# Outline

- Cloud application workload characteristics
- Performance metrics for cloud applications
- Cloud application testing
- Performance testing tools
- Load test and bottleneck detection case study

# Benchmarking

Why?

Provisioning and capacity planning

- Determining  computing, memory and network resources to provision for the application.
- Comparing alternative deployment architectures and choosing the  best.
- Ensure proper utilization of resources
- Determining the utilization of computing, memory and network  resources for applications and identify resources which are either under-utilized or over-utilized

Market readiness of applications

- Performance of an application depends on the characteristics of the workloads it experiences.  Different types of workloads can dead to different performance for the same application.
- To ensure the market readiness of an application it is important to model all types of  workloads the application can experience and benchmark the application with such workloads.

# Cloud Application Benchmarking - Steps

## Trace Collection/Generation

Application is instrumented to log information  such as the requests submitted by the users, the time-stamps of the requests, etc.

## Workload Modeling

Creation of mathematical models that can be used for generation of  synthetic workloads.

## Workload Specification

Workload Specification Language (WSL) is often used for specification of  application workloads. WSL can provide a structured way for specifying the workload attributes  that are critical to the performance of the applications. WSL can be used by synthetic workload  generators for generating workloads with slightly varying the characteristics.

## Synthetic Workload Generation

Generates workloads which is representative of the  real workloads.

# Synthetic Workload Generation Approaches

## Empirical approach
- Traces of applications are sampled and replayed to generate the synthetic workloads.
- Lacks flexibility as the real traces obtained from a particular system are used for workload generation which may not well represent the workloads on other systems with different configurations and load conditions.

## Analytical approach
- Uses mathematical models to define the workload characteristics that are used by a synthetic workload generator.
- Analytical approach is flexible and allows generation of workloads with different characteristics by varying the workload model attributes.
- It is possible to modify the workload model parameters one at a time and investigate the effect on application performance to measure the application sensitivity to different parameters.

# User Emulation vs Aggregate Workloads

The commonly used techniques for workload generation are:

User Emulation

- Each user is emulated by a separate thread that mimics the actions of a user.
- The attributes for workload generation in the user emulation method include think time, request types, inter- request dependencies, for instance.
- Allows fine grained control over modeling the behavioral aspects of the users interacting with the system under test, however, it does not allow controlling the exact time instants at which the requests arrive the system.

Aggregate Workload Generation:

- Specifies the exact time instants at which the requests should arrive the system under test.
- However, there is no notion of an individual user in aggregate workload generation, therefore, it is not possible to use this approach when dependencies between requests need to be satisfied.
- Dependencies can be of two types inter-request and data dependencies.
- An inter-request dependency exists when the current request depends on the previous request, whereas a data dependency exists when the current requests requires input data which is obtained from the response of the previous request.

# Workload Characteristics

- Session
  - A set of successive requests submitted by a user constitute a session.

- Inter-Session Interval
  - Inter-session interval is the time interval between successive sessions.

- Think Time
  - In a session, a user submits a series of requests in succession. The time interval between two successive requests is called think time.

- Session Length
  - The number of requests submitted by a user in a session is called the session length.

- Workload Mix
  - Workload mix defines the transitions between different pages of an application and the proportion in which the pages are visited.

# Application Performance Metrics

The most commonly used performance metrics for cloud applications are:

- Response Time
  - Response time is the time interval between the moment when the user submits a request to the  application and the moment when the user receives a response.

- Throughput
  - Throughput is the number of requests that can be serviced per second.

# Considerations for Benchmarking Methodology

- Accuracy
  - Accuracy of a benchmarking methodology is determined by how closely the generated synthetic workloads mimic the realistic workloads.

- Ease of Use
  - A good benchmarking methodology should be user friendly and should involve minimal hand coding effort for writing scripts for workload generation that take into account the dependencies between requests, workload attributes, for instance.

- Flexibility
  - A good benchmarking methodology should allow fine grained control over the workload attributes such as think time, inter-session interval, session length, workload mix, for instance, to perform sensitivity analysis.
  - Sensitivity analysis is performed by varying one workload characteristic at a time while keeping the others constant.

- Wide Application Coverage
  - A good benchmarking methodology is one that works for a wide range of applications and not tied to the application architecture or workload types.

# Types of Tests

- Baseline Tests
  - Baseline tests are done to collect the performance metrics data of the entire application or a component of the application.
  - The performance metrics data collected from baseline tests is used to compare various performance tuning changes which are subsequently made to the application or a component.
- Load Tests
  - Load tests evaluate the performance of the system with multiple users and workload levels that are encountered in the production phase.
  - The number of users and workload mix are usually specified in the load test configuration.
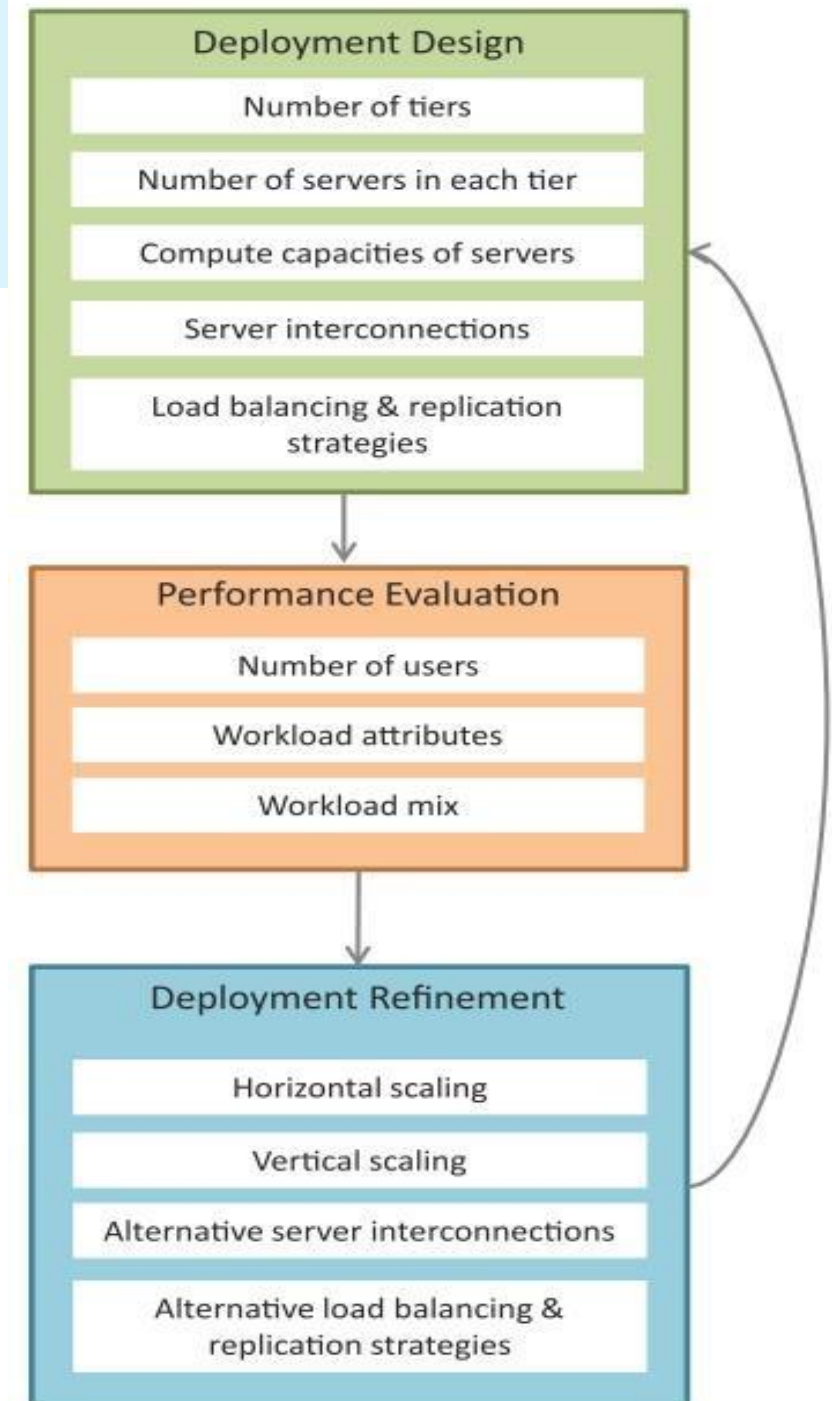- Stress Tests
  - Stress tests load the application to a point where it breaks down.
  - These tests are done to determine how the application fails, the conditions in which the application fails and the metrics to monitor which can warn about impending failures under elevated workload levels.
- Soak Tests
  - Soak tests involve subjecting the application to a fixed workload level for long periods of time.
  - Soak tests help in determining the stability of the application under prolonged use and how the performance changes with time.

# Deployment Prototyping

- Deployment prototyping can help in making deployment architecture design choices.

- Deployment design is an iterative process that involves the following steps:

- Deployment Design
  - Create the deployment with various tiers as specified in the deployment configuration and deploy the application.

- Performance Evaluation
  - Verify whether the application meets the performance requirements with the deployment.

- Deployment Refinement
  - Deployments are refined based on the performance evaluations. Various alternatives can exist in this step such as vertical scaling, horizontal scaling, for instance.



**Deployment Design**
- Number of tiers
- Number of servers in each tier
- Compute capacities of servers
- Server interconnections
- Load balancing & replication strategies

**Performance Evaluation**
- Number of users
- Workload attributes
- Workload mix

**Deployment Refinement**
- Horizontal scaling
- Vertical scaling
- Alternative server interconnections
- Alternative load balancing & replication strategies

# Performance Evaluation Workflow

## Semi-Automated Workflow (Traditional Approach)

- In traditional approach, a real user's interactions with a cloud application are first recorded as virtual user scripts.

- The recorded virtual user scripts then are parameterized to account for randomness in application and workload parameters.

- Multiple scripts have to be recorded to create different workload scenarios. This approach involves a lot of manual effort.

- To add new specifications for workload mix and new requests, new scripts need to be recorded and parameterized.

- Traditional approaches which are based on manually generating virtual user scripts by interacting with a cloud application, are not able to generate synthetic workloads which have the same characteristics as real workloads.

- Traditional approaches do now allow rapidly comparing various deployment architectures.
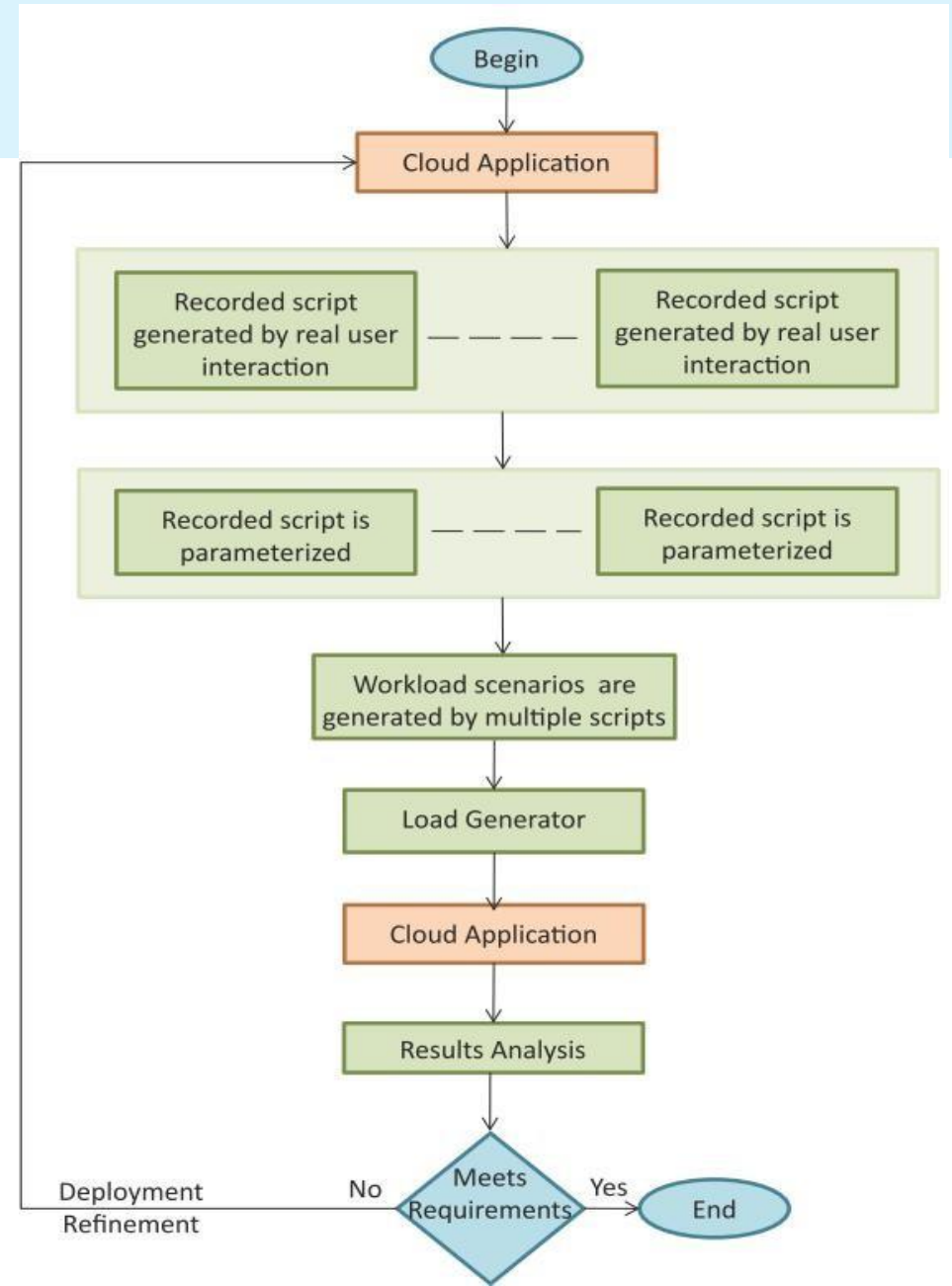
# Performance Evaluation Workflow

## Fully-Automated Workflow (Modern Approach)

- In the automated approach real traces of a multi-tier application which are logged on web servers, application servers and database servers are analyzed to generate benchmark and workload models that capture the cloud application and workload characteristics.

- A statistical analysis of the user requests in the real traces is performed to identify the right distributions that can be used to model the workload model attributes.

- Real traces are analyzed to generate benchmark and workload models.

- Various workload scenarios can be created by changing the specifications of the workload model.

- Since real traces from a cloud application are used to capture workload and application characteristics into workload and benchmark models, the generated synthetic workloads have the same characteristics as real workloads.

- An architecture model captures the deployment configurations of multi-tier applications.