系统性能的测量与分析

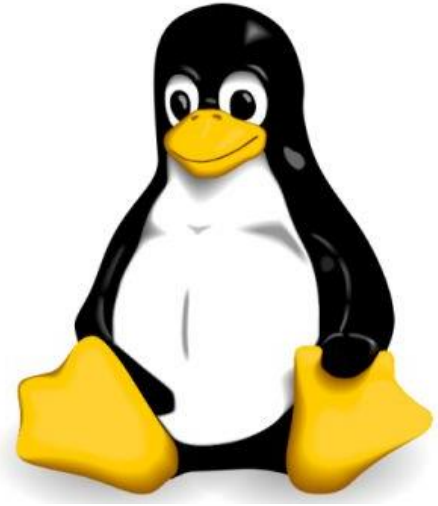
贾枭
上海皮格猫信息科技有限公司


SHANGHAI JIAO TONG UNIVERSITY
1896


NUS
National University
of Singapore


Max
Planck
Institute
for
Software Systems

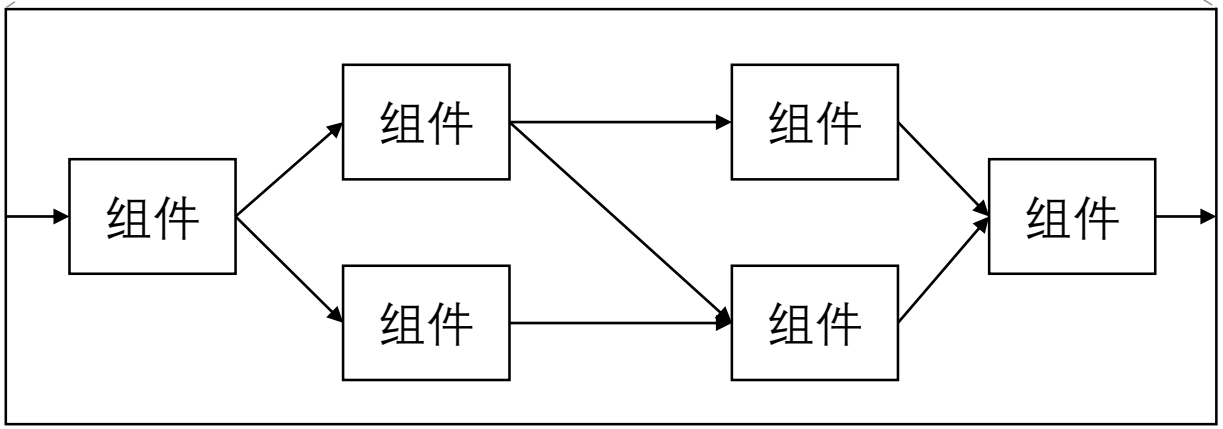

Google Cloud Platform



什么是系统？

黑盒模型

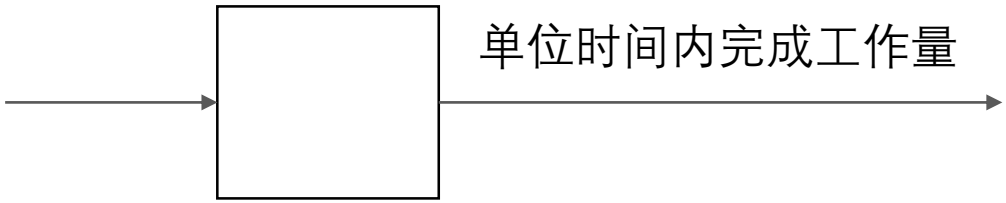
白盒模型



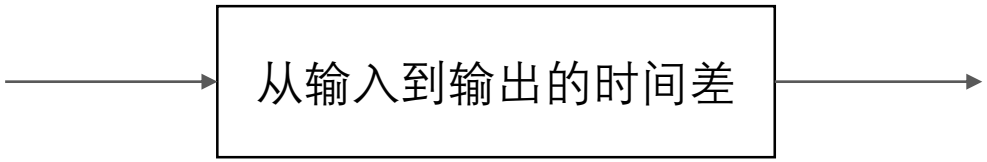
什么是性能？

常用的两个衡量性能的指标：

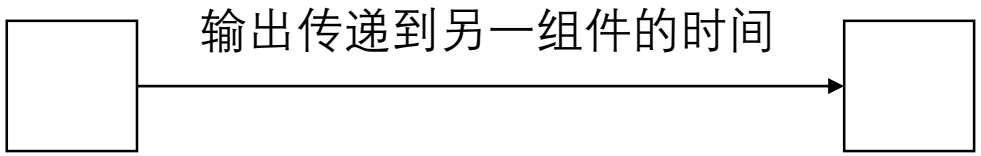
吞吐量 Throughput



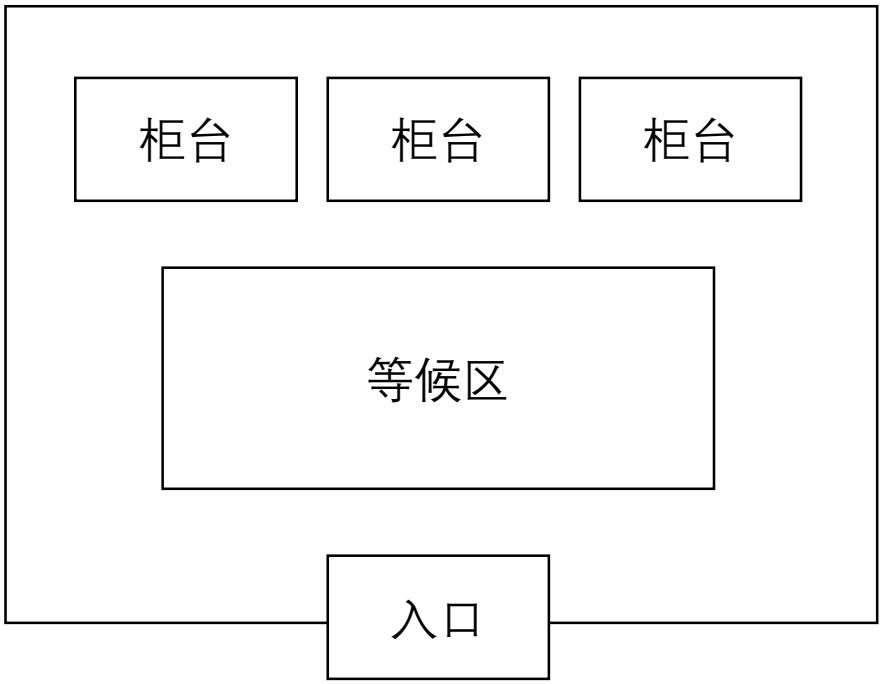
延迟 Latency



（两种模型等价）



银行模型

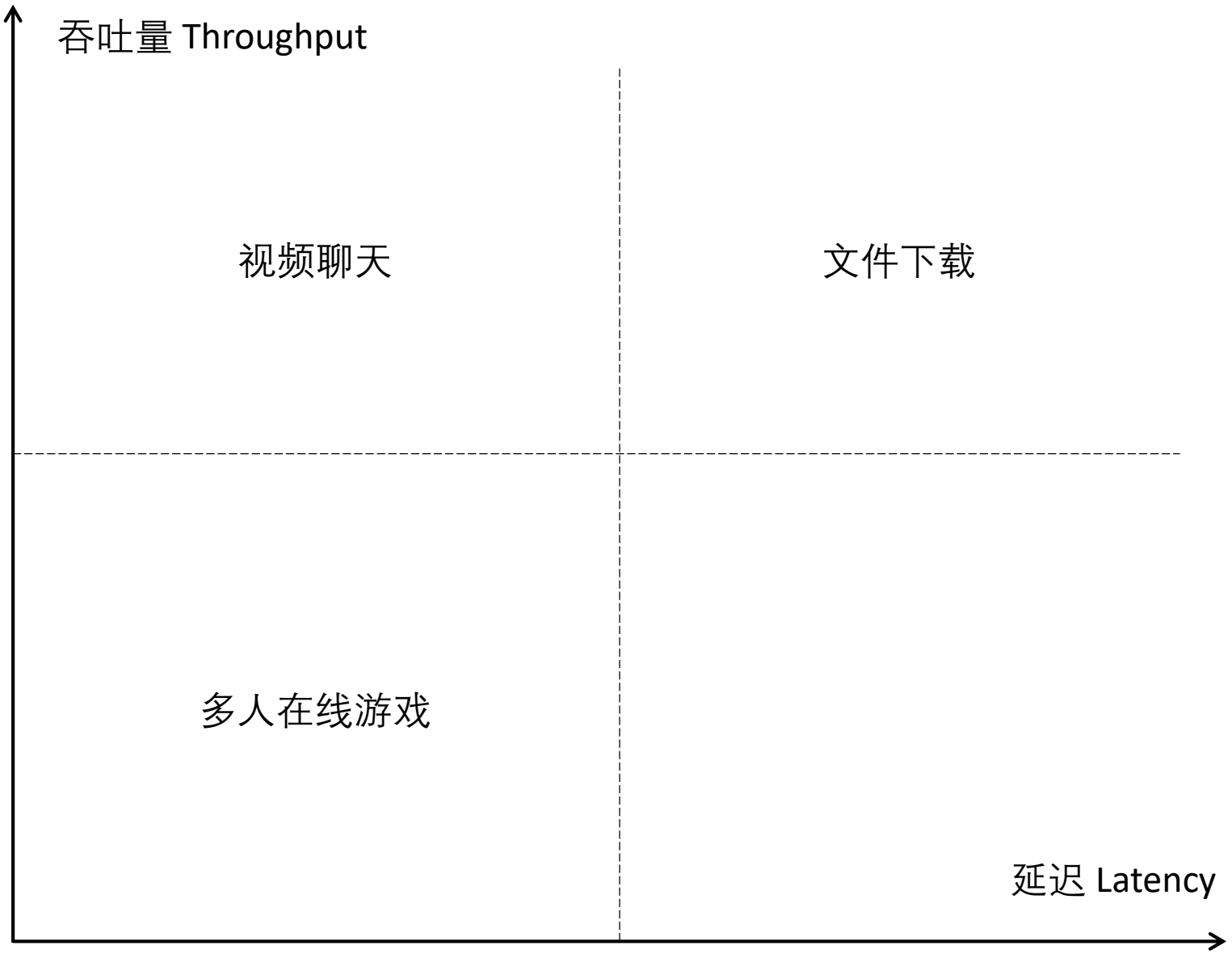


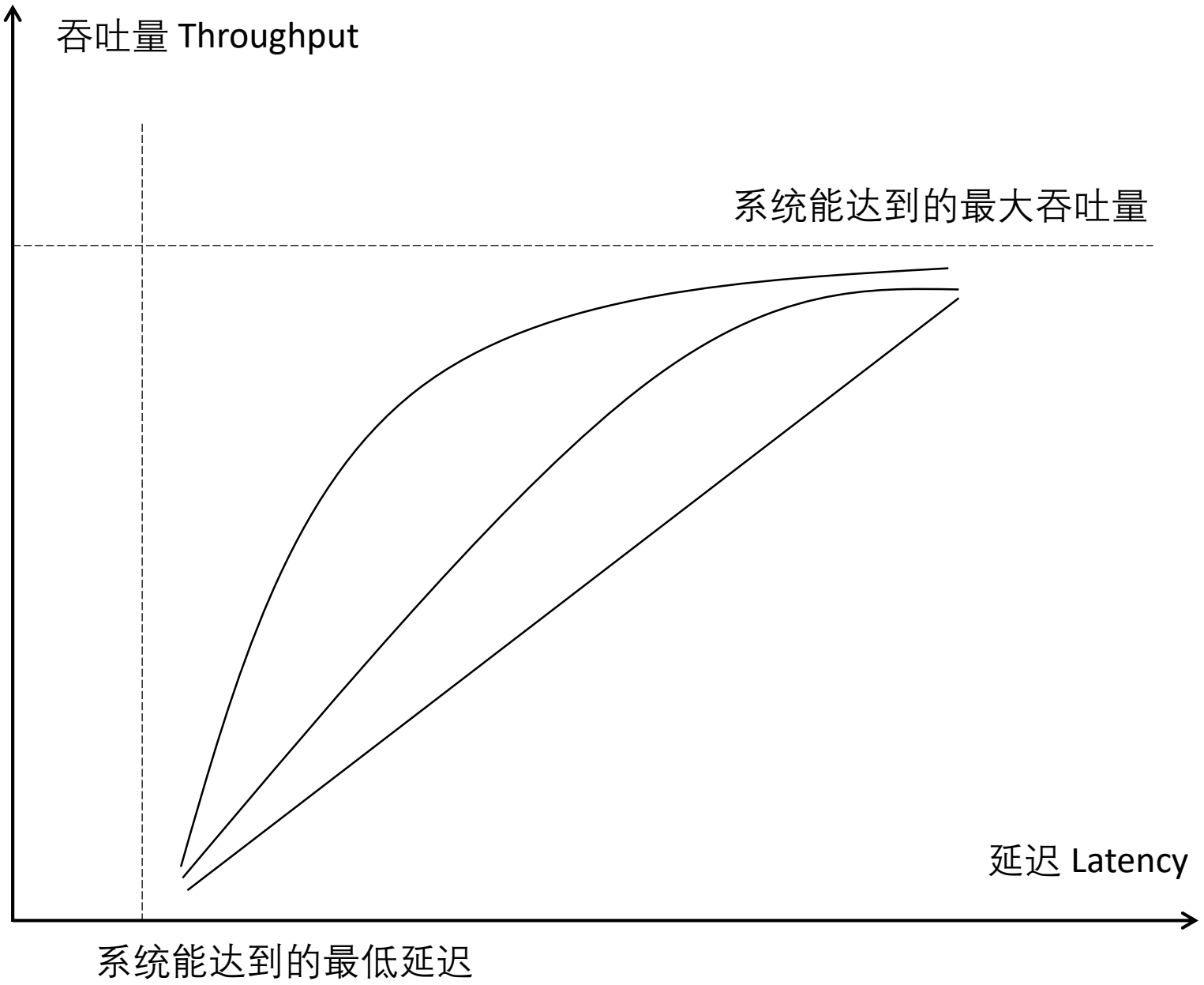
每个柜台的处理性能

有几个柜台

等候区有多大

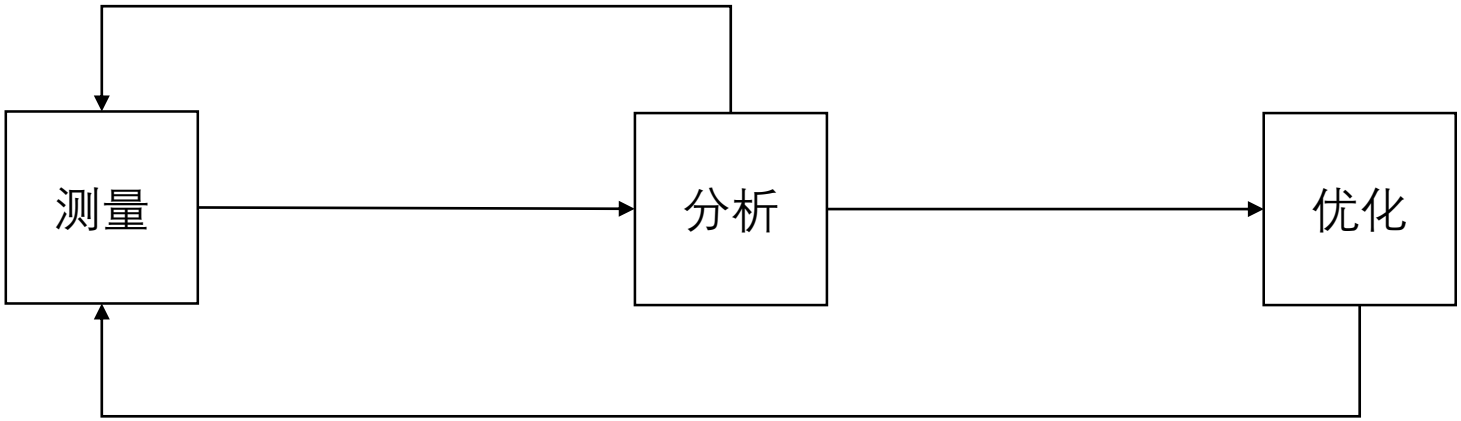
客户进入的速率


吞吐量 Throughput
视频聊天
文件下载
多人在线游戏
延迟 Latency


吞吐量 Throughput
系统能达到的最大吞吐量
延迟 Latency
系统能达到的最低延迟

测量与分析的关系

在系统保持不变的情况下，调整测量方法



在系统改变后，再次从测量开始

理论作为指导至关重要：对计算机系统的整体理解，以及对被测系统的理解

性能测量的核心是对时间的测量

基本问题：如何测量一个过程执行所需的时间？

```
void do_something() { ... }
```

性能测量的核心是对时间的测量

基本问题：如何测量一个过程执行所需的时间？

```
void do_something() { ... }

void measure() {

        start = GetCurrentTime();
        do_something();
        latency = GetCurrentTime() – start;

}
```

常见错误：只测量一次

性能测量的核心是对时间的测量

基本问题：如何测量一个过程执行所需的时间？

```
void do_something() { ... }

void measure() {
    for (sum = 0, t = 1; t <= 3; t++) {
        start = GetCurrentTime();
        do_something();
        sum += GetCurrentTime() – start;
    }
    latency = sum / 3;
}
```

常见错误：测量三次取平均值

性能测量的核心是对时间的测量

基本问题：如何测量一个过程执行所需的时间？

```
void do_something() { ... }

void measure() {
    for (t = 1; t <= 30; t++) {
        start = GetCurrentTime();
        do_something();
        sample[t] = GetCurrentTime() – start;
    }
}
```

常见错误：盲目使用经验法则，或“不舍得”花时间获取数据

性能测量的核心是对时间的测量

基本问题：如何测量一个过程执行所需的时间？

```
void do_something() { ... }

void measure() {
    for (t = 1; t <= 1000; t++) {
        start = GetCurrentTime();
        do_something();
        sample[t] = GetCurrentTime() – start;
    }
}
```

建议：　在可控的时间内，获取尽量多的有效数据。

根据分析阶段对样本分布、置信度与置信区间的选取，调整样本量。

这样做可能会有哪些问题？

GetCurrentTime() 的精度是多少？

1秒

1毫秒 ~ 1微秒　　如何实现？

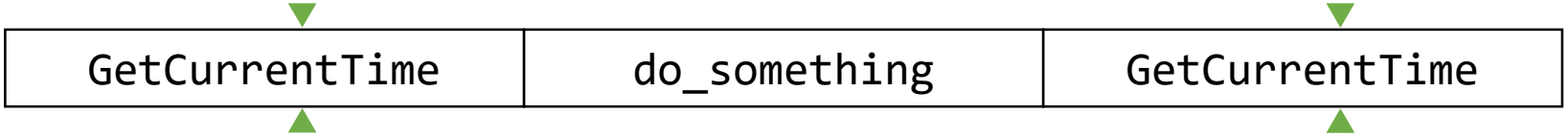
纳秒　　如何实现？

```
void measure() {
    for (t = 1; t <= 1000; t++) {
        start = GetCurrentTime();
        for (n = 1; n <= 1000; n++) do_something();
        sample[t] = GetCurrentTime() – start;
    }
}
```

统计学上的意义是否有改变？

这样做可能会有哪些问题？

什么是“当前时间”？



相关问题：GetCurrentTime() 本身需要多少时间？

基本保证：对当前时间的测量偏差，不超过 GetCurrentTime() 本身所需时间

延伸问题：如何估计这一偏差？这一偏差自身有多稳定？分布形态？

性能测量的核心是对时间的测量

更常见的问题：如何测量一个服务中某一部分执行所需的时间？

```
void server() {
    while (!state->stop) {
        do_action_1(&state);

        do_action_2(&state);

        do_action_3(&state);
    }
}
```

性能测量的核心是对时间的测量

更常见的问题：如何测量一个服务中某一部分执行所需的时间？

```
void server() {
    while (!state->stop) {
        do_action_1(&state);
        start = GetCurrentTime();
        do_action_2(&state);
        sample[++t] = GetCurrentTime() – start;
        do_action_3(&state);
    }
}
```

这样做可能会有哪些问题？

样本太多，保留哪些？

保留最开始的N个　　如何实现？

保留最近的N个　　如何实现？

如何保留所有样本？

`GetCurrentTime()` 本身需要多少时间？

1毫秒　　1,000 QPS

1微秒　　1,000,000 QPS

如何测量超过百万 QPS 的服务？

性能测量的核心是对时间的测量

另一个常见的问题：服务的瓶颈在哪里？

即：　　程序哪里最慢？

或：　　CPU在哪里停留的时间最长？

思路一：对每一段代码，逐一测量它们的执行时间

思路二：时不时地暂停程序，观察CPU在执行哪一段代码

如何实现？分别有什么问题？又如何应对？

性能测量的核心是对时间的测量

常见但更困难的问题：如何测量两台机器间的通信性能？

难以让两台机器时钟同步

时间从哪里来？

思路一：测量 round-trip time (RTT) 来估计延迟

思路二：假设信号稳定送达，发送多个信号并分别记录两地时间，估计波动

分析性能数据

类型一：第 N 个样本的测量值为 X（如时间差）

每个样本包含的测量值可能有多个

常见错误：忽略 N（样本次序）

类型二：在时间 T 的测量值为 X（如内存占用、吞吐量、Program Counter）

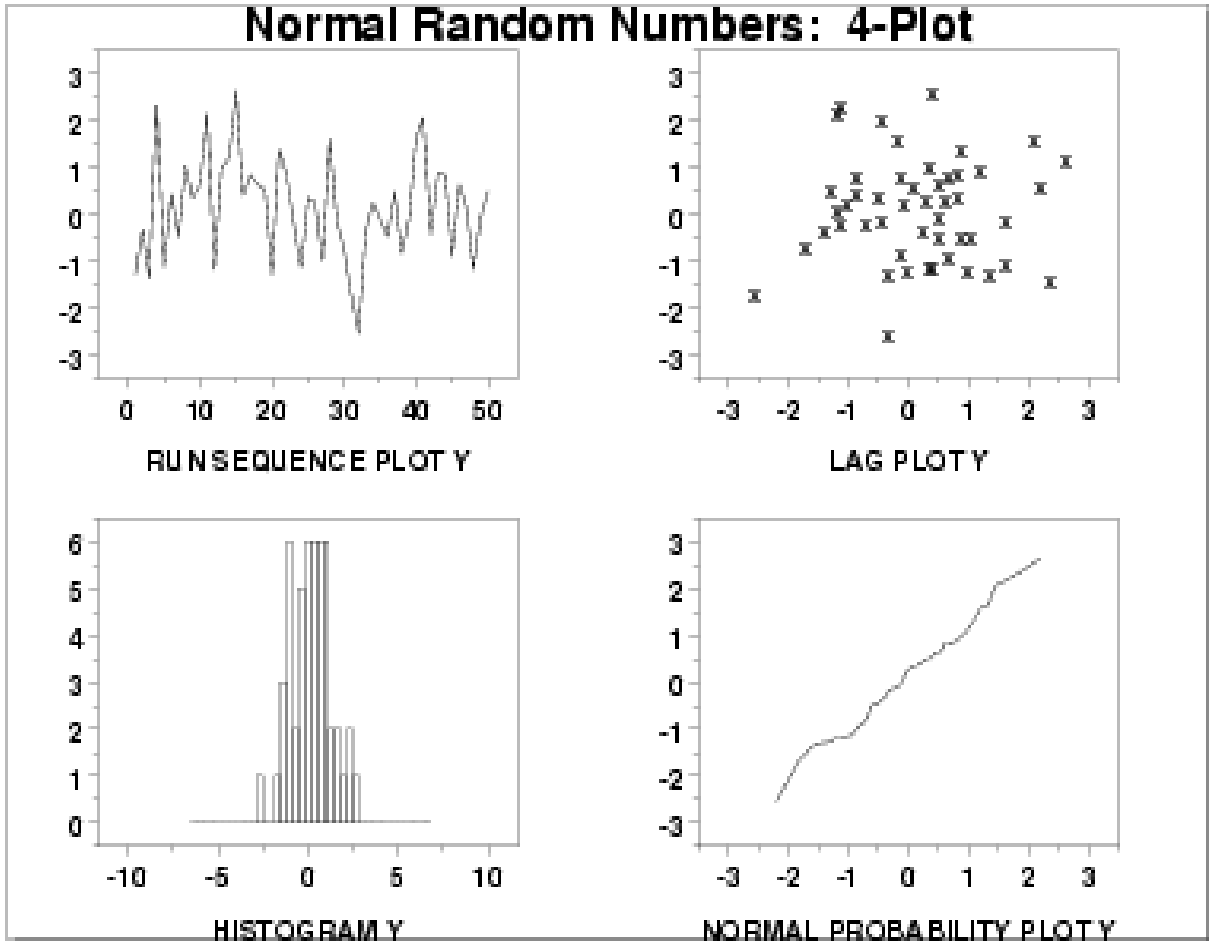
每个样本包含的测量值可能有多个

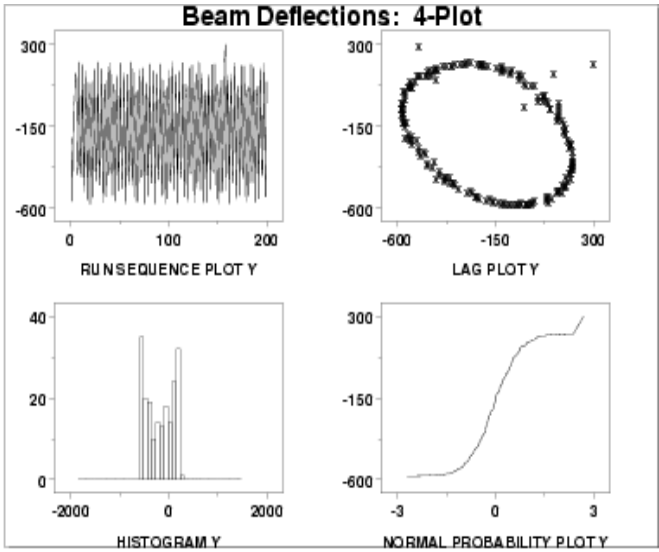
如何在同一时间测量多个值？

测量时间 T 是否要间隔均匀？如何实现？

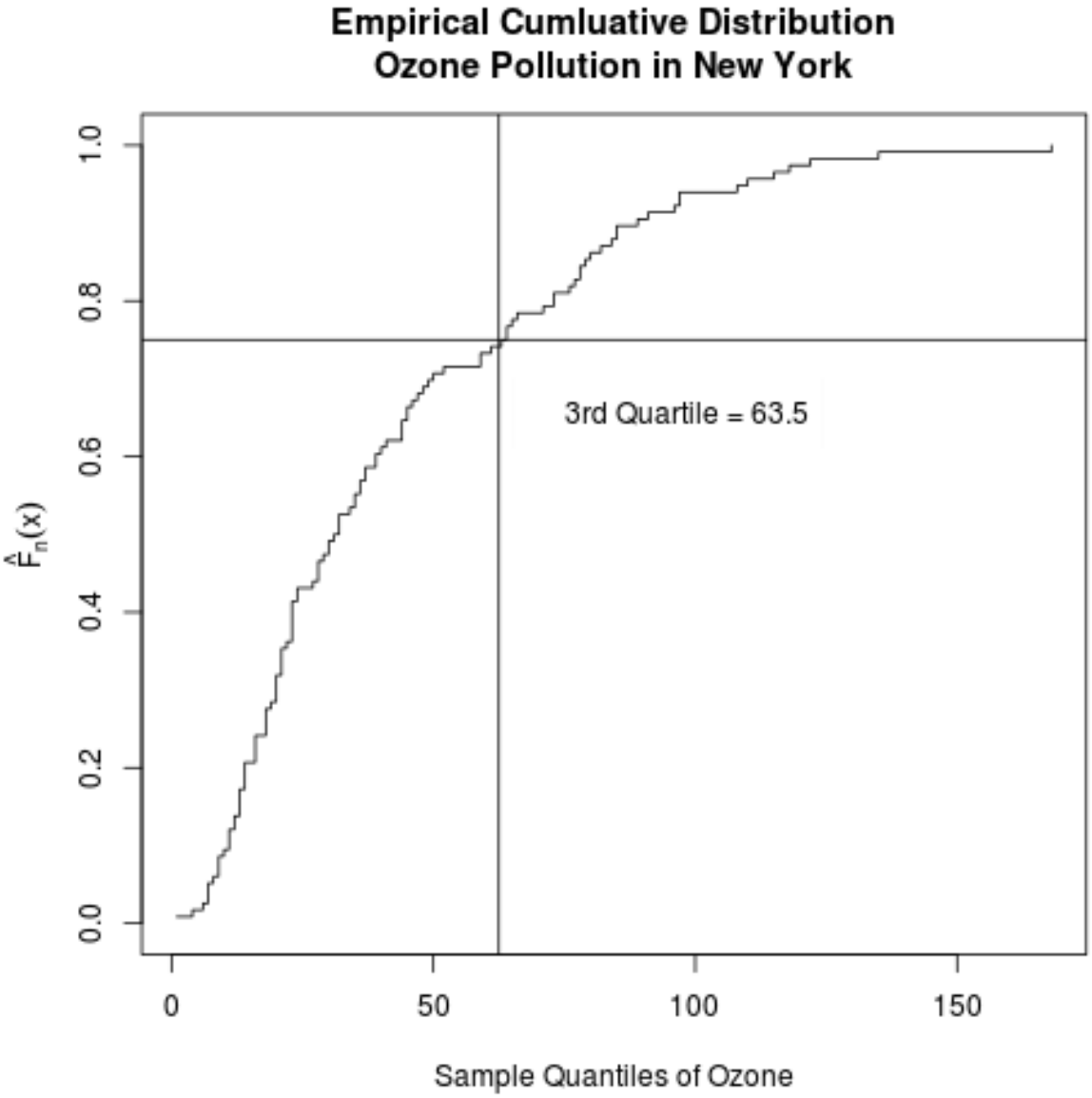
探索性数据分析 Exploratory Data Analysis

四张图：run sequence plot;　lag plot;　histogram;　normal probability plot





代替直方图：ECDF


Empirical Cumluative Distribution
Ozone Pollution in New York
3rd Quartile = 63.5
$\hat{F}_n(x)$
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
Sample Quantiles of Ozone

测量数据为什么会呈现规律性？

```
void server() {
    while (!state->stop) {
        do_action_1(&state);
        start = GetCurrentTime();
        do_action_2(&state);
        sample[++t] = GetCurrentTime() – start;
        do_action_3(&state);
    }
}
```

该问题必要性：样本之间是相互独立的吗？

如何检验测量数据的规律性？

常见方式：　对 X ~ N 做线性回归，尝试解释回归系数非零的原因

计算 partial autocorrelation function 并画图

如何对测量数据进行定量分析？

常用统计值：最大值、最小值、平均值、中位数、标准差

假设中一定会需要有置信度和置信区间

如何比较系统改动前后的性能？

常见方式：　明显的变化也要注意检验标准差、置信区间

没有明显瓶颈的“长尾”系统，考虑做频域分析

实习机会

大规模建筑模型在 VR 中的实时渲染

实现不同的光线追踪算法，并进行性能评估与优化

并行化已有算法，在上百核+上百G内存的机器上进行优化

地点：上海 · 零号湾（5000/月）

高性能数字加密货币交易系统

参与交易系统的开发与测试、区块链技术的研究与开发

实现超低延迟网络下的高频套利、做市算法

地点：北京 · 三元桥（5000/月，包食宿）

10月28日交大分享

该二维码7天内(11月4日前)有效，重新进入将更新