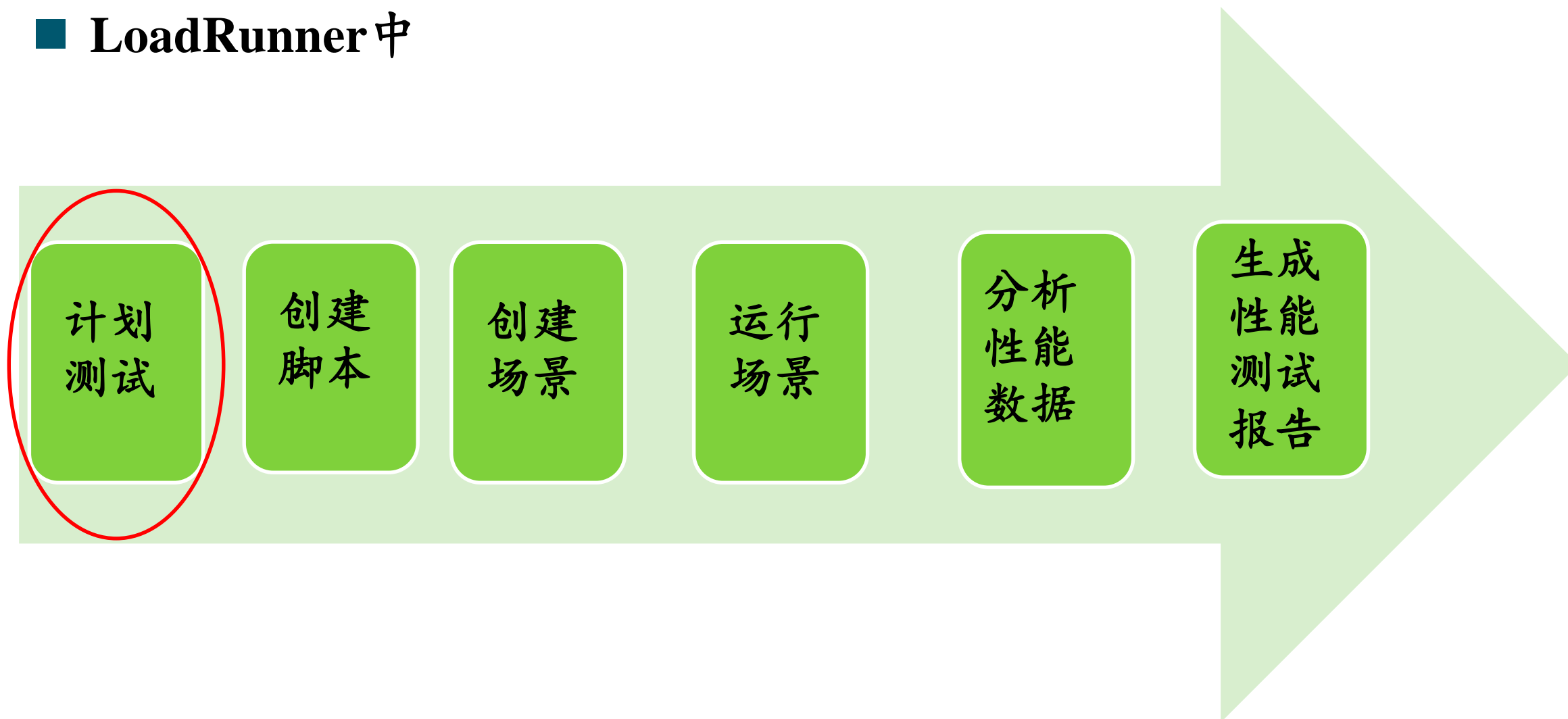


性能测试

--计划测试—分析系统

相关知识回顾—性能测试流程

■ LoadRunner中



计划测试流程



目录

- 熟悉业务流程
- 定义测试目标
 - 通过用户数据进行分析
 - 通过系统日志进行分析

熟悉业务流程

- 查看相关文档
- 实际运行执行
- 使用VuGen录制回放
 - 确定能够使用性能测试工具模拟用户行为
 - 获得系统相关信息
 - 需要关联的动态数据（比如：session，页面参数）
 - 用户操作返回的ID等

目录

■ 熟悉业务流程

■ 定义测试目标

- 通过用户数据进行分析
- 通过系统日志进行分析

定义测试目标

- 思考：SRS（需求规格说明书）能不能包含所有的用户需求？
 - 不能
- 思考：用户自身能不能提供准确有效的性能需求？
 - 不能
 - 性能需求分析工程师对其进行显示和隐式分析

分析测试目标的目的（性能测试的关键）

- 知道用户会做什么
- 知道用户怎么做这件事
- 知道多少用户做这件事情
- 知道对应的性能指标

分析性能测试目标

■ 思考：地铁站电梯和换乘大厅的楼梯应该修建多宽？

— 功能问题？

— 性能问题？

- 如果修建过窄，导致大量乘客无法疏散出月台引起事故
- 如果修建过宽，带来不必要的浪费
- 合适的疏散能力就是性能的吞吐量指标

目录

- 熟悉业务流程
- 定义测试目标
 - 通过用户数据进行分析
 - 通过系统日志进行分析

通过用户提供的数据进行分析

■ 分析哪些数据

- 容量
- 响应时间
- 并发量
- 服务器资源

通过用户提供的数据进行分析

- 举例：用户业务数据：OA系统，公司有500个用户会使用该系统，主要在上面完成各种订单的内部处理，每笔业务的提交大概平均15分钟，每个负责订单用户每天提交20笔订单，高峰期会有33个订单的提交量。整个公司大概有400个用户负责订单提交，50个用户负责订单审查，每个订单会被两个审查人员复审，复审的平均时间为5分钟

通过用户提供的数据进行分析

■ 每秒钟几笔操作？

— 最悲观的考虑方式：按500个用户全部做订单提交操作，每个用户的提交时间按照10-12分钟计算，可以得到系统的处理能力需要达到：

- $500 * (60/10) = 3000$ 笔业务/小时
- 进一步计算得到 50 笔/分钟
- 每秒不到1笔操作
- 这基本没有并发量，不需要做性能测试

通过用户提供的数据进行分析

- 分析：不能按平均值进行计算，适当放松，避免多用户并发操作时，系统出现问题
- 需求修改：系统支持500用户并发操作订单提交操作，提供每小时3000笔以上的操作能力，单位时间内的处理能力大于10笔业务/秒（一般按峰值数据计算，峰值数据按平均业务的10倍计算）
 - 标准处理能力下，响应时间在2秒以内
 - 普通负载下，10个用户并发，响应时间在3秒以内
 - 50个用户以内的大用户并发，响应时间在5秒以内

通过用户提供的数据进行分析

- 按照每笔50个数据字段计算，一笔业务需要500KB的数据量
 - 系统对于100Mb/s带宽的设计能够支持多少个用户并发提交订单
 - $100 * 1000 / (500 * 8) = 25$ (个)
 - 由此能够确定响应时间在哪个范围是合适的
 - 50个用户以内的大用户并发，响应时间在5秒以内

通过用户提供的数据进行分析

- 数据库中每存放一笔订单需要1MB的磁盘空间，以每小时3000笔业务为基础加上8—10小时的工作时间，每天的业务应该有多少笔？需要开销多少磁盘空间？
 - 每天业务： $3000 * 10 = 30000$
 - 磁盘空间： $30000 * 1 \text{ MB} = 30\text{GB}$

通过用户提供的数据进行分析

■ 如上需求，如何进行性能测试？

- 如何设计场景？
- 如何评估服务器处理能力？
- 如何确定服务器配置及容量？

目录

- 熟悉业务流程
- 定义测试目标
 - 通过用户数据进行分析
 - 通过系统日志进行分析

通过系统日志进行分析

- 旧系统升级时，历史数据即系统日志是获得真实用户需求最有效的参考数据
 - 尤其是峰值并发量的计算通过日志才是最可靠的

通过系统日志进行分析

■ 怎样查看日志？

- 客户端发送Web请求到服务器端时，都会在访问日志中记录下来
- IIS
- Apache
- Nginx

通过系统日志进行分析

■ 以Apache为例：

1 打开httpd.conf文件，找到如下关于LogFormat的设置

```
<IfModule log_config_module>
#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
#
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common

<IfModule logio_module>
# You need to enable mod_logio.c to use %I and %O
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" %I %O" combinedio
</IfModule>
```

通过系统日志进行分析

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

- %h:发送请求到服务器的客户IP地址
- %l:客户端identd进程判断的RFC身份，输出“-”表示此处信息无效（将IdentityCheck指令设为 On 时，才有此信息）
- %u:HTTP认证系统得到的访问网页的客户标识
- %t:时间[日/月/年：时：分：秒 时区]
- %r:改变时间的输出形式

通过系统日志进行分析

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

- %>s:服务器返回给客户端的状态码
- %b:服务器返回给客户端的不包括响应头的字节数
- common:别名

通过系统日志进行分析

■ 设置 CustomLog，将CustomLog前的“#”号去掉

```
# The location and format of the access logfile (Common Logfile Format).  
# If you do not define any access logfiles within a <VirtualHost>  
# container, they will be logged here. Contrariwise, if you *do*  
# define per-<VirtualHost> access logfiles, transactions will be  
# logged therein and *not* in this file.  
#  
☐ CustomLog "logs/access.log" common  
  
#  
# If you prefer a logfile with access, agent, and referer information  
# (Combined Logfile Format) you can use the following directive.  
#  
#CustomLog "logs/access.log" combined  
</IfModule>
```

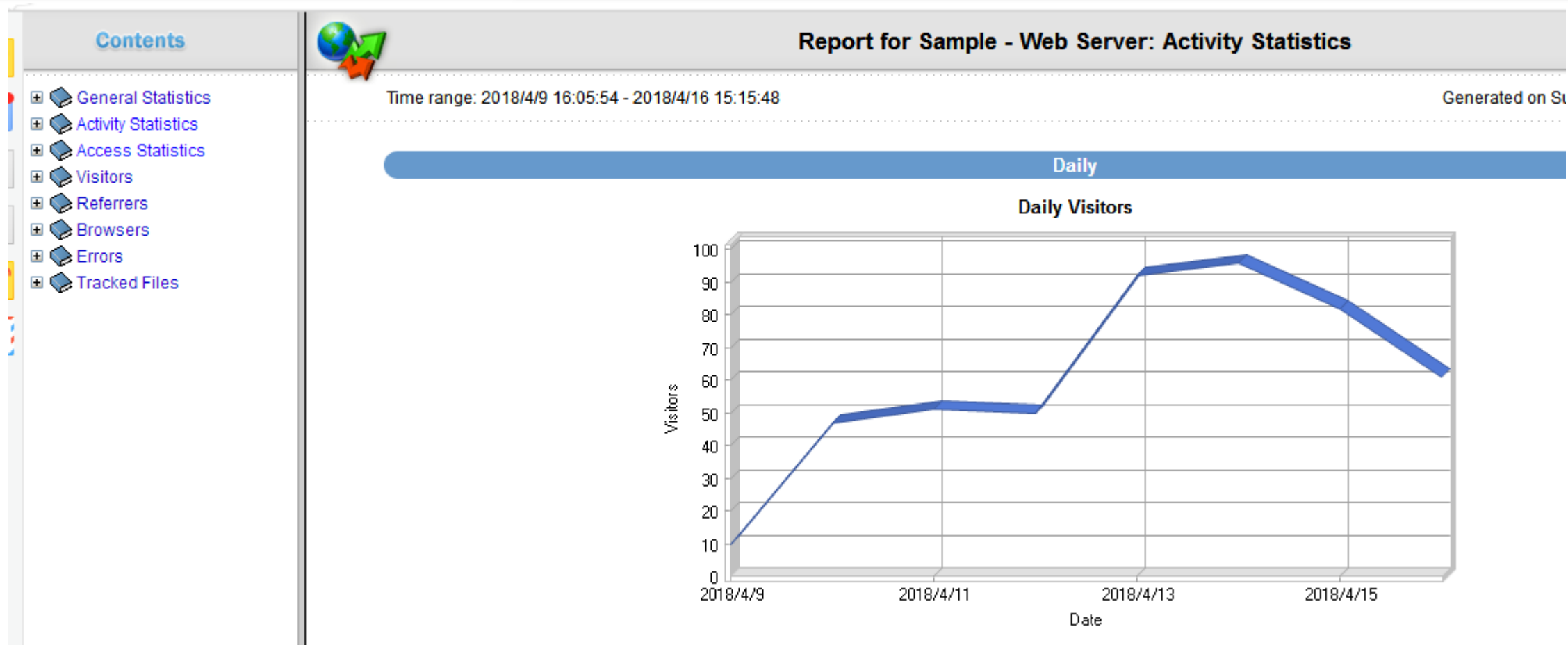
```
CustomLog "logs/access.log" common|
```


通过系统日志进行分析

■ 使用WebLog Expert分析日志

- 时间过滤规则 **Time Range :All activity**
- 输出格式和分析报告的内容: **Report Contents**

通过系统日志进行分析



通过系统日志进行分析

- 通过Summary可以粗略地了解系统当前的一些基础性能指标
 - 该时间段内，整个系统共有多少人访问，总点击量，总带宽
 - 是不是得到这些数据就能做性能测试了呢？
 - 不是

通过系统日志进行分析

- 平均数据和总计数据其实对于做性能测试没有什么用处
 - 因为访问不是平均的
 - 从测试角度讲：更在意峰值数据
 - 整个系统只要能够满足用户在真实访问下的峰值数据，就能保证整个系统能够满足用户的性能需求

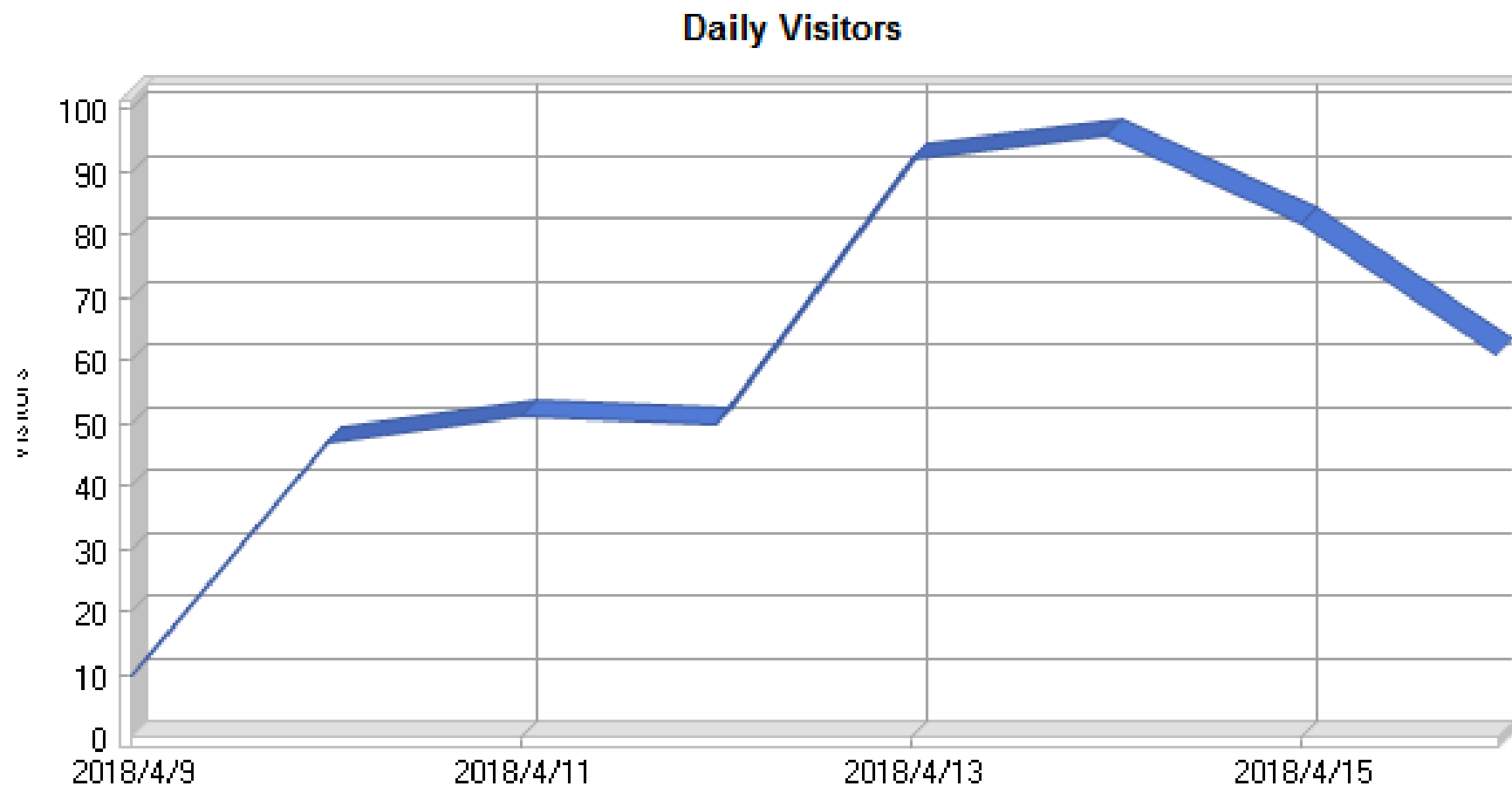
通过系统日志进行分析

■ 打开Activity Statistics 标签，查看访问量的统计

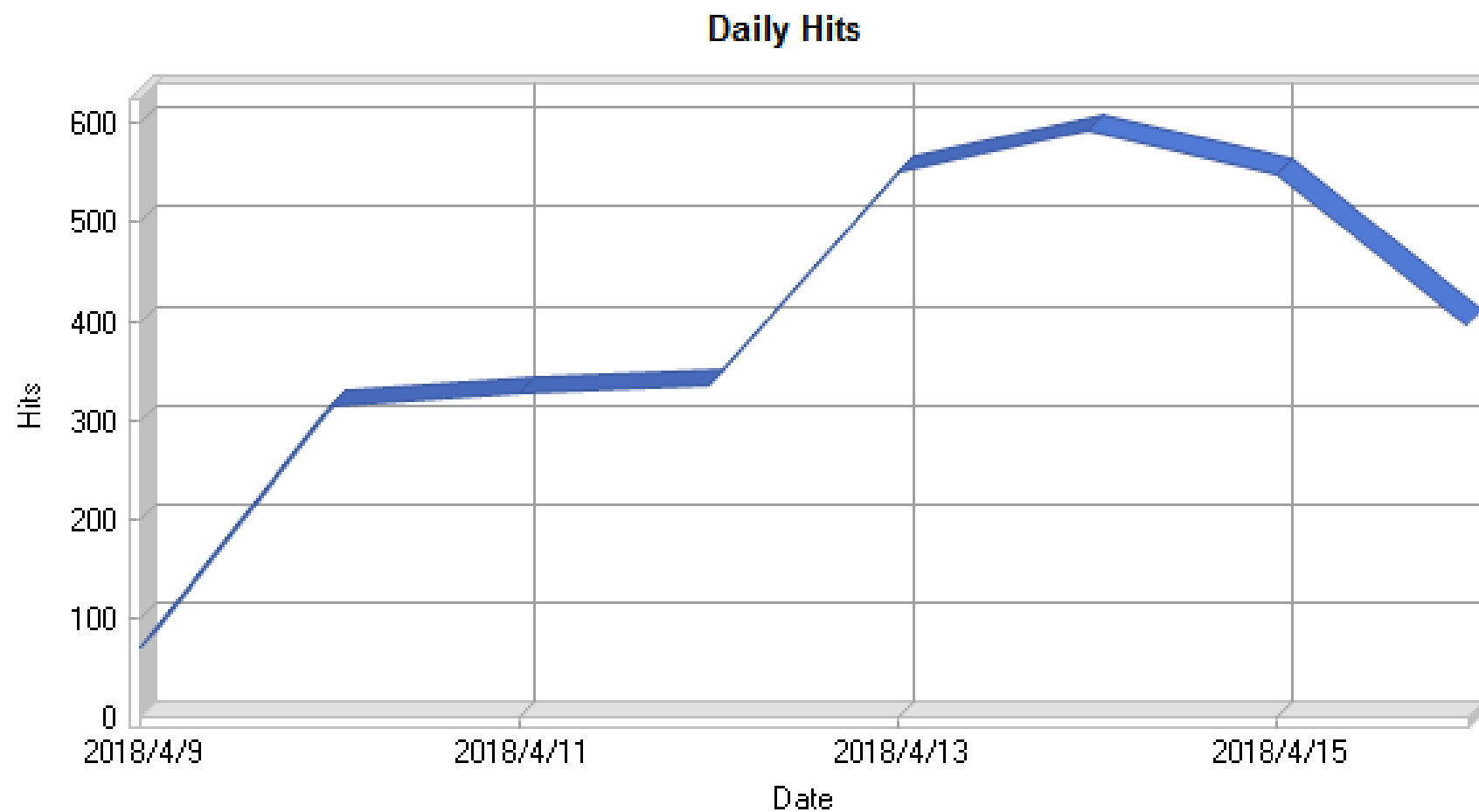
— 查看每个时间段的系统的访问数据

- 每日访问量统计
- 每日点击量统计
- 每日带宽统计
- 生成这些图所使用的每日数据分析表

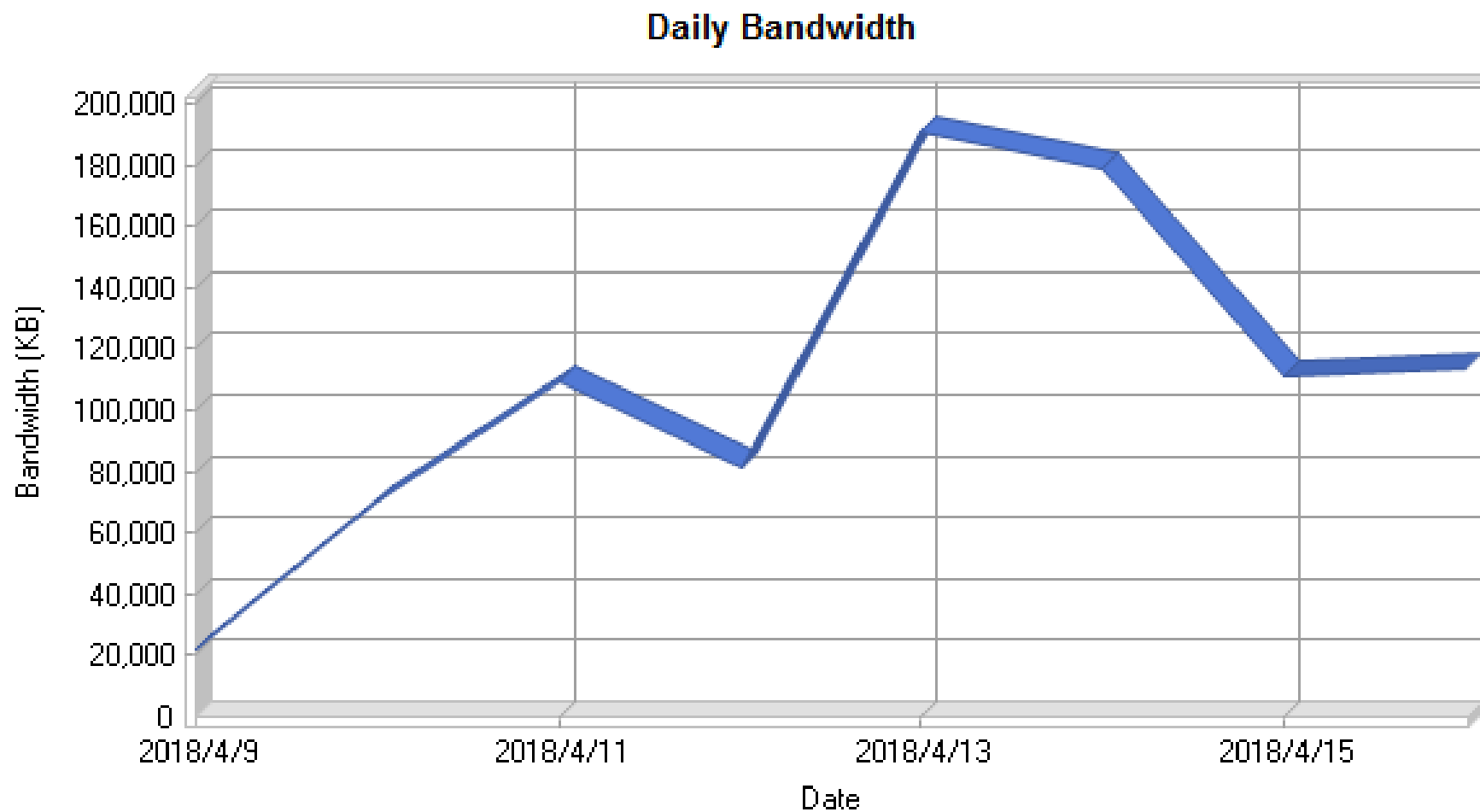
通过系统日志进行分析



通过系统日志进行分析



通过系统日志进行分析



通过系统日志进行分析

Daily Activity

Date	Hits	Page Views	Visitors	Average Visit Length	Bandwidth (KB)
Mon 2018/4/9	71	18	10	00:55	21,956
Tue 2018/4/10	314	86	47	00:48	70,264
Wed 2018/4/11	327	103	51	01:17	109,347
Thu 2018/4/12	335	111	50	01:16	81,251
Fri 2018/4/13	552	180	92	01:01	190,847
Sat 2018/4/14	592	196	96	01:11	178,831
Sun 2018/4/15	548	156	82	01:00	110,751
Mon 2018/4/16	396	107	61	01:00	113,163
Total	3,135	957	489	01:04	876,415

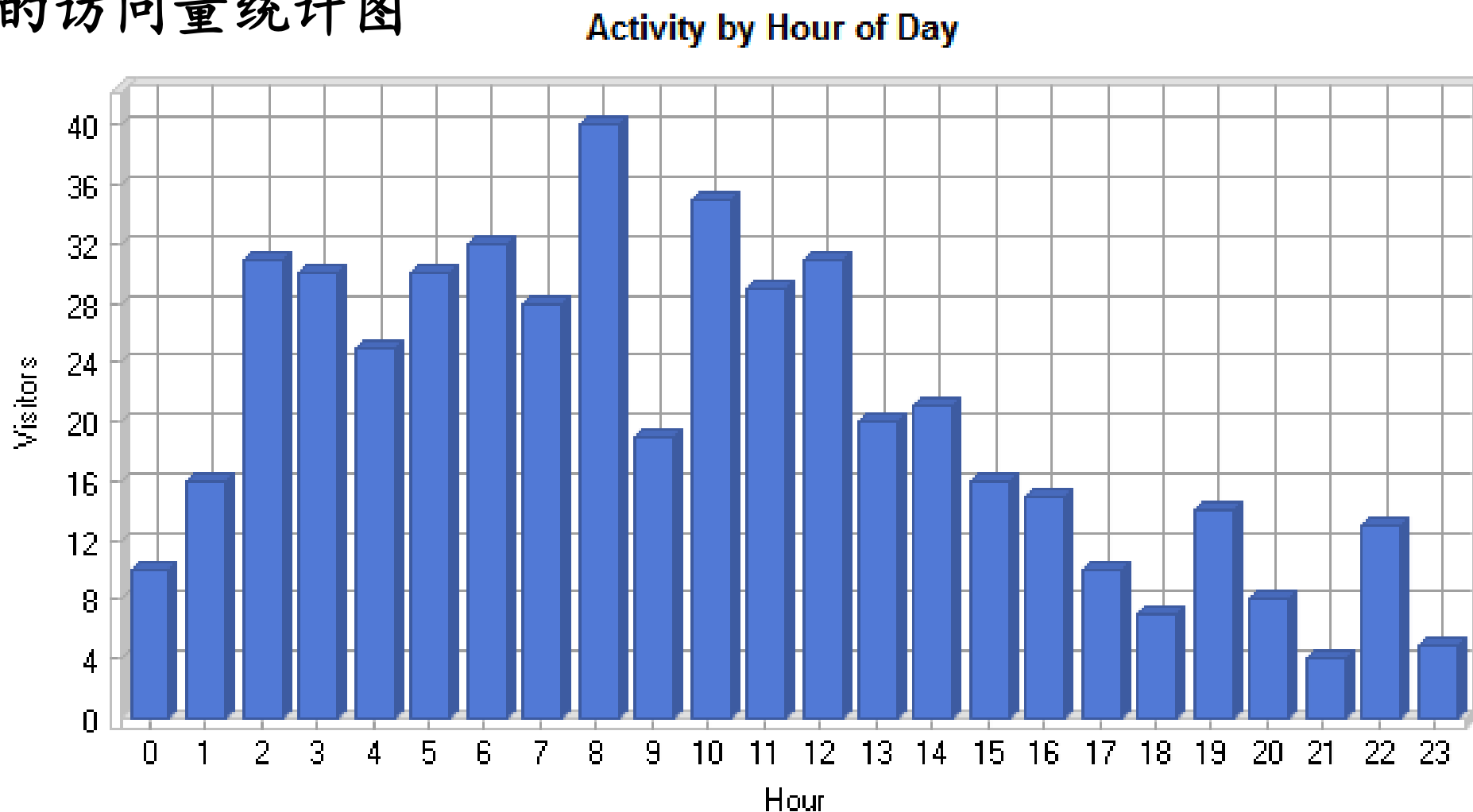
- 通过分析这些数据，可以获得系统的每日峰值吞吐量和峰值访问量，从而建立性能测试需求的指标

通过系统日志进行分析

- 是否可以参考上述数据，设计场景呢？
 - 不可以
 - 以天为单位显得太大，无法得到更精确的数据，不利于性能分析

通过系统日志进行分析

■ 查看每小时的访问量统计图



通过系统日志进行分析

Activity by Hour of Day

Hour	Hits	Page Views	Visitors	Bandwidth (KB)
00:00 - 00:59	47	19	10	8,601
01:00 - 01:59	115	33	16	9,840
02:00 - 02:59	184	55	31	29,269
03:00 - 03:59	156	64	30	54,423
04:00 - 04:59	132	46	25	38,565
05:00 - 05:59	176	53	30	67,819
06:00 - 06:59	216	72	32	65,821
07:00 - 07:59	190	46	28	63,046
08:00 - 08:59	251	67	40	92,376
09:00 - 09:59	140	42	19	43,815
10:00 - 10:59	204	65	35	40,107
11:00 - 11:59	190	64	29	68,234
12:00 - 12:59	214	65	31	50,254
13:00 - 13:59	144	40	20	36,246
14:00 - 14:59	124	37	21	35,920
15:00 - 15:59	123	35	16	30,601
16:00 - 16:59	104	32	15	38,110
17:00 - 17:59	53	20	10	24,162
18:00 - 18:59	50	13	7	6,124
19:00 - 19:59	94	28	14	19,703
20:00 - 20:59	61	16	8	14,067
21:00 - 21:59	24	7	4	5,656
22:00 - 22:59	114	27	13	25,342
23:00 - 23:59	29	11	5	8,301
Total	3,135	957	489	876,415

通过系统日志进行分析

■ 峰值数据分析

Hour	Hits	Page Views	Visitors	Bandwidth (KB)
08:00 - 08:59	251	67	40	92,376

Hour	Hits	Page Views	Visitors	Bandwidth (KB)
08:00 - 08:59	2 679 676	3 99 673	41 440	159 594 667

- 1小时内产生了260万次点击量，接近40万次页面刷新、4.1万个访问用户和159G的带宽

通过系统日志进行分析

- 假设带宽是1000Mb/s，每小时提供多少带宽
 - 每小时提供450GB的带宽
 - $1000\text{Mb} / 8$ （换算一下每秒提供125MB的带宽）
 - $125\text{MB} * 3600 / 1000 = 450 \text{ GB}$
 - 450GB 与 159GB相比，没有出现带宽瓶颈

通过系统日志进行分析

- 如果系统访问量达到600万次点击量、120万次页面刷新、12万个在线用户时，带宽是多少？
 - $159\text{GB} * 3 = 477\text{GB} > 450\text{GB}$
 - 带宽会成为瓶颈

通过系统日志进行分析

■ 通过上述日志中数据分析，可以进行场景设计

— 目标场景：预估系统扩张规模，预留20%的增长空间

- 1小时内，系统承受5万个用户在线，页面刷新48万次，点击量312万次，带宽吞吐量191GB，在该负载下CPU及内存资源利用率低于80%，无明显磁盘网络瓶颈

通过系统日志进行分析

■ 通过上述日志中数据分析，可以进行场景设计

— 手工场景 (Real-World Scenario)



通过系统日志进行分析

■ 通过上述日志中数据分析，可以进行场景设计

— 手工场景（Basic Scenario）（留20%的上升空间）

- 设置用户最大访问量为5万用户，根据实际业务，设置负载生成和负载取消的速度，查看系统在逐渐增加到5万用户的过程中，系统各资源是否逐渐走向瓶颈，在峰值负载下，得到峰值性能数据，并验证能否满足性能需求

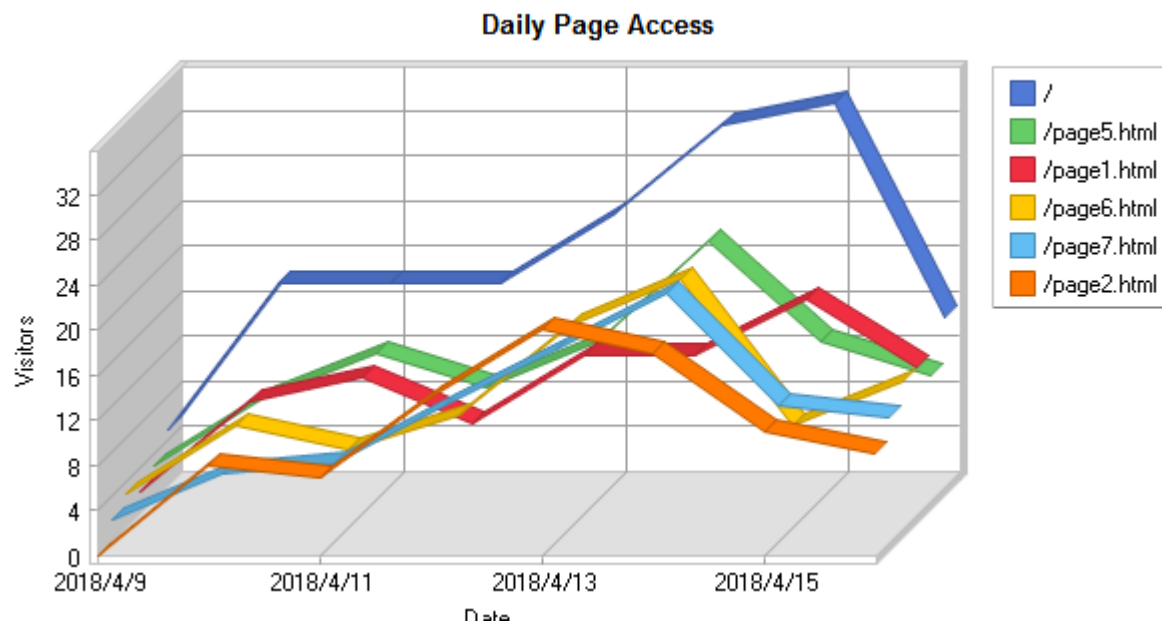
通过系统日志进行分析

- 分析日志除了能得到目标场景和手工场景的信息外，还能获得什么信息？
 - 对于性能测试来说，其实并不需要整个系统的所有功能都达到某个性能
 - 因为有些功能用的多，有些功能用的少
 - 用的多的，性能要求高；用的少的，性能要求低
 - 怎样知道哪些功能和页面被经常访问？

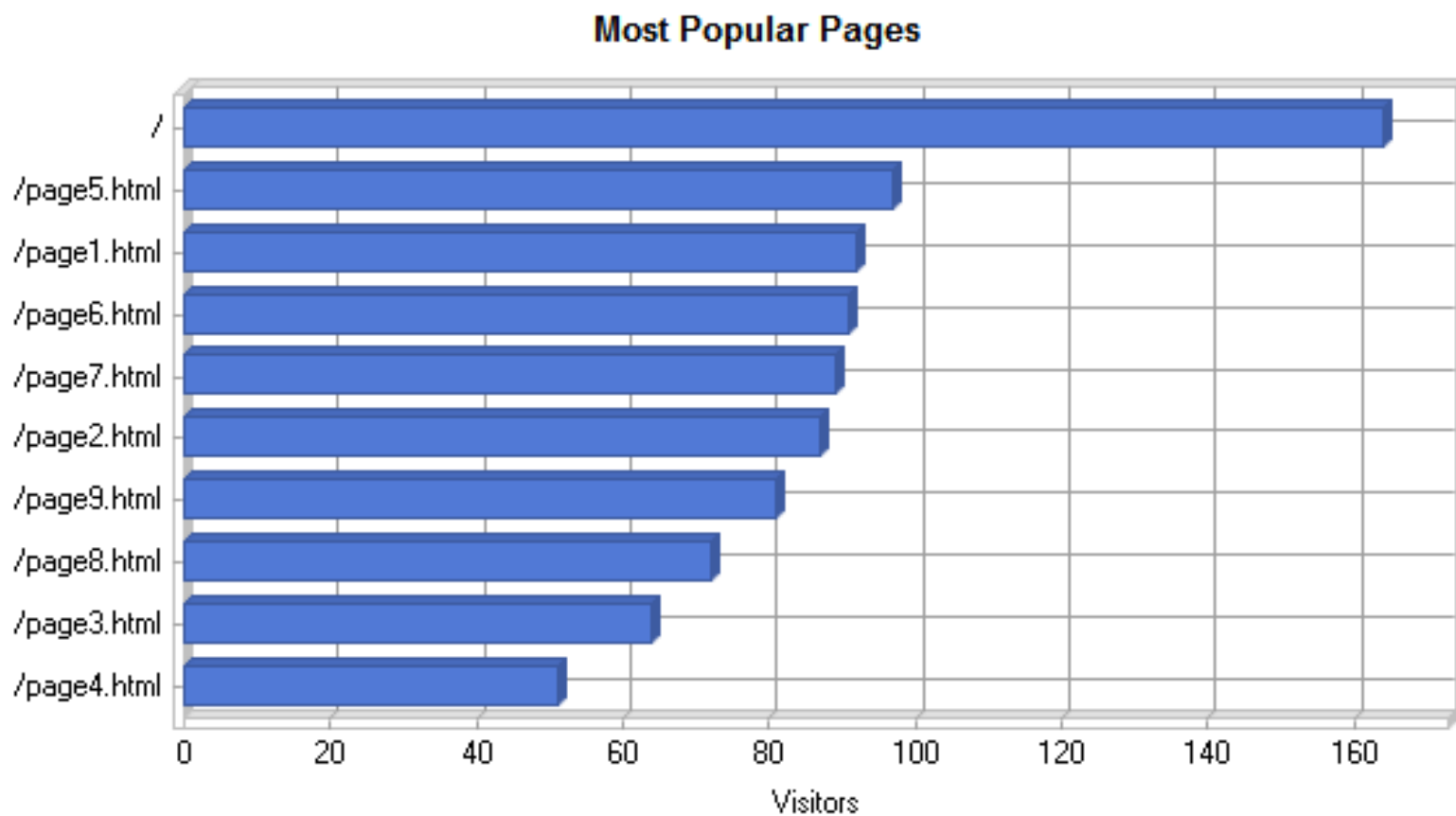
通过系统日志进行分析

■ 打开日志Access Statistics 功能，即可得到

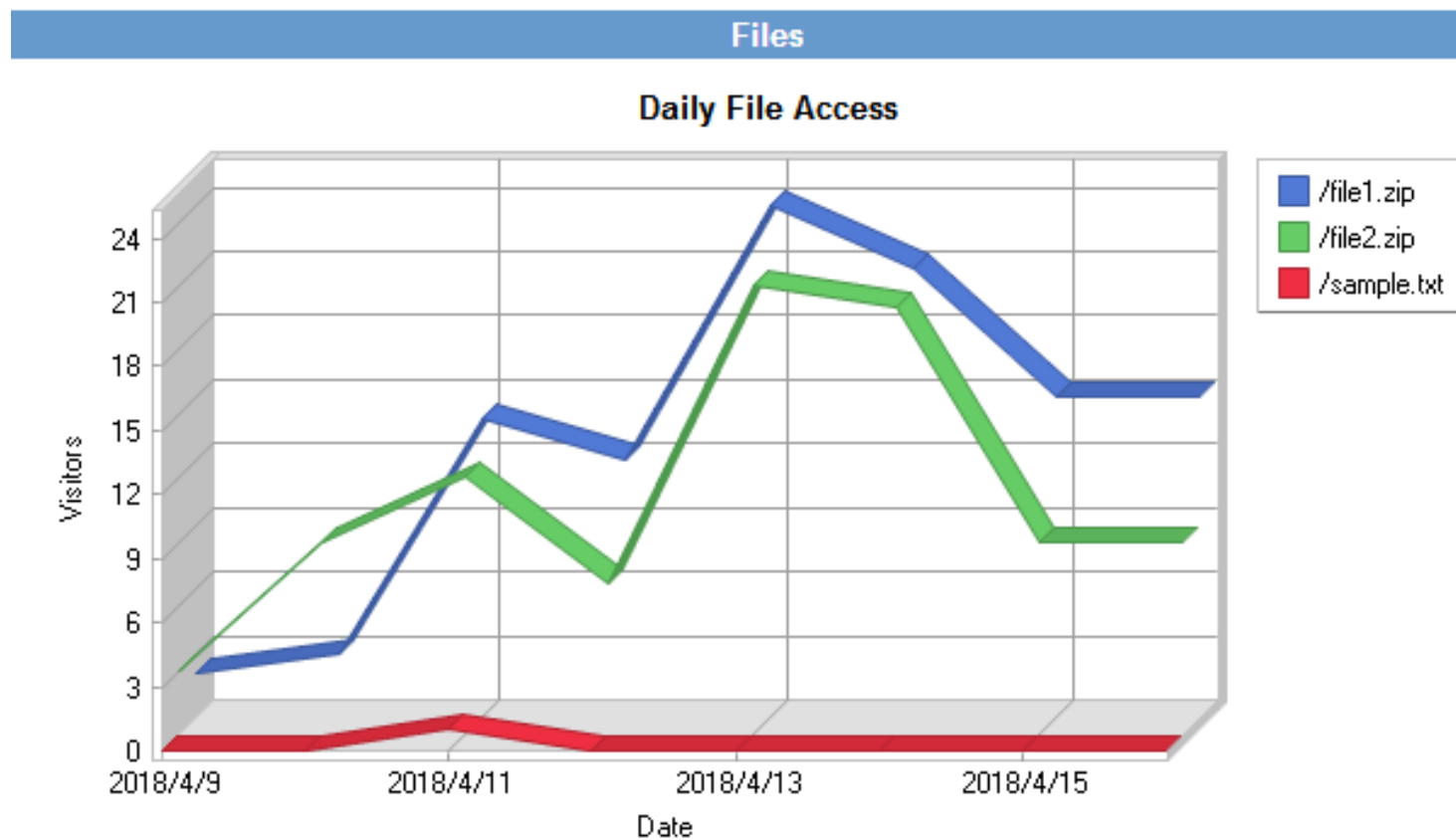
- 整个系统用户访问最多的页面分析
- 系统中文件、图片、进入系统页面、用户在系统中的操作路径等信息



通过系统日志进行分析



通过系统日志进行分析



通过系统日志进行分析

■ 补充：

— 对于服务器部署地，也需要考虑

- 同城部署，速度快
- 异地和异国都需要考虑网速问题
- 分析主要客户群体访问速度，避免出现主要客户群体访问慢的问题

通过系统日志进行分析

- 衡量性能测试的有效性和真实性也可以通过日志来实现
 - 如果性能测试日志与历史日志数据接近，说明性能测试几乎完全模拟了用户的行为，性能负载生成是成功的

内容总结

- 熟悉业务流程
- 定义测试目标
 - 通过用户数据进行分析
 - 通过系统日志进行分析



Question
