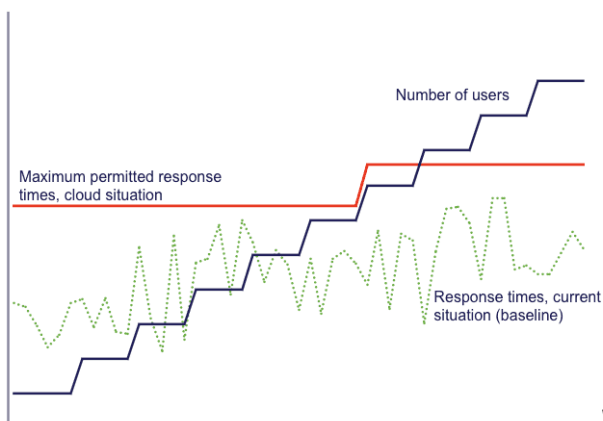


This section is about applying known performance test techniques in the cloud context. In addition, we'll describe an approach to new performance test techniques, such as testing elasticity. However, not all traditional performance tests are run. For example, executing a stress test is in most cases not possible because it jeopardizes the stability of the service for the other customers. User experience of response times and the system load are key in testing performance. Depending on the identified risks, this leads to a limited or thorough review of different circumstances in which certain performance is desired (during the day, working hours, affects of other users, etc.). Also to be considered is the performance in the case of an (unexpected) increase in user numbers, peaks in load, and prolonged periods of high load. To determine whether a service meets the performance requirements and expectations, acceptance criteria are needed. What are requirements regarding response times and at which loads do these requirements have to be met? When "going into the cloud," one has to realize that with services it can be difficult to achieve the same response times one is accustomed to. Introducing more communication steps and the fact that services are not specifically made for the customer are probable factors in slower response times. If it is needed to determine the response times as a result of going into the cloud, the performance on the current infrastructure has first to be measured. The resulting outcome is used as a reference (baseline) with which to compare the cloud performance (see figure below).



With SaaS the center of gravity for performance lies fully with the service provider: they are responsible for the performance that is experienced by the end users (not including equipment and Internet connection speed). With PaaS and IaaS this is more complex because the application and software environments are partially or

fully the responsibility of the customer. The performance as experienced by the end user is dependent on the efficiency of the application software (customer), the speed/efficiency of the software platform (customer and/or supplier), and the speed of the infrastructure (supplier). Most performance tests are applicable to all service models. If necessary, the effect the service model has on the preparation and execution of performance tests should be discussed. The following subsections provide overviews of the different kinds of performance tests and include practical examples for applying these tests to services. In addition, the various aspects of testing performance, such as creating performance test cases, the use of tools, and test setups, will be addressed.

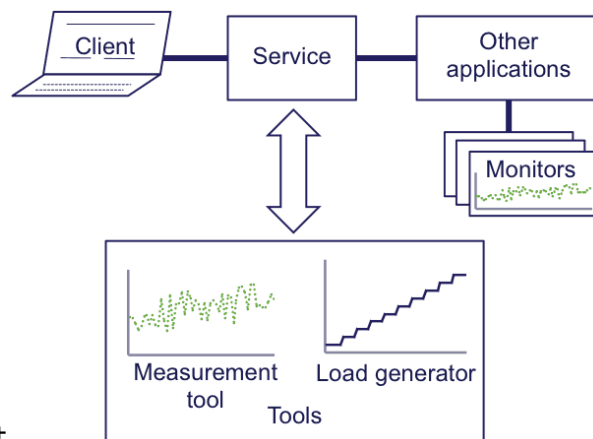
Checklist with test ideas for 'testing performance'

- 5.2.1 Load test
 - Average load (such as the number of users)
 - Peak load
 - Many users logging on simultaneously
 - Many highly active users
 - Load during working hours (or other relevant times)
 - Steadily increasing load (user organization is growing)
 - Steadily increasing number of customers for the service
 - Increased load on the Internet
 - Suddenly increasing load
 - Heavy usage actions (uploading or downloading large files)
 - Frequently occurring actions
 - Business critical actions
- 5.2.2 Stress test
- 5.2.3 Endurance test or volume test
- 5.2.4 Testing elasticity and manual scalability
 - (Automatic) scaling up does not perform as required.
 - (Automatic) scaling down does not perform as required.
 - During scaling functional problems emerge.
 - Information about use-based costs is not adequate.
- 5.2.5 Setting up test cases with operational profiles (Software Reliability Engineering by

Initiators	Operations	Frequency (number/week)	Likelihood
Administrator time sheets application	Arrange screen flow	0.1	0.0005
	Mark obligatory fields	0.1	0.0005
Project administrator	Add project	0.4	0.002
	Close project	0.4	0.002
	Add project employee	3	0.015
		8	0.04
	Connect project employee to project	20	0.1
Project employee	Fill in time sheet each week	80	0.4
	Release time sheet each week	80	0.4
Manager	Approve time sheets	8	0.04
	Total	200	1

John D. Musa (McGraw-Hill, 1999)

- Step 1: Identify the initiators of operations
- Step 2: List operations
- Step 3: Review listed operations
- Step 4: Determine the frequency of operations
- Step 5: Determine the likelihood of each operation occurring
- 5.2.6 Test cases aimed at specific bottlenecks
 - Insufficient bandwidth in parts of the infrastructure
 - Insufficient throughput
 - High latency
- 5.2.7 Including cloud aspects in test cases
 - The world (localization aspects)
 - Customer's resources (like internet access bandwidth)
- 5.2.9 Test cases for endurance/volume test
- 5.2.10 Test cases for elasticity
 - Elasticity (automatic scalability)
 - Manual scalability



- 5.2.11 Test setup for a performance test
 - Measurement tool
 - Load generator
 - Monitors
- 5.2.12 Representative test environment

[Terug naar Testing cloud services](#) | [Terug naar Test Measures](#)