

## Scalable Performance Testing: Facilitating The Assessment Of Application Performance Under Substantial Loads And Mitigating The Risk Of System Failures

**Authors :** Solanki Ravirajsinh

**Abstract :** In the software testing life cycle, failing to conduct thorough performance testing can result in significant losses for an organization due to application crashes and improper behavior under high user loads in production. Simulating large volumes of requests, such as 5 million within 5-10 minutes, is challenging without a scalable performance testing framework. Leveraging cloud services to implement a performance testing framework makes it feasible to handle 5-10 million requests in just 5-10 minutes, helping organizations ensure their applications perform reliably under peak conditions. Implementing a scalable performance testing framework using cloud services and tools like JMeter, EC2 instances (Virtual machine), cloud logs (Monitor errors and logs), EFS (File storage system), and security groups offers several key benefits for organizations. Creating performance test framework using this approach helps optimize resource utilization, effective benchmarking, increased reliability, cost savings by resolving performance issues before the application is released. In performance testing, a master-slave framework facilitates distributed testing across multiple EC2 instances to emulate many concurrent users and efficiently handle high loads. The master node orchestrates the test execution by coordinating with multiple slave nodes to distribute the workload. Slave nodes execute the test scripts provided by the master node, with each node handling a portion of the overall user load and generating requests to the target application or service. By leveraging JMeter's master-slave framework in conjunction with cloud services like EC2 instances, EFS, CloudWatch logs, security groups, and command-line tools, organizations can achieve superior scalability and flexibility in their performance testing efforts. In this master-slave framework, JMeter must be installed on both the master and each slave EC2 instance. The master EC2 instance functions as the "brain," while the slave instances operate as the "body parts." The master directs each slave to execute a specified number of requests. Upon completion of the execution, the slave instances transmit their results back to the master. The master then consolidates these results into a comprehensive report detailing metrics such as the number of requests sent, encountered errors, network latency, response times, server capacity, throughput, and bandwidth. Leveraging cloud services, the framework benefits from automatic scaling based on the volume of requests. Notably, integrating cloud services allows organizations to handle more than 5-10 million requests within 5 minutes, depending on the server capacity of the hosted website or application.

**Keywords :** identify crashes of application under heavy load, JMeter with cloud Services, Scalable performance testing, JMeter master and slave using cloud Services

**Conference Title :** ICSTVV 2025 : International Conference on Software Testing, Verification and Validation

**Conference Location :** San Francisco, United States

**Conference Dates :** September 27-28, 2025