch, and disks that are arranged in RAID-5 arrays fall into this category as well. Hardware component redundancy is there to reduce the reliance on system redundancy. Some hardware components however, have not been made redundant, because the benefits do not outweigh the costs (as the benefits are limited with system redundancy already in place).

Those non-redundant hardware components are important to monitor, as a failure will result in a (partial) system failure, which will lead to reliance on system redundancy, with the increased possibility of performance problems.

On the storage system the ports are usually not redundant, as in general a LUN is made only available on one port per fabric. A failing port will thus result in at least some LUNs to become available via one fabric only.

The same principle holds for hosts: a single host usually will only have one port connected to each fabric. As hosts ports and storage ports have been identified as probable non-redundant hardware components, everything directly connected has the same status, e.g. the fibre and

switch port. Not all switch ports are not redundant, inter switch trunks generally consist of multiple ISLs, and multiple trunks exist in parallel.

3.2 Capacity shortage

A capacity shortage can be defined as a situation where the demand (e.g. for IO or bandwidth) is higher than the capacity of a segment of the path between host and storage. Capacity shortages, wherever they occur, do not only decrease performance that is experienced by hosts, but can have far worse effects. If, for example, a storage system is too busy, its response times might slow down, which could lead to “buffer credit zero” conditions and frame discards in the Fibre Channel fabric.

A very primitive form of flow control is active in the Fibre Channel fabric that works per link (between two devices) rather than all the way from host to storage system. Frames are being sent and received between so called buffers. A fixed maximum number of buffer credits is negotiated between two devices on a link. Each time a device sends a frame, that frame will take up a buffer credit, until a frame has been acknowledged, which will free up a buffer credit. When an end of a link runs out of buffer credits, it has to wait.

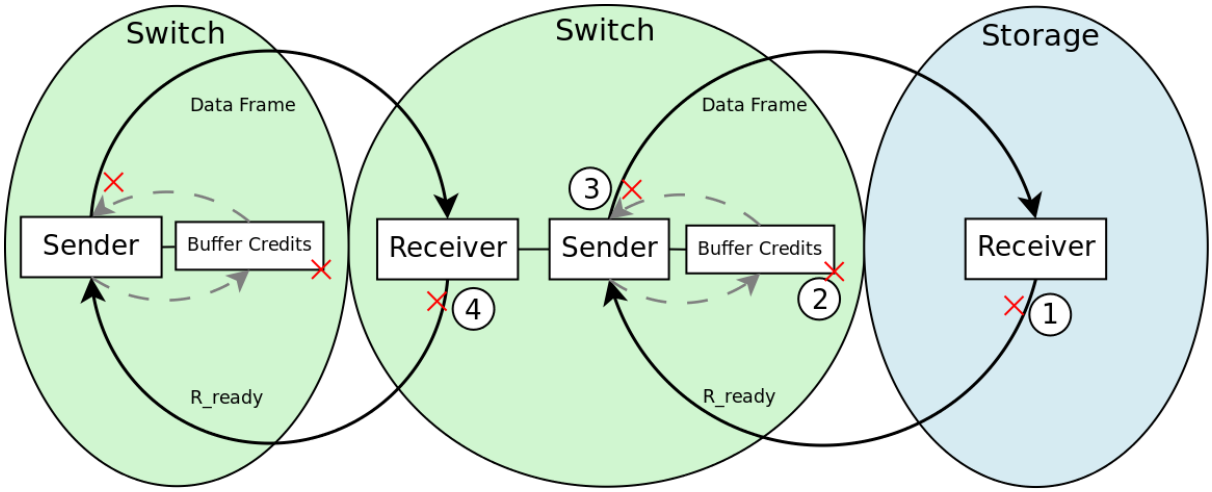
If a switch has to wait too long before the buffer credit zero situation is resolved, it will start discarding frames. The time to live (TTL) on the fabric is 500ms, so a frame will be discarded when it has not been delivered to its destination within 500ms.

A buffer credit zero situation can propagate through a fabric, because the sending end of a link that cannot send frames will cause the receiving end of another link, with frames destined to the blocked link to stop receiving (by not releasing buffer credits to its sender), etc. See also figure 3. And not only will a prolonged buffer credit zero situation propagate along a single path, but when ISLs are involved other hosts and destinations can experience discarded frames as well, as on ISLs multiple host/storage pairs can potentially share buffer credits (for data up to four so called virtual channels that share buffer credits exist per ISL).

A discarded frame will result in an I/O error or time out on a host. Different operating systems and their Fibre Channel/SCSI/multipathing implementations react differently to this situation, but in general this situation should be avoided.

As has been explained, a single slowly responding storage port can cause frame discards that affect several host/storage pairs that do not have capacity shortages themselves. Therefore it is very important to monitor capacity shortages and frame discards.

However, not every frame discard is a result of a buffer credit zero situation, as will be explained in more detail in section 3.4.2.



When a storage system is slow in accepting write data:
1. The storage systems (temporarily) stop sending R_ready messages to the connected switch port.
2. Buffer credits will not be increased anymore.
3. Once run out of buffer credits, the switch cannot send data to the storage system any more.
4. While the switch cannot send data to a storage port, it will stop sending R_ready messages upon receiving data destined for that storage port . This way the buffer credit zero condition can propagate through the fabric over ISLs, potentially affecting data streams from other host and storage systems.

Figure 3 - Buffer credit zero propagation

Several factors play a role in possible capacity shortages. Multiple hosts access LUNs on shared storage clusters, which have a limited capacity in terms of I/Os and MBs per second, as well as a limited number of ports connected to each fabric. If too many hosts access LUNs on the same storage cluster, a bottleneck might occur at the storage side. Furthermore, ISLs are being oversubscribed as the sum of bandwidth demand by devices on any given time will be lower than the total maximum bandwidth that devices have available. For example, a trunk of four ISLs might handle the traffic of thirty hosts, as during normal operation, those hosts never utilise their full capacity at the same time.

As explained in the previous section, (hardware) failures can cause extra stress on components that take over, which will increase the load beyond normal levels. Sound capacity management, however, should ensure that the sum of the load on redundant components will not exceed the capacity of a single component.

3.3 Problem classification

In order to determine the overall health of the storage infrastructure, the problems, as described in section 3.4.1 through 3.4.3 need to be classified. One of the common monitoring systems used within KLM IS assigns a severity level (harmless, warning, minor, critical or fatal) to a single event. For the overall status, however, a per event/problem classification is not enough, as two separate problems should be able to attribute a combined status level.

The SAN health status should apply to a system as a whole, and be valid at a certain moment in time. As already discussed in section 2 (see also figure 1) the storage infrastructure has two redundancy axes: independent fabrics, and independent sites. In section 3.2 has been explained how problems on a fabric can propagate. As problems within a fabric are storage-specific, and problems within a site are most probably of a physical nature, it makes sense to

monitor the SAN health status on a per fabric basis, and combine the two into an overall health status.

As for the time between status updates, an interval needed to be chosen. When a too short interval time is chosen, insignificant peaks might become exaggerated, and a too long time interval will have too much of an averaging effect on peaks that are significant. Furthermore, not all hardware is capable of reporting status information in (near) real time. Information given by people from different groups within KLM IS suggests that some systems report every couple of minutes, other systems have a minimum reporting interval of fifteen minutes, and others report even less often. Those latter systems, however, can most probably be reconfigured to report at least every fifteen minutes, and possible more frequently. As fifteen minutes seems to be the smallest common denominator for all monitored hardware components, it makes sense to fix a status level each fifteen minutes. However, some problems might develop much quicker, and are obvious even before all data is in. Therefore, at each given time a status level can be determined from data and relations that have been processed since the last fixed status level (maximum fifteen minutes old). With this procedure a (preliminary) status level can be reported at (near) real time, and at the end of the fifteen minute period the last (and maximum) status level is fixed. This makes it possible to quickly report fast developing problems, based on info that is at hand, without having to wait for all possible data, which could take up to fifteen minutes.

With regard to impact on the infrastructure, four status levels seem to be desirable:

- No problems detected
- Problem(s) detected which have no immediate impact
- Problem(s) detected with limited impact on the infrastructure
- Problem(s) detected with severe impact on the infrastructure

In this scheme a single problem, or more likely, a combination of problems will determine the SAN health status on a single fabric. In order to combine the status levels of the two fabrics into one, these levels need to be quantified.

A quantification seems appropriate where moving up to a next status level ‘doubles the severity’. If then the two statuses of the fabrics are multiplied, moving up a status level on one fabric will have the same result on the composite. If one fabric is healthy, the combined status will be equal to the status of the other fabric. When 1 is the value of a healthy fabric, the matrix follows from the described model.

Fabric 0 \ Fabric 1	No problems	No impact	Limited impact	Severe impact
No problems	1	2	4	8
No impact	2	4	8	16
Limited impact	4	8	16	32
Severe impact	8	16	32	64

The colours green, yellow, orange and red have been chosen to represent each of the per-fabric status levels. Lighter and darker shades of these colours are used to represent a relative severity level of the overall status.

3.4 Problem detection

In section 3.1 and 3.2 a number of problem situations have been discussed, and explanations have been given as to why they could give valuable contributions to an overall SAN health status monitor. Sections 3.4.1 through 3.4.3 explain how these problem situations can be detected on hosts, switches ad storage clusters respectively.

3.4.1 On the hosts

The three problem indicators that can be found on the hosts: DCB errors, path failures and mirror synchronisation failures. Implementation specific details can be found in appendix I. In this section their relevance and relative impact on the overall SAN health status will be explained.

Although failing HBAs and host ports can likely also be monitored on the hosts, problems with host ports will also be detectable on the switches. As the switches are a far more homogeneous environment than hosts, it is more convenient to monitor host link status on the switches.

Both DCB errors and path failures can provide valuable information about the overall SAN health, as these problems can be related to other SAN components other than those directly connected. DCB errors and path failures both provide information about the LUN and remote storage port involved. See appendix II for the method of extracting such details. This extra information makes it possible to better estimate the cause and or impact of a problem, and it makes it possible to find positive or negative relations between different problems detected on different components. For example, paths through the fabric can be reconstructed; possible identifying other affected systems using the same path, or discarded frames could be matched to DCB errors and path failures.

A single path failure can be caused by hardware failure of a port or excessive frame discards somewhere on the path between host and storage. Both situations can be detected on the fabric, as discussed in section 3.4.2. If a path failure occurs isolated, this might indicate a problem on the host itself, and is thus not likely to have an impact on the fabric the path belonged to. However, if a path over one fabric has failed, there is an increased risk of overloading the other fabric.

DCB errors appear to occur regularly on hosts. Mostly, however, less than 100 occur in total on all hosts, on a single day. A single DCB error could be an isolated problem, but several happening in a short amount of time, especially if they occur on several hosts simultaneously, could indicate a problem on the fabric with limited impact.

For both path failures and DCB errors on hosts (per fabric) the following impact matrix has been defined:

	Related fabric	Other fabric
1 or 2 hosts	No problems	No problems
3 or more hosts	Limited impact	No impact

It should be noted that this situation applies to isolated problems; if related problems are detected somewhere else status levels will increase accordingly. Such interrelations are discussed in section 3.5.

Mirror synchronisation issues, especially when only one host is involved, will not have immediate impact, although an increased risk of capacity shortage arises from the fact that resynchronisation of the mirror volumes may produce a higher than normal load, as the complete volume will have to be read from one storage cluster, and written to another. If multiple hosts have mirror synchronisation problems this risk grows, and the possibility arises that there is a serious problem with one of the storage clusters.

One or two mirror synchronisation problems should not be seen as problems; three or more constitute to problems with no (immediate) impact on both fabrics, as mirror synchronisation issues are fabric independent.

3.4.2 On the switches

The switched fabric forms the heart of the storage infrastructure, connecting all hosts and storage devices. Fibre Channel switches however, have little knowledge of the connected devices, and problems that it detects often do not point directly at a cause. Therefore it is important to relate switch information to other problems, as done in section 3.5.

Frame discards appear for two common reasons: the buffer credit zero condition as described in section 3.2 and topology changes of a path in a fabric, which occur when devices leave the fabric (all frames destined for such a device will be discarded). Frame discards due to buffer credit zero conditions on ISLs are potentially more severe than frame discards on host or storage ports. Therefore a distinction should be made between those two.

Although buffer credit zero conditions can be reported by the switches, they cannot be related to discarded frames, as they are both summaries, and buffer credit zero conditions that do not trigger frame discards have proven to be quite common. The SAN group's experiences with monitoring buffer credit zero conditions in the past has never lead to meaningful information about the fabric's health.

Repeated fabric logins by a single host can cause problems on the fabric. The login process takes CPU cycles from the switch, and when a host is trying to send and/or receive large amounts of data over a flapping interface, frame discards can be the result when the host goes offline during data transfers. A legitimate fabric login occurs whenever a host reboots, or when the HBA is being reset. This behaviour should not occur on a regular basis for a host. Three or more fabric logins within fifteen minutes, can therefore lead to a health status level of problems without impact.

Blade failures and port failures are reported when switch hardware fails. A single interface failure should not have too much impact, apart from increasing stress on the corresponding interface of the other fabric. A blade failure, however, has at least limited, and potentially severe impact on a fabric when several ISLs fail at the same time.

Documentation from Brocade[6,7,8,9] has been consulted to find problem indicators.

3.4.3 On the storage systems

Because different storage systems exist, and researching them individually was beyond the scope of this project, only indicators common to all storage systems have been considered. Performance statistics, measured in Bytes or I/Os per second are available for all storage systems. However, experience from the SAN group has learned that it is very difficult to determine the maximum performance of the different storage systems, and often a single metric does not stand on its own, Therefore it is difficult to find a threshold for performance figures.

An indicator that is common to all storage systems is overall latency[1]. This is the time it takes to complete read and write requests, on the storage system. Even here it is hard to set a hard limit on what constitutes to decreased performance, but a threshold of 20ms seems a good sign that a system is being over stressed, especially when combined with frame discards on the fabric.

3.5 Interrelating the problems

The real power of this project is that it combines data from different components along the path from host to storage, and that it can determine impact of problems better through the relational model as discussed in section 2.2.

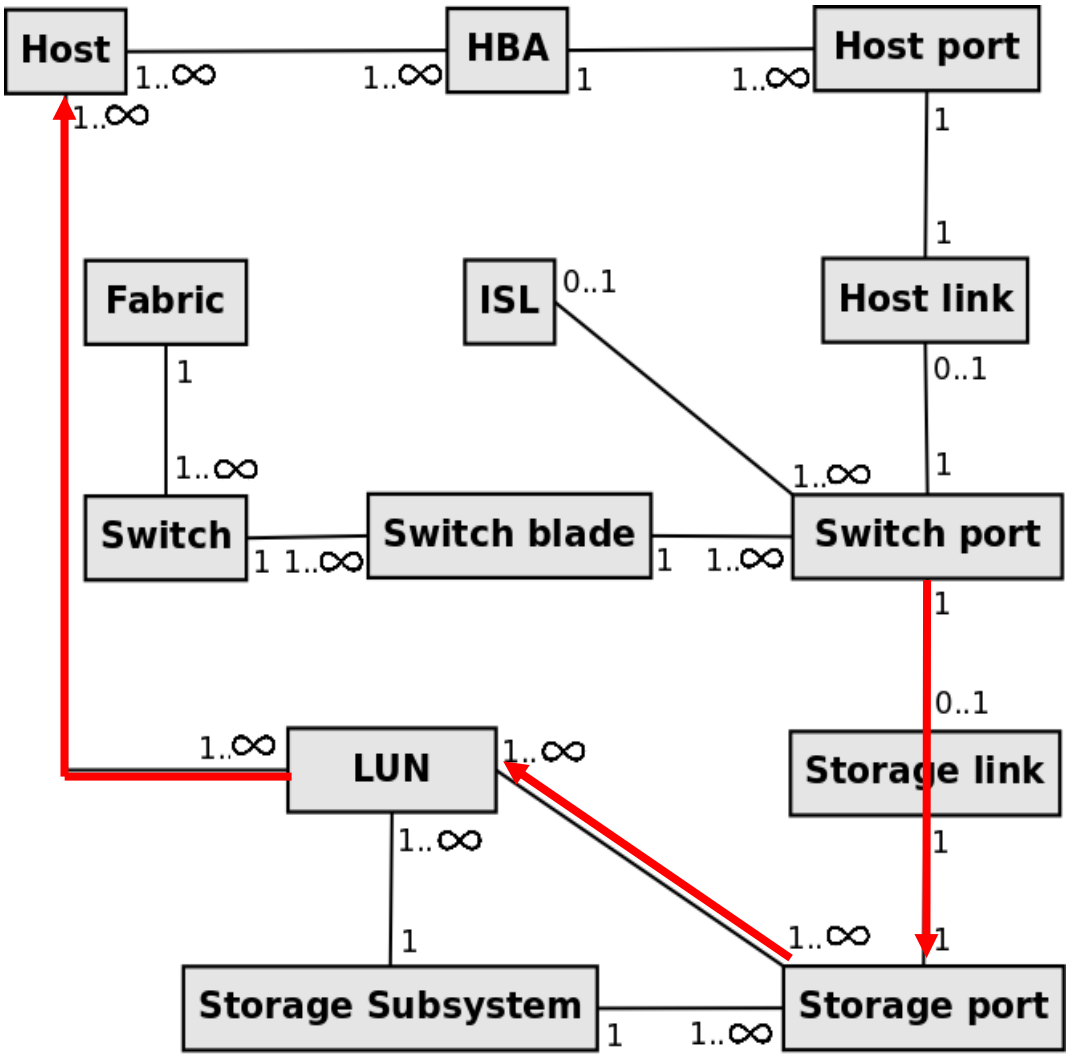
ISLs connect the switches in a fabric, and therefore their health greatly influences the health of the fabric. So problem indicators that can be related to ISLs provide important information. First of all, ISLs themselves can display problem indicators. However, when one is down, over provisioning might prevent any impact. So when looking at ISLs, all ISLs that connect the same two switch ports need to be considered at the same time.

One important metric is the performance of an ISL. This is given as a percentage of the capacity of the ISL. Two sets of ISLs bundles, one per fabric, should never in total exceed 100% of the performance a single ISL bundle could take, or a capacity shortage will surely occur if one fabric fails. This should lead to a problem without impact on at least one of the two fabrics. Furthermore, a threshold of 45 percent can be given to a single set of ISLs, also triggering a problem without impact status, for a single ISL. If at any time a single ISL set is experiencing a load of over 95%, one can be sure impact will be the result, however only limited if no other problem indicators are found.

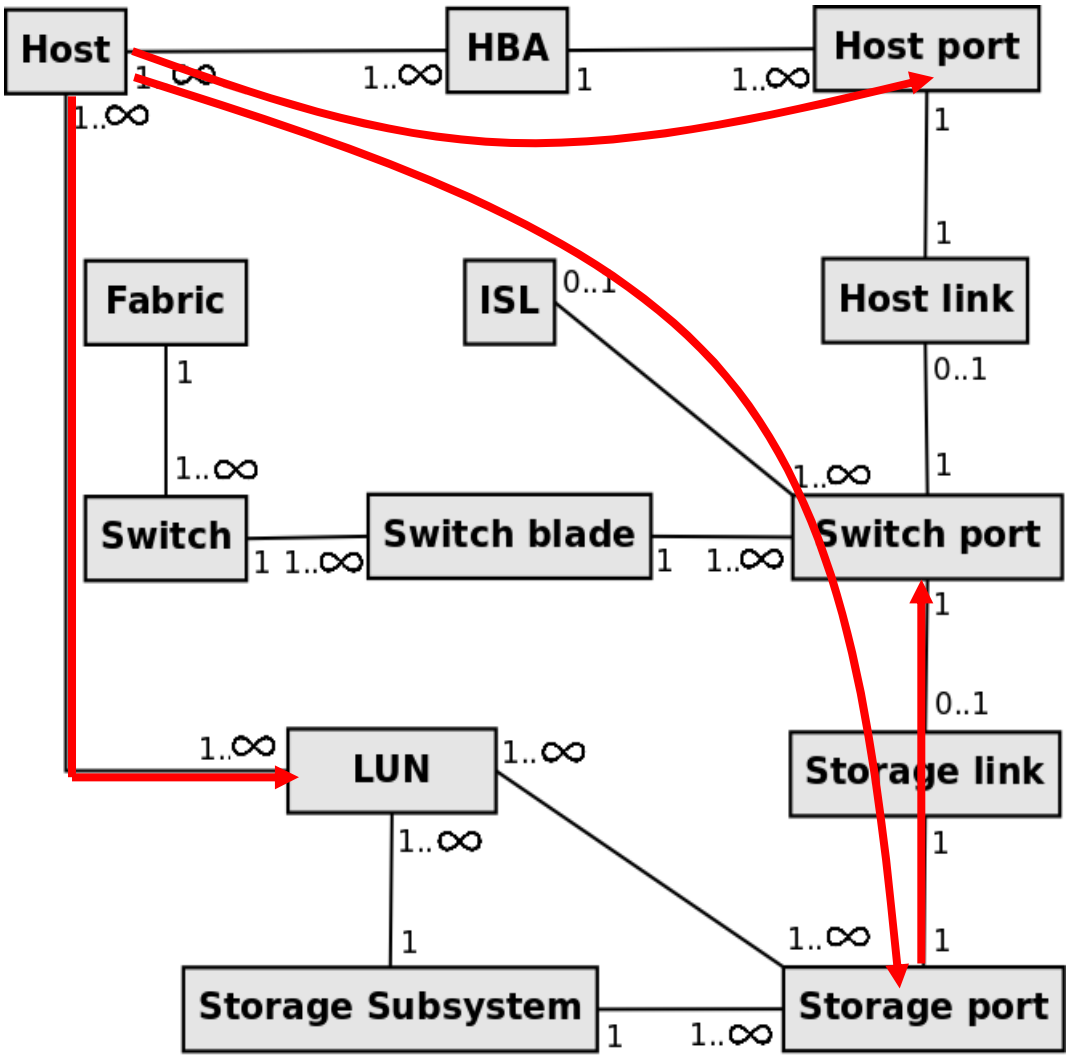
If ISLs are well over subscribed, a port, or even blade failure involving ISL ports of a switch, should only lead to a problem with (limited or severe) impact situation if the performance of the still operational ISLs exceed the predetermined threshold.

The same principle holds for other hardware failures. Only when a problem is confirmed to have impact, by relating the hardware problem condition to other problem conditions which can be related through the model in figure 2 from section 2.2, impact can be confirmed.

Two examples of situations where and how the relational model can be used will be given. One for a failed switch port, and one for a DCB error.



In this example a problem with a switch port (a failure, capacity shortage, or frame discards) can be linked with possibly affected LUNs and hosts. This model can be used to determine the possible impact of a problem, by following the relations, but two problem indicators from distant components can be checked against the model as well, to verify if a relation is possible.



This example shows that a DCB error provides information that bypasses some relations in the model. A DCB error occurs on a host, and refers to a LUN, its own host port, and the WWPN, which is the Fibre Channel address of the storage port. This way a DCB error points to a single switch port to which the storage port is connected. In appendix II more information can be found about how all information can be deduced from a DCB error.

So apart from relating problems along the model, as done in the previous example, it is also possible to bypass some of the many-to-many relations, and see more specific relations.

4 The SAN health status

An important part of the project was to provide a way to present the health status of the storage infrastructure in a continuous way, as well as historically. In section 3.3 the different health status levels have already been introduced. In this section their representation will be discussed.

4.1 In real time

As explained in section 3.3, the health status level is fixed every fifteen minutes because this is the minimum interval at which some of the hardware components report their status. Some problems, however, develop more quickly, and can be determined from problem indicators and relations that are available at shorter intervals. Therefore a preliminary status level can be indicated in near real time. Both the real time status, as well as the fixed status levels of the (recent) past can be combined in a bar chart, where the last bar displays the preliminary status.

The three examples that follow (one for each fabric, and the combined product of the two) show how such a chart could look, using arbitrary status levels just to show some of the possible combinations. When the bars are being displayed narrower, a whole day (96) bars might fit on a single screen. The combination of different heights and colours emphasises the severity, whenever it reaches higher levels. The last bar of each chart displays the current (pending) status, as the final status is always fixed every fifteen minutes. Because the last bar is being updated continuously, this bar acts as a meter.

A feature not displayed is the possibility of clicking on a bar, which should lead to an overview of all problem indicators and their relations, time stamped, which led to the particular status level.

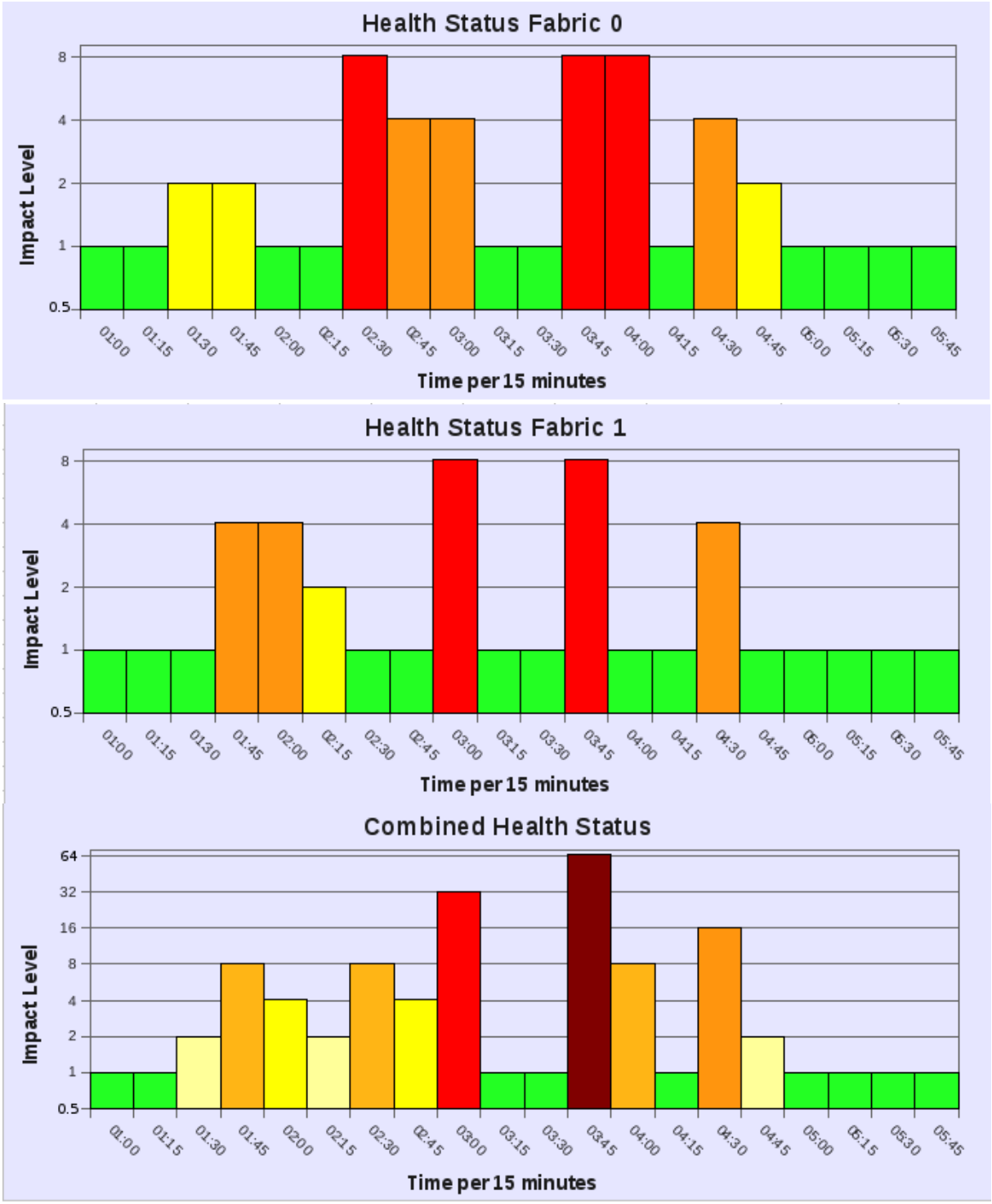


Figure 4 - Continuous health status

4.2 Historically

To display trends over a longer period of time a similar system has been used. Only now a number of bars from the instant view, in this case sixteen, have been averaged. Because intermittent problems might become obfuscated by the averaging process, an extra metric has been added: the average change within a summary.

The average change is indicated by the line above each bar, and is calculated as follows: within a set of statuses that is to be summarised, each time a status level moves up, the amount of change is added to the change total. In the end this number is divided by the total number of intervals.

Apart from the average status and average change the maximum status might also be a relevant fact, as this can show the severity of intermittent or incidental problems. Upon implementation the decision can be made to include it as well.

In the following charts the first bar is based on the first sixteen bars of the charts from the previous section. In this example four hours (sixteen samples) are being averaged. When the bars are being made narrower, allowing 96 to fit on a screen (as suggested in the previous section), sixteen days fit on one screen. Other levels of summarisation (per day, or even week or month) can be made using the same procedure, and as the process is generic, the level of summarisation could even be made real time user configurable.

In any case all data should be kept such that it is always possible to go from a summary to the continuous chart of the requested period, and ultimately to the single events, for analysis of the past. Manual interpretation as well as automated predictive models might be applied on the historical data sets.

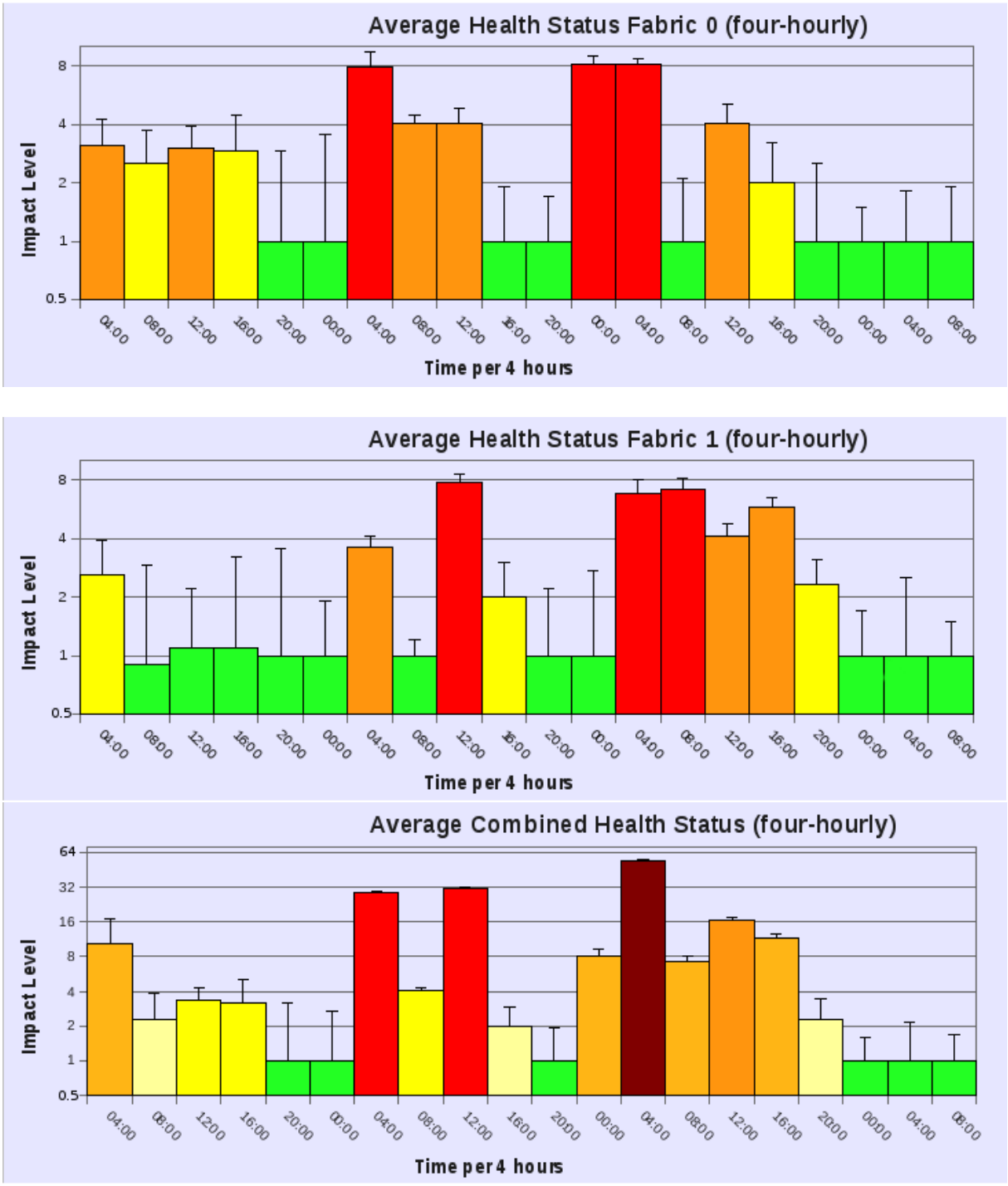


Figure 5 - Historical health status

Conclusions

Different subjects of problem indicators, which are being related to each other in a relational model that has been developed specifically for this purpose, form the basis of a concept SAN health status monitor.

When redundant Fibre Channel fabrics are being deployed, the health status of each of the fabrics can be determined separately, and by multiplying the individual status levels the overall health status can be determined. Health status levels have been defined based on their impact, ranging from no problems, to problems with no immediate impact, via problems with limited, ultimately to problems with severe impact.

Identified problem indicators, possibly amplified by found relations between them, determine the health status. The status of each fabric, as well as the composite, should be kept at (near) real time to be able to respond to fast progressing problem situations quickly. At a fixed time interval the final status of that interval is to be determined, and those statuses can be shown as a graph of the recent past. Trends over longer time periods can be shown by aggregating the determined status levels over a certain time frame, keeping track of the average status, average change, and possibly the maximum status level.

One of important factors for the success of such a system is the broad scope of the problem indicators, which originate not only from SAN and storage equipment managed by SAN people, but also from hosts that are the responsibility of different groups within the organisation.

Future work

This project has led to a concept SAN health status monitor, which has yet to be implemented. From this report a proof of concept may be created. However, not all problem indicators and relations have been fully quantified yet. Thus, further development of a translation scheme between problem indicators and relations needs to be developed.

Even though the entities and relations in the component relation model have been carefully defined, it is still based on theory, and in reality some modifications will have to be made.

Furthermore, more components (e.g. Windows hosts) and more problem indicators might be defined in the future, to enhance the health status monitor, and make it even more accurate.

After this concept SAN health status monitor has been implemented, historical data will be collected. This data can be used to improve the translation scheme between problem indicators, relations and health status levels, and thresholds can be tuned. Furthermore, a predictive model can be developed.

Something on a more organisational rather than technical level, the research question about responsibilities and actions that follow from different health statuses, will have to be defined. Unfortunately the four weeks that were reserved for this research project have proven to be insufficient to define reasonable policies.

Bibliography

[1] Monitoring Your Storage Subsystems Using TotalStorage Productivity Center; Mary Lovelace, Arthur Letts, Massimo Mastrorilli, Markus Standau, Terry X Zhou; IBM 2007

[2] Red Hat Enterprise Linux 4.5.0; System Administration Guide; Red Hat 2007

[3] DM Multipath DM Multipath Configuration and Administration Edition 1.0; Red Hat 2009

[4] Multipath Subsystem Device Driver User's Guide; IBM 2007\

[5] AIX Logical Volume Manager from A to Z: Troubleshooting and Commands; Laurent Vanel, Ronald van der Knaap, Dugald Foreman, Keigo Matsubara, Antony Steel; IBM 2000

[6] Fabric OS Message Reference; Supporting Fabric OS v5.3.0; Brocade 2007

[7] Fabric OS MIB Reference; Supporting Fabric OS v5.3.0; Brocade 2007

[8] Fabric OS Message Reference; Supporting Fabric OS v6.1.0; Brocade 2008

[9] Fabric OS MIB Reference; Supporting Fabric OS v6.1.0; Brocade 2008

Appendices

Appendix I

Although the concept of storage management between UNIX and Linux as used within KLM IS is similar, the implementations differ.

One general concept to which all hosts adhere is host based mirroring. This means that the host is responsible for keeping its mirror volumes in sync. Writes will always have to be committed to both mirrored volumes; reads can be done from either mirror. A typical failure condition is an out of sync mirror. Not only does this affect the host itself, resynchronisation will cause stress on the storage systems as complete volumes will have to be read from one storage cluster, be transported over the fabric via the host, and written to the other cluster. Both Linux and UNIX systems can report mirror synchronisation failures.

Linux uses RedHat's md[2] application to create a software RAID 1 system across two LUNs located on the different storage clusters. The ReHat device mapper multipath driver[3] is used for load balancing and fail-over behaviour between the different HBAs, and their associated paths to each LUN. Multipath load balancing is done in a round robin fashion, and md routes most read operations (about 80%) to the storage cluster that is physically located closest to the host (as latency is lower). Unfortunately no detailed information on logging capabilities of device mapper multipath has been found in the documentation[3]. The only useful status information, currently already being monitored by the Linux group, is the occurrence of path failures.

The UNIX systems use IBM's sdd driver[4] to handle mutipathing, and LVM[5] to handle file system mirroring. In addition to information about failed paths and failed mirrors (as with Linux) the UNIX systems provide another, more detailed status message: the DCB error. Such an error occurs whenever an IO operation fails, and provides information not only on which HBA the error occurred on, but also the LUN that was involved.

Appendix II

To argument that logical and physical things for example hosts and LUNs can be presented in one data structure, the Management Information Base (MIB, RFC1066) can deliver proof of such practice. The MIB is a hierarchical tree of physical parts that have a relation with logical parts. An example of such practice can be found in the definition of MIB-II (RFC1213), that is used to monitor many different types of TCP/IP based devices and its physical interfaces. The interface group of MIB-II has various logical object types that contain a value, for example ifInDiscards, ifInErrors, ifOutDiscards, ifOutErrors.

In the view here below you see logical object types that are part of the interface group of MIB-II. An interface is a physical part and the object types that are bold, that are logical parts that contains in this case information about the interface.

```
          IfEntry ::=
              SEQUENCE {
                  ifIndex
                      INTEGER,
                  ifDescr
                      DisplayString,
                  ifType
                      INTEGER,
                  ifMtu
                      INTEGER,
                  ifSpeed
                      Gauge,
                  ifPhysAddress
                      PhysAddress,
                  ifAdminStatus
                      INTEGER,
                  ifOperStatus
                      INTEGER,
                  ifLastChange
                      TimeTicks,
                  ifInOctets
                      Counter,
                  ifInUcastPkts
                      Counter,
                  ifInNUcastPkts
                      Counter,
                  ifInDiscards
                      Counter,
                  ifInErrors
                      Counter,
                  ifInUnknownProtos
                      Counter,
                  ifOutOctets
                      Counter,
                  ifOutUcastPkts
                      Counter,
                  ifOutNUcastPkts
                      Counter,
                  ifOutDiscards
                      Counter,
                  ifOutErrors
                      Counter,
```

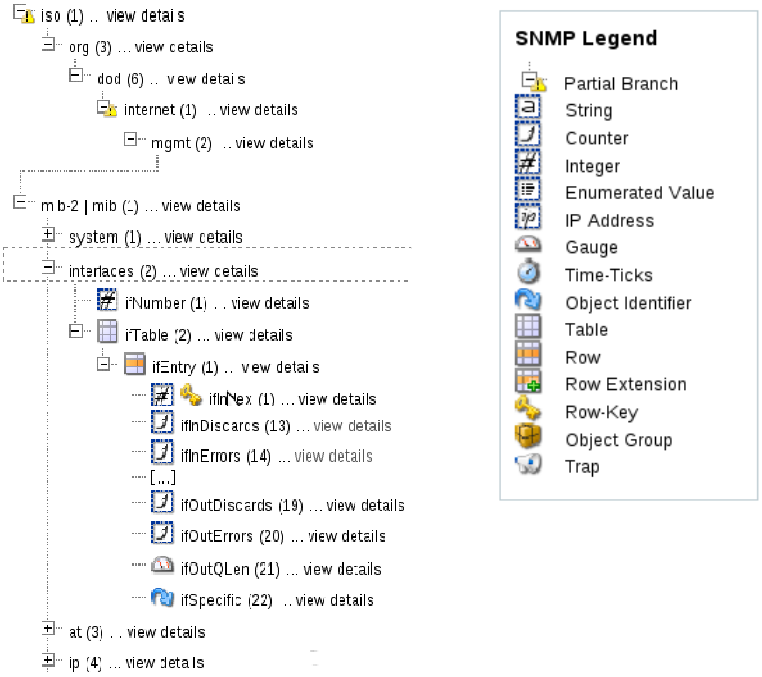
```
[…]
```

The object types contain information about the interfaces, that is logical and can change every second, because data is travelling through the port. That this object types are logical, that is supported by the RFC 1213. In the view below you can see that the object types are counters that can increase when inboud packets will be discarded. This counter is a logical value that only increases when inbound packets are discarded.

```
 ifInDiscards OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
                      "The number of inbound packets which were chosen
                      to be discarded even though no errors had been
                      detected to prevent their being deliverable to a
                      higher-layer protocol.  One possible reason for
                      discarding such a packet could be to free up
                      buffer space."
              ::= { ifEntry 13 }

          ifInErrors OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
                      "The number of inbound packets that contained
                      errors preventing them from being deliverable to a
                      higher-layer protocol."
              ::= { ifEntry 14 }

 ifOutDiscards OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
                      "The number of outbound packets which were chosen
                      to be discarded even though no errors had been
                      detected to prevent their being transmitted.  One
                      possible reason for discarding such a packet could
                      be to free up buffer space."
              ::= { ifEntry 19 }

          ifOutErrors OBJECT-TYPE
              SYNTAX  Counter
              ACCESS  read-only
              STATUS  mandatory
              DESCRIPTION
                      "The number of outbound packets that could not be
                      transmitted because of errors."
              ::= { ifEntry 20 }
```

Source:
http://tools.ietf.org/html/rfc1213#section-6 (Page 16 till 23)

In figure the following figure you see a graphical hierarchical view of a MIB.



Source: http://support.ipmonitor.com/mibs_byoidtree.aspx?oid=1.3.6.1.2.1.2#h

Appendix III

The UNIX department collects all interesting errors on their systems, for which they use scripts and tools. In the part here below you see an example of a DCB error. The DCB errors are reported on the hosts. This example contains information on what kind of DCB error it was (DCB47997), the first four digits are the sequence number (0618) and the other part is the date (220509). And the error gives a disk operation error at "hdisk7".

```
# errpt |grep DCB |head -1
DCB47997 0618220509 T H hdisk7 DISK OPERATION ERROR -> ERROR in errpt
```

With the knowledge that there was an error on "hdisk7" a relation can be made to the storage and the following command output gives information about the storage. The "vpath1" is the path to the storage. The number "510428230" is a serial numbers that tells with disk are available, namely "hdisk6" and "hdisk7". The first four digits are the LUNid (5104) and the last part tells the storage device (28230) where the disk are part of. The five is standard not view in the overview.

```
# lsvpcfg |grep hdisk7
vpath1 (Avail pv lvg00025) 10428230 = hdisk6 (Avail ) hdisk7 (Avail )
```

The number "1048230" is called also a serial and with this number you can see the path to the disk (fscsi0/hdisk7) . Vpath1 is a configuration that is one path to the disk.

```
# datapath query device |grep -p 10428230
DEV#: 0 DEVICE NAME: vpath1 TYPE: 2105800 POLICY: Optimized
SERIAL: 10428230
==========================================================================
Path# Adapter/Hard Disk State Mode   Select    Errors
0     fscsi1/hdisk6     OPEN  NORMAL 19415412       0  ->
1     fscsi0/hdisk7     OPEN  NORMAL 19154233       0
```

It is possible to get information on different ways. In the part here below you see the serial number again, what tells what the LUNid (5104) is and the on what kind of storage the disk is connected (28230). The ABDE tells also that it's an ESS 28230 storage device.

```
# lscfg -vl hdisk7
hdisk7 U0.4-P1-I5/Q1-W5005076300C4ABDE-L5104000000000000 IBM FC 2105
   Manufacturer................IBM
   Machine Type and Model......2105800
   Serial Number...............10428230
[…]
```

The information here above is usefull, because with this information you can het other infromation about the system see below. For example you see the location of the LUN, the machine type / model and serial number.

```
# errpt -a |pg

LABEL:          SC_DISK_ERR4
IDENTIFIER:     DCB47997

Date/Time:       Thu Jun 18 22:05:38 DFT
Sequence Number: 95826
Machine Id:      00518D3A4C00
Node Id:         kl10065e
Class:           H
Type:            TEMP
Resource Name:   hdisk7
Resource Class:  disk
Resource Type:   2105
Location:        U0.4-P1-I5/Q1-W5005076300C4ABDE-L5104000000000000
VPD:
    Manufacturer................IBM
    Machine Type and Model......2105800
    Serial Number...............10428230
[…]
```

By hand of the serial number you found the adapters where the hosts connects to, to communicated with the storage. The search on the adapter you find the World Wide Port Name (WWPN), which communicates with the storage ports, which has a relation with the switch port and de other switch port with the host. So it's possible for the Unix department to make a relation between information on onside of the SAN network to the other.

```
# lscfg -vl fcs0 (voor fscsi0 protocol)
fcs0        U0.4-P1-I5/Q1 FC Adapter


    Part Number….............80P4383
    EC Level….................A
    Serial Number…............1A62502078
    Manufacturer….............001A
    Feature Code/Marketing ID…2765
    FRU Number….............80P4384
    Network Address…..........10000000C95815AA
[…]
```