Towards a Framework for Failure Impact Analysis and Recovery with Respect to Service Level Agreements

Andreas Hanemann, David Schmitz
Munich Network Management Team
Leibniz Supercomputing Center
Barer Str. 21, D-80333 Munich, Germany
{*hanemann, schmitz*}*@lrz.de*

Martin Sailer
Munich Network Management Team
University of Munich
Oettingenstr. 67, D-80538 Munich, Germany
sailer@informatik.uni-muenchen.de

Keywords

Impact analysis, service management, service level agreements, fault management

1. Introduction

In today's IT environments provisioning of services with guaranteed QoS has become a crucial point for IT service providers. Thus, it is an important problem for a provider to ensure that the agreements with their customers are met. As the Mean Time between Failures (MTBF) and the Mean Time to Repair (MTTR) are often part of the Service Level Agreements (SLAs), it has become vital for a provider to react quickly and accurately when detecting a failure in a resource.

To be able to react in this way, a holistic view of the inner structure of service provisioning is required. That is, to know the dependencies of the offered services on subservices and resources as well as the customers' SLAs, their QoS parameters, and the current service usage. Appropriate recovery actions can then be chosen by applying this knowledge.

Therefore, we propose a modeling framework that formalizes the mentioned relationships and allows to automate the decision procedure. This framework shall be able to address the following issues:

Short-term effect: There are currently failures in one or more resources. Which services and SLAs are affected by the failures? How important are the failures especially with respect to possible SLA violations? As a consequence, how can recovery alternatives be identified and which one should be chosen to deal with the current situation?

Mid-term considerations: The IT provider could forecast what would happen in case of resource failures. This is useful to identify critical resources and therefore to optimize the way services are provisioned.

Long-term considerations: The service offers could be reviewed whether they are appropriate for the resources which are currently used for their provisioning. The service pricing could be adapted in a way that e.g. peak utilization can be avoided.

After having introduced these general requirements, we now proceed with a detailed analysis which has been conducted on the basis of a service provisioning scenario.

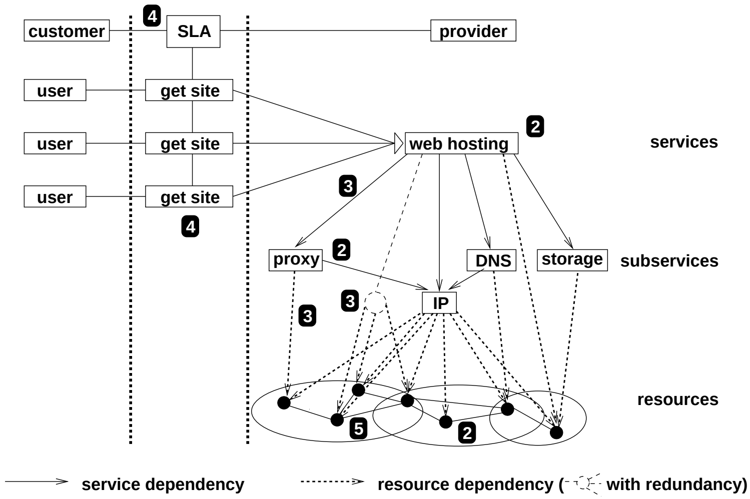

Figure 1: Web hosting scenario

2. Requirement Analysis and Component Identification

The Web Hosting Service offered by the Leibniz Supercomputing Center (LRZ) serves as an example IT service. The LRZ, which is the computing center for the Munich universities and runs the scientific network in Munich, hosts web sites for other research institutions. In this Web Hosting Scenario the impact of resource failures for different kinds of QoS parameters shall be examined.

Besides more general QoS parameters such as availability and delay, there also exist parameters specific to this service. An example is the up-to-dateness of the content. It could e.g., be the case that a customer has transferred new content for the hosted web site to the LRZ, but this content is not displayed to the service users yet (e.g., because of server-side web caching). The SLA could contain a maximum duration until a content change comes into effect for the customers. On the other hand, a maximum number of content changes with a maximum size could be defined.

The Web Hosting Scenario is shown in Figure 1. The service is offered by the LRZ (provider) to its customers at the customer provider interface. A customer can allow several users to use the service by creating a password protected part of the hosted site or can allow everybody to see the content. The quality and cost issues of the service are laid down in SLAs. On the provider side, the Web Hosting Service is provided using subservices. In the scenario these subservices are DNS, proxy service, connectivity service (IP), and storage service. Both services and subservices depend on resources which they are provisioned upon. These are e.g., network components, network links, an end system's main memory or processes running on a server. As depicted in Figure 1 a service can depend on more than one resource and a resource can be used by one or more services.

As the LRZ would like to ensure that the agreed SLAs are met, it is necessary to react accurately to faults occurring in one or more resources. To achieve this, an impact analysis has to be performed where different kinds of dependencies are used to identify affected services and corresponding SLAs. Furthermore, a methodology has to be defined to decide which recovery steps should be carried out in order to deal with faults.

In detail, the following issues need to be addressed which were derived from the scenario. The numbers refer to the locations in the Figure.

1. **Workflow requirements:** A workflow has to be defined to identify steps needed during failure impact analysis and recovery. The information retrieved during the workflow steps has be stored in a way that the service provisioning can be documented with respect to the customers. An interface to the customers has to be capable of exchanging information about the current SLA reports and failure/repair notifications.
2. **Modeling of services and resources:** A service model is needed covering the service features relevant for the impact analysis. It is especially necessary to model the QoS parameters. In [2] we defined detailed requirements for a QoS definition. A key requirement is, that parameters are specified independent from a provider's service implementation. The provider may have subscribed services from other providers which drives the need for an implementation independent definition in order to reliably provide own services. In addition, the customers demand provider-independent QoS definitions to easily compare offers of different providers.
   It also has to be investigated which resource modeling could be used. It should e.g. be possible to distinguish between different failure states, because situations like a fiber working, but only at high error rates, need also to be considered. Another case could be a component which only works sporadically (either deterministic or indeterministic).
3. **Dependency modeling:** In the scenario, different kinds of dependencies can be distinguished. There are dependencies between different services, dependencies between services and resources, and dependencies on the resource level. It is important to identify the characteristics of these dependencies and their necessary attributes.
   For the modeling of dependencies between services and resources, redundancies in the service provisioning have to be covered by the modeling. For instance, the Web Hosting Service is provided using several redundant servers, therefore failures of certain servers do not result in an unavailable service. There are different ways to define that the service is working. It could be regarded as working properly if at least one of the servers is available or a certain percentage of them is working.
   The modeling of dependencies on the service level needs to reflect interchangeability of subservices. Accordingly, some dependencies may be regarded as strong if there are no alternate services and weak if it is possible to easily use another service.
   In general, the trade-off between the effort to be invested for a detailed dependency modeling and the resulting benefit [1] needs to be taken into account. Consequently, the dependency modeling should be designed in a way that it can be applied for different modeling granularity.
   Concerning the processing of the dependency information, efficient data structures and search algorithms need to be in place.
4. **SLA:** As the impact analysis is performed with respect to SLAs, an SLA modeling is needed based on the QoS modeling mentioned above. In case a subservice has been outsourced to another provider, the consequences of a failure in this subservice also

have to be considered. From a business point of view, it is necessary to ensure that a provider's SLA with a subprovider contains appropriate penalties. For instance, if the provider cannot meet the SLAs with its customers due to a failing subservice, these penalties have to cover the resulting costs. Accordingly, the SLA definition should allow for a derivation of such a mapping.

A monitoring infrastructure is needed to measure the QoS as defined in the SLAs. Together with a history of past QoS violations the current status of the SLAs can then be determined.

The current service usage should be taken into account for performing the impact analysis. If e.g., a failure in a resource leads to a malfunction of a service, but the service is currently not used, there is no impact on the SLAs at the moment.

5. **Recovery measures:** After the impact of a resource failure has been determined, recovery measures have to be performed. The ways a provider can react to a certain situation have to be modeled depending on the kind of resource failure. If there are multiple errors at the same time, then it has to be decided furthermore how the recovery resources (e.g., staff, test equipment) are applied to handle them.

From these requirements some components can be derived which have to be part of the framework. At least one repository is needed which contains the information about services and resources (requirement 2) as well as their dependencies (requirement 3). In addition, an entity is necessary which is able to efficiently search inside these dependencies (requirement 3). Besides an SLA repository, a monitoring component is required to check the provided QoS and to determine the effect onto the SLAs (requirement 4). To determine the actual consequences more precisely a service usage monitoring entity should also be in place. For the recovery measures a repository containing possible recovery actions is needed as well as a decision component (requirement 5).

3. Conclusions and Future Work

In this extended abstract requirements for a framework for impact analysis where derived by using a real-world scenario. In addition, components which have to be part of this framework have been identified.

In the future a detailed component design for entities not covered by the state-of-the-art is needed. Open issues here are especially the service dependency modeling, service usage monitoring, and recovery actions modeling. Results for a proof-of-concept are also required that demonstrate and quantify the benefits of the approach.

References

[1] M. Gupta, A. Neogi, M. Agarwal, and G. Kar. Discovering dynamic dependencies in enterprise environments for problem determination. In *Proceedings of the 14th IFIP/IEEE Workshop on Distributed Sytems: Operations and Management*. IFIP/IEEE, October 2003.

[2] A. Hanemann, M. Sailer, and D. Schmitz. Variety of QoS - the MNM service model applied to web hosting services. In *Proceedings of the 11th International Workshop of the HP OpenView University Association (HPOVUA04)*. HPOVUA, June 2004.