# Performance

- How well resource requirements of an application are being met.
- Expressed in terms of latency, memory, bandwidth and (for mobile apps) battery consumption metrics.
- Usually apps involving playout of multimedia content, 2D/3D graphics/animation, AR/VR and data analytics etc. impose resource requirements on the environment.
- Performance Testing involves measuring and/or profiling latency, memory, bandwidth and battery consumption of an app under normal working conditions.
  - Involves use of system calls and/or profilers provided by the software development kits.
  - e.g.SystemClock, System.nanoTime(), Hierarchy Viewer, Debug, Traceview, Runtime, Network Profiler, BatteryManager

# 性能

- 应用程序的资源需求在多大程度上得到了满足。
- 通常以延迟、内存占用、带宽以及（针对移动应用）电池消耗等指标来衡量。
- 通常涉及多媒体内容播放、2D/3D图形/动画、增强现实（AR）/虚拟现实（VR）以及数据分析等的应用程序，会对运行环境提出资源需求。
- 性能测试是指在正常工作条件下，对应用程序的延迟、内存占用、带宽及电池消耗等指标进行测量和/或性能剖析。
  - 该过程需借助软件开发工具包（SDK）所提供的系统调用和/或性能分析器。
  - 例如：SystemClock、System.nanoTime()、Hierarchy Viewer、Debug、Traceview、Runtime、Network Profiler、BatteryManager

## Scalability

- Scalability is how performance trends as various system, environment or contextual factors change.
- Scalable architectures aim at ensuring that the application either continues to perform well as these factors stress the application or at least degrades gracefully without resulting in system failure when the system is stressed.
- Conversely, a scalable system is one whose performance improves in proportion to the resources being added.

## Scalability Testing

- Load Testing involves incrementing the load progressively and monitoring the resulting trends in the performance in terms of latency or throughput.
- Commonly used system scalability/load testing tools include LoadRunner, JMeter.

## Scalability Models

- The increase in computational latency or memory requirements of an algorithm, scheduling scheme or data structure as the size of the input increases is typically expressed using O notation. O(1), O(log N), O(N), O($N^2$), O(N!) and O($N^N$) represent a deteriorating trend in performance as the input size N increases.
- Queuing models are used for representing statistical trends in the performance in response to a random input process and predict impact of load balancing and scalability of architectures.

Note: The PhotoGallery app is not a queuing system and therefore the appropriate scalability or load test would be the performance of the location, time or keyword search as the number of pictures in the storage increase.

## 可扩展性

- 可扩展性是指当各类系统、环境或上下文因素发生变化时，系统性能的变化趋势。
- 可扩展的架构旨在确保：当上述因素对应用造成压力时，应用仍能保持良好性能；或至少在系统承压时以优雅方式降级，而不会导致系统失效。
- 反之，可扩展的系统是指其性能随所增加资源呈比例提升的系统。

## 可扩展性测试

- 负载测试通过逐步增加负载，并监控延迟或吞吐量等性能指标的变化趋势来实现。
- 常用的系统可扩展性/负载测试工具包括 LoadRunner 和 JMeter。

## 可扩展性模型

- 算法、调度方案或数据结构在输入规模增大时，其计算延迟或内存需求的增长趋势通常用大 O 记号（O notation）表示。O(1)、O(log N)、O(N)、O($N^2$)、O(N!) 和 O($N_N$) 表示随着输入规模 N 的增大，性能呈现逐步下降的趋势。
- 排队模型用于表征系统在随机输入过程下的性能统计趋势，并预测负载均衡及架构可扩展性对性能的影响。

注意：PhotoGallery 应用并非排队系统，因此恰当的可扩展性或负载测试应关注位置搜索、时间搜索或关键词搜索的性能表现——即当存储中的图片数量增加时，上述搜索功能的性能变化。

## Queuing Models

M/M/1 queue


- For an M/M/1 queue, the average time a request/transaction spends in the system is given as follows:

  E [Tsystem] = 1 / [μ – λ],

where λ is the average rate at which requests/transactions arrive at the system, μ is average service rate (inversely proportional to the average request/transaction time/length), under the condition that μ >> λ.

The arrival process is assumed to be Poisson distributed and the service process is assumed to be exponentially distributed.

- Example: Assuming that a component that could be modelled as an MM1 queue in a system, can process 10 requests/transactions per second on the average, and the average arrival rate of the requests to the component is 3 requests per second, the average latency or time it takes for a request to be handled by the system is then

  1/[10-3] = 0.14 seconds.

## 排队模型

M/M/1 排队系统


- 对于 M/M/1 排队系统，请求/事务在系统中所花费的平均时间计算公式如下：

E [Tsystem] = 1 / [μ – λ]，

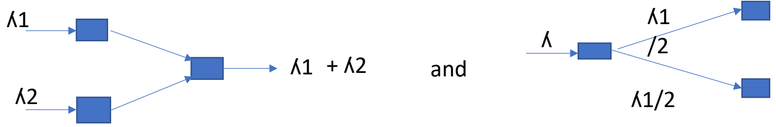
其中 λ 表示请求/事务到达系统的平均速率，μ 表示平均服务速率（与平均请求/事务处理时间/长度成反比），且需满足条件 μ >> λ。

假设到达过程服从泊松分布，服务过程服从指数分布。

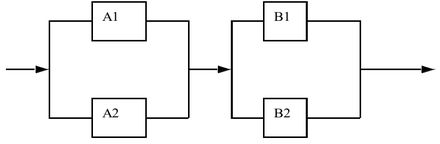
- 示例：假设系统中某个组件可建模为 M/M/1 排队模型，其平均处理能力为每秒 10 个请求（或事务），而请求到达该组件的平均速率为每秒 3 个，则请求在系统中被处理所需的平均延迟（即平均响应时间）为

  1/[10-3] = 0.14 seconds.

- For Poisson arrival processes and exponentially distributed service processes, the following relationships hold:



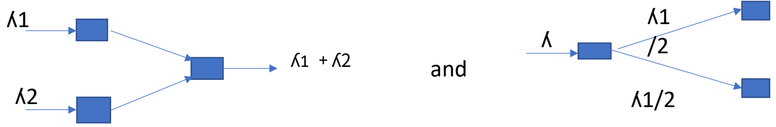

- The average time a request spends in a system that is composed of multiple such queues, arranged in tandem, is simply the sum of the average times at each queue, as long as the number of queues in such network stays small. The two M/M/1 queues of the previous example connected in tandem will thus have E[Tsystem] = .14 + .14 = .28s
- Example:



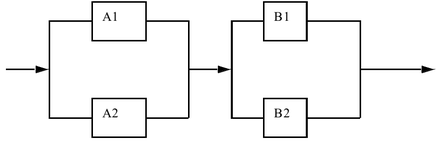

- Assuming that each component, in the above system, on the average can process 10 requests/transactions per second and the average arrival rate of the requests to the system is 3 requests per second, the average latency or time it takes for a request to be handled by the system assuming that the system could be modeled as a network of queues is estimated as:

1/[10-1.5] + 1/[10-1.5] = 0.25 seconds.

- 对于泊松到达过程和指数分布的服务过程，以下关系成立：




- 若一个系统由多个此类队列串联组成，则请求在该系统中的平均停留时间，即为各队列平均停留时间之和，前提是此类网络中的队列数量保持较少。因此，前述示例中两个串联的 M/M/1 队列的平均停留时间为 E[Tsystem] = .14 + .14 = .28s
- 示例：




- 假设上述系统中每个组件平均每秒可处理 10 个请求/事务，且请求到达系统的平均速率为每秒 3 个请求；若将该系统建模为一个队列网络，则请求被系统处理所需的平均延迟（即平均耗时）估计为：

1/[10-1.5] + 1/[10-1.5] = 0.25 seconds.