

# RWP团队谈性能优化之大开眼界篇

## (一) 连接池策略

2020年6月5日上午11:00

董志平 (Cary), 曲卓 (Christine)

公益讲座11点准时开始, 请大家先浏览云技术微信公众号技术文章  
资料会在各群同步发布, 已入群客户请勿重复入群!

扫码加入:  
19c新特性讲座群



欢迎关注:  
甲骨文云技术公众号



ORACLE

# 连接池策略



董志平 (Cary)

曲阜 (Christine)

Real-World Performance Team

Oracle Database Development

## 我们是谁

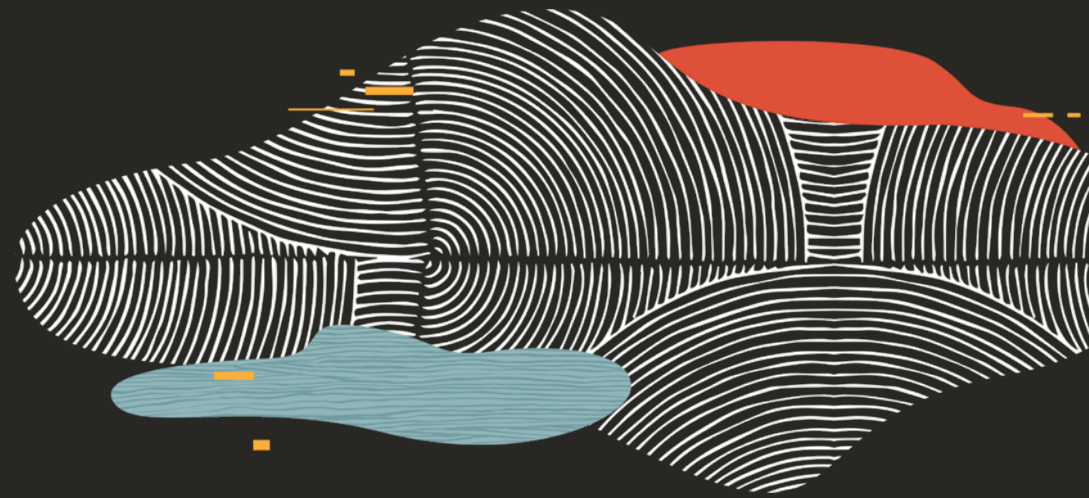
- Oracle数据库研发团队的一部分
- 成员分布在美国，欧洲，亚洲
- 加在一起几百年的经验

## 我们是怎样工作的

- 按照产品设计的方法去使用
- 全局视角看问题
- 目标是最好的性能
- 以数据为驱动的分析
- 分享我们所学

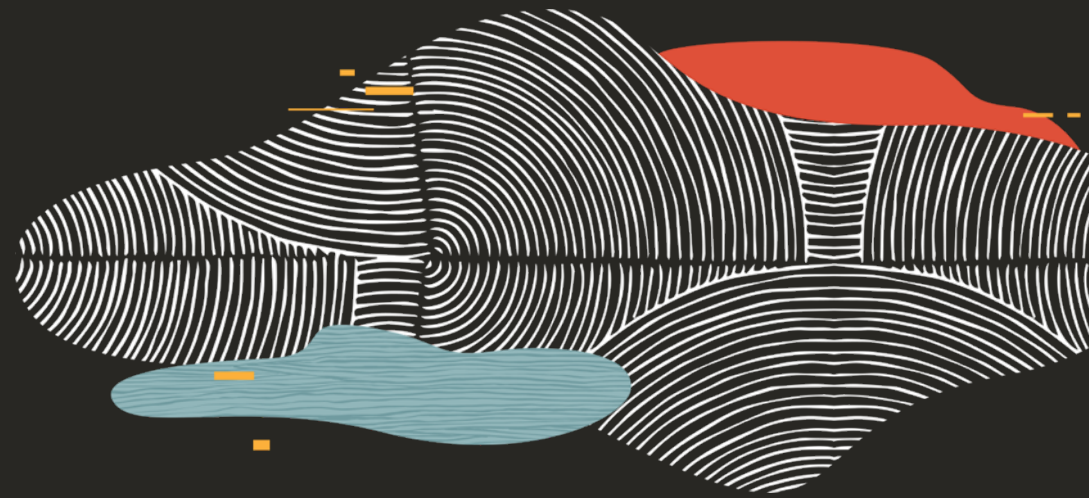
# 议程

- 1 连接池demo
- 2 连接池策略分析
- 3 总结

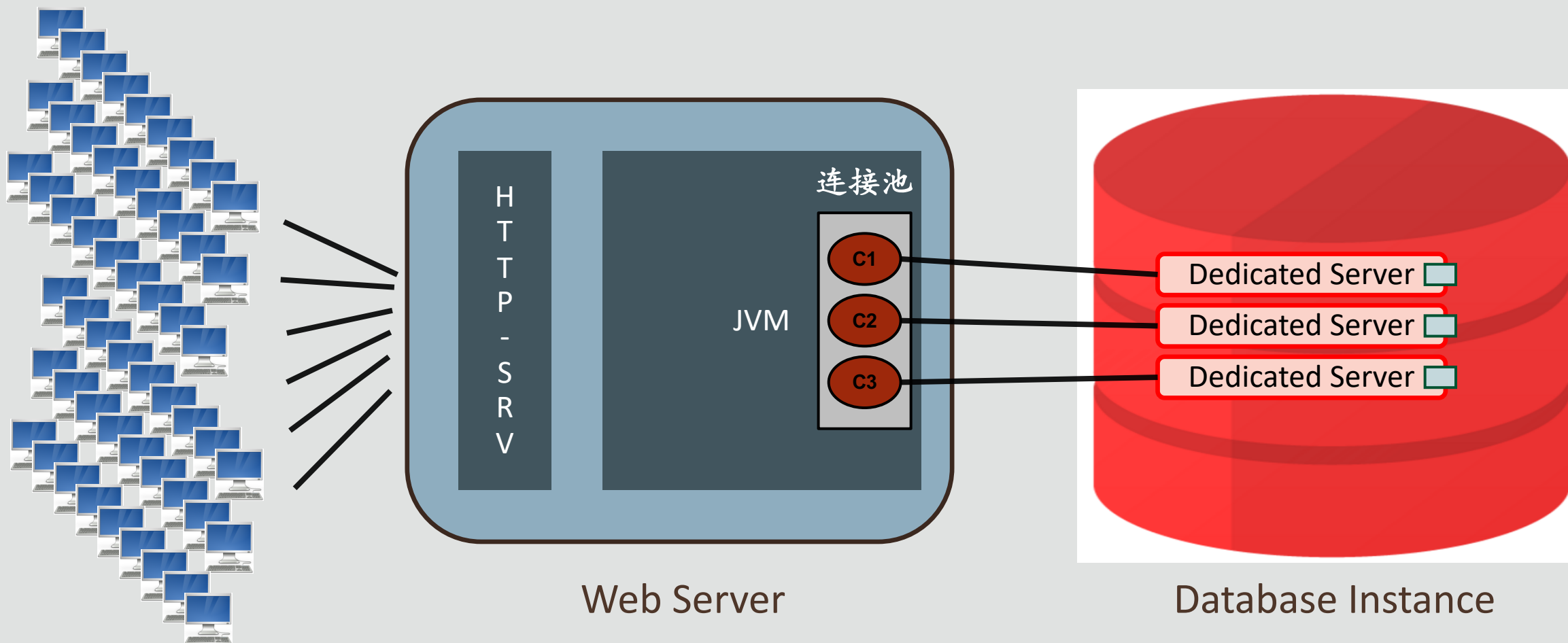


# 议程

- 1 连接池demo
- 2 连接池策略分析
- 3 总结



## Web应用的常见架构



- 很多OLTP系统出现问题，找到RWP团队，根本原因是\*\*不好的连接池策略
- 不好的连接池策略，通常表现为：

大量的（几千个）连接到数据库上

动态连接池导致的大量的logon/logoff数据库 (>1/Sec)

性能时好时坏，时而可接受，时而非常差

无法实时确定系统的性能表现

## Demo的硬件环境

---

- 数据库服务器：X8-2的1个计算节点，48 CPU core
- 应用服务器：与数据库服务器同一个X8-2的2个计算节点

1. 启动OLTP 工作负载, 设置UCP 连接池的最大值为6000, Think Time 10000 ms
2. 响应时间和TPS 都很稳定
3. 数据库服务器 CPU使用率很低, 不到10%
4. 数据库里的活跃会话数很低, 大概2-4



1. 将Think Time 设成 5000 ms, 使得工作负载翻倍
2. 响应时间不变, TPS翻倍, 系统线性扩展
3. 数据库服务器 CPU使用率依然很低, 不到10%
4. 数据库里的活跃会话数略有增加, 但总体依然很低, 大概不超过6



1. 将Think Time设成 2500 ms, 使得工作负载翻倍
2. 响应时间发生了剧烈的抖动, TPS也变得不稳定, 但基本翻倍, 基本线性增长了
3. 数据库服务器CPU使用率依然很低, 不到20%
4. 第一次连接风暴: 在短短1分钟左右的时间里, 数据库里的会话数从200多增加到5000多, 活跃会话数峰值稳定在1000左右。活跃下来之后, 会话数降低, 响应时间和TPS也再次趋于稳定



1. 将Think Time设定成 2500 ms, 使得工作负载翻倍
2. 响应时间发生了剧烈的抖动, TPS也变得不稳定, 但基本翻倍, 基本线性增长了
3. 数据库服务器CPU使用率依然很低, 不到20%
4. 第一次连接风暴: 在短短1分钟左右的时间里, 数据库里的会话数从200多增加到5000多, 活跃会话数峰值稳定在1000左右。活跃下来之后, 会话数降低, 响应时间和TPS也再次趋于稳定



1. 将Think Time设成 2500 ms, 使得工作负载翻倍
2. 响应时间发生了剧烈的抖动, TPS也变得不稳定, 但基本翻倍, 基本线性增长了
3. 数据库服务器CPU使用率依然很低, 不到20%
4. 第一次连接风暴: 在短短1分钟左右的时间里, 数据库里的会话数从200多增加到了5000多, 活跃会话数峰值稳定在1000左右。活跃下来之后, 会话数降低, 响应时间和TPS也再次趋于稳定



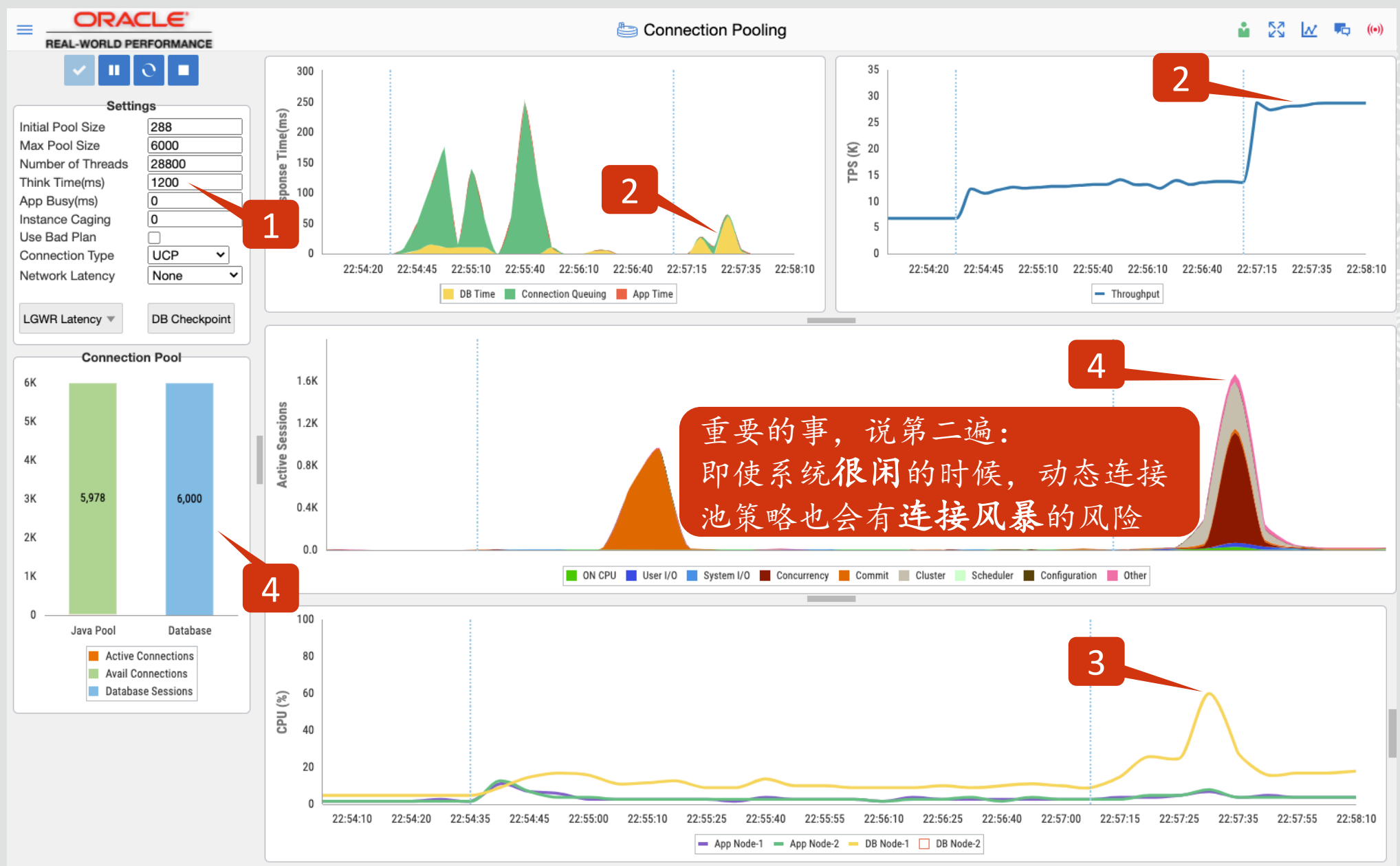
即使系统很闲的时候, 动态连接池策略也会有连接风暴的风险

动态连接池, 是指连接池的最大值不等于最小值。当最大值与最小值相差很大, 比如相差几千的时候, 连接风暴的风险尤其大

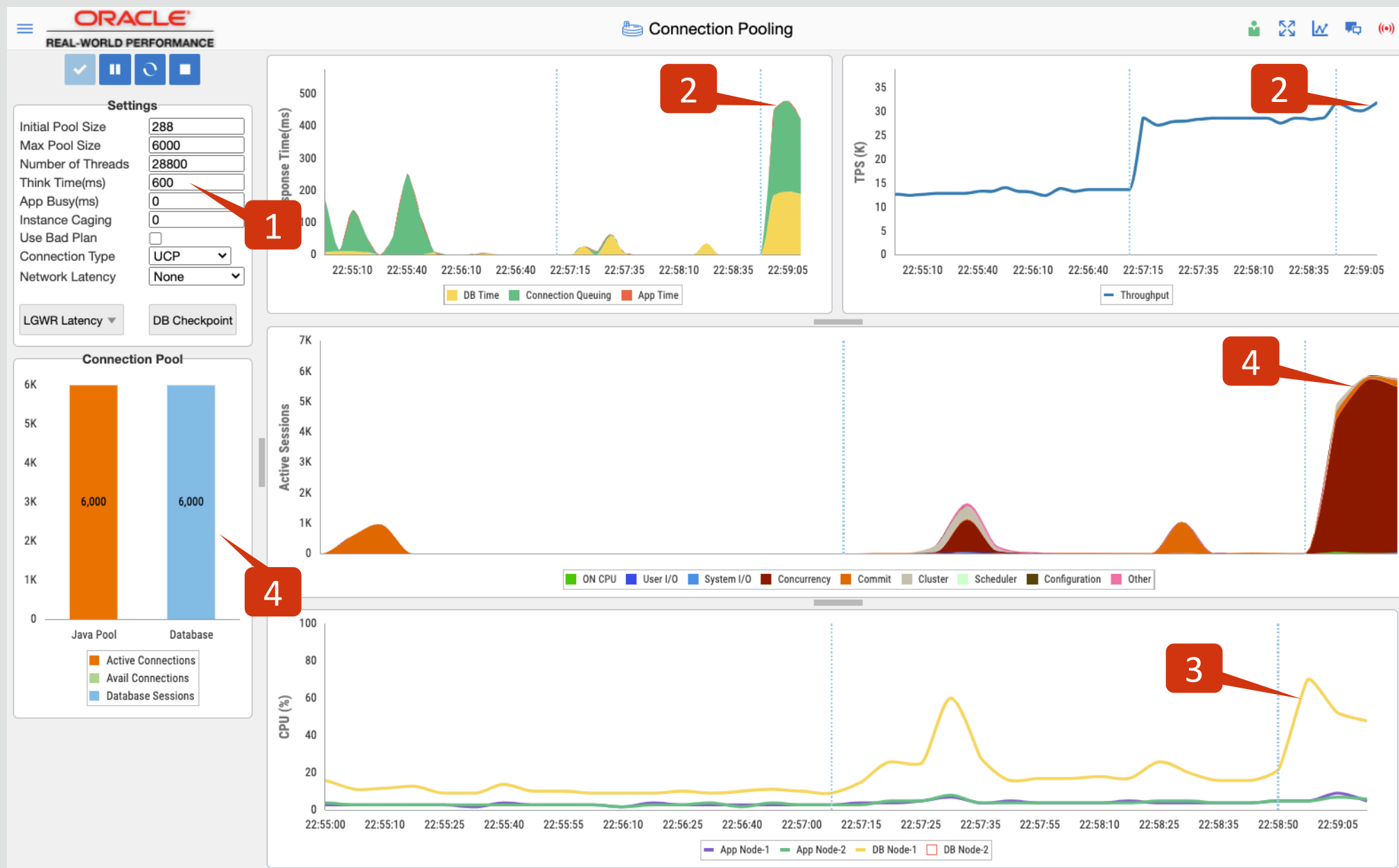
1. 将Think Time设成1200ms, 使得 workload 翻倍
2. 响应时间再次剧烈抖动, TPS也基本翻倍, 基本线性增长了
3. 数据库服务器CPU使用率不高, 峰值50%左右, 稳定下来之后20%左右
4. 第二次连接风暴: 在不到1分钟的时间里, 数据库的话数增加了5000多, 活跃会话数峰值1600左右。稳定下来之后, 活跃会话数降低, 响应时间和TPS也再次趋于稳定



1. 将Think Time设成1200 ms, 使得 workload 翻倍
2. 响应时间再次剧烈抖动, TPS也基本翻倍, 基本线性增长了
3. 数据库服务器CPU使用率不高, 峰值50%左右, 稳定下来之后20%左右
4. 第二次连接风暴: 在不到1分钟的时间里, 数据库的话数增加了5000多, 活跃会话数峰值达到6000多, 稳定下来之后, 活跃会话数降低, 响应时间和TPS也再次趋于稳定



1. 将Think Time设成 600 ms, 使得工作负载翻倍
2. 响应时间, 剧烈抖动, 比之前的峰值还高很多, TPS基本没有增长, 也不稳定
3. 数据库服务器 CPU使用率增高, 峰值60%左右
4. 数据库里的会话数已经达到连接池设置的最大值, 6000, 所以不会再有连接风暴了。活跃会话数达到6000, 数据库里面出现大量等待事件



1. 将Think Time设成 600 ms, 使得工作负载翻倍
2. 响应时间, 剧烈抖动, 比之前的峰值还高很多, TPS基本没有增长, 也不稳定
3. 数据库服务器 CPU使用率增高, 峰值60%左右
4. 数据库里的会话数已经达到连接池设置的最大值, 6000, 所以不会再有连接风暴了。活跃会话数达到6000, 数据库里面出现大量等待事件



1. 将Think Time 设成 600 ms, 使得工作负载翻倍
2. 响应时间, 剧烈抖动, 比之前的峰值还高很多, TPS基本没有增长, 也不稳定
3. 数据库服务器 CPU使用率增高, 峰值60%左右, 稳定下来之后 30%多
4. 数据库里的会话数已经达到连接池设置的最大值, 6000, 所以不会再有连接风暴了。活跃会话数持续维持在6000左右, 数据库里面持续有大量等待事件



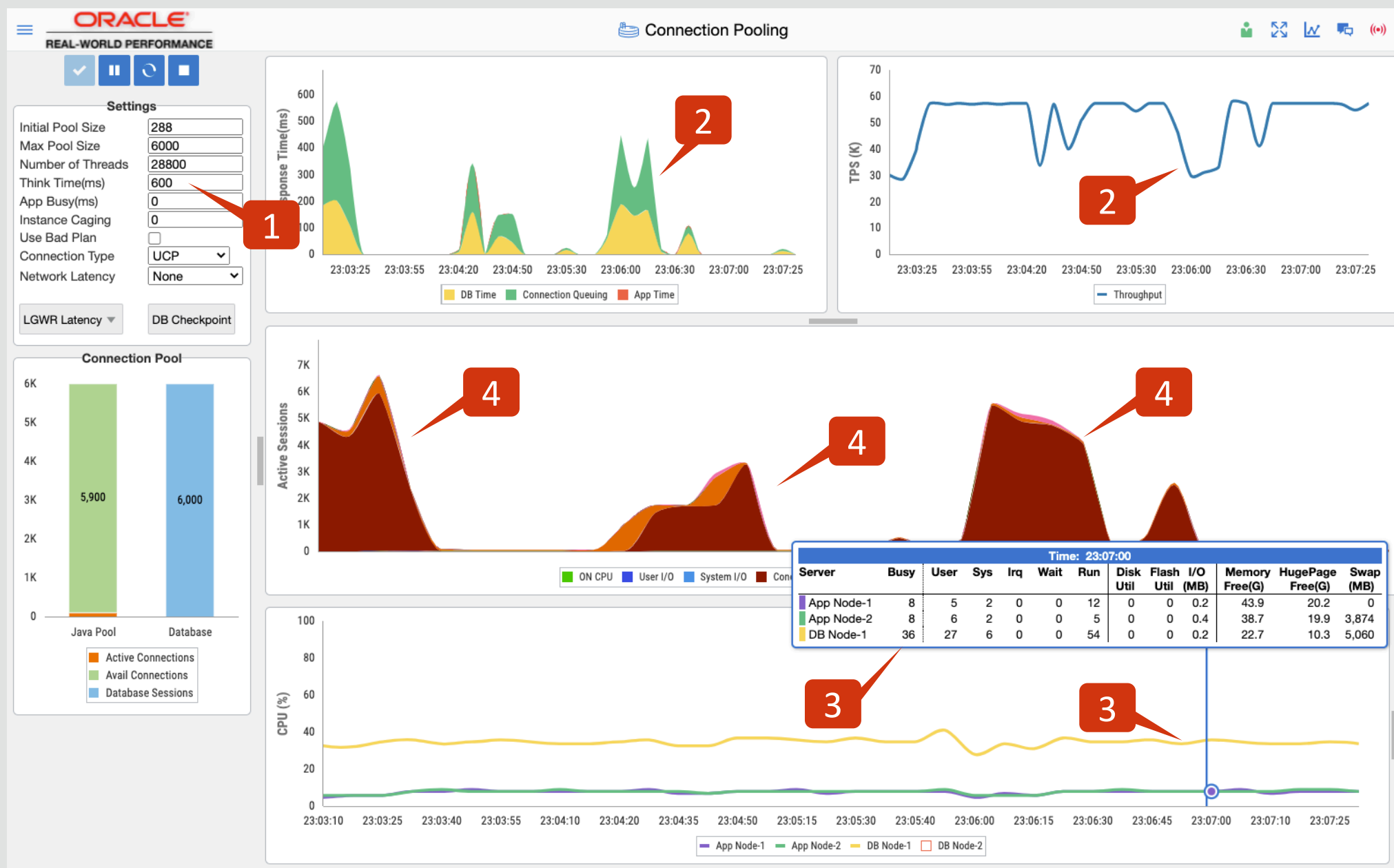
1. Think Time 设成 600 ms, 持续跑一段时间
2. 响应时间, 和 TPS 都非常不稳定
3. 数据库服务器 CPU 使用率大概 30% 多
4. 数据库里的活跃会话数也非常不稳定, 活跃会话数高的时候, 会伴随数据库里面有大量等待事件



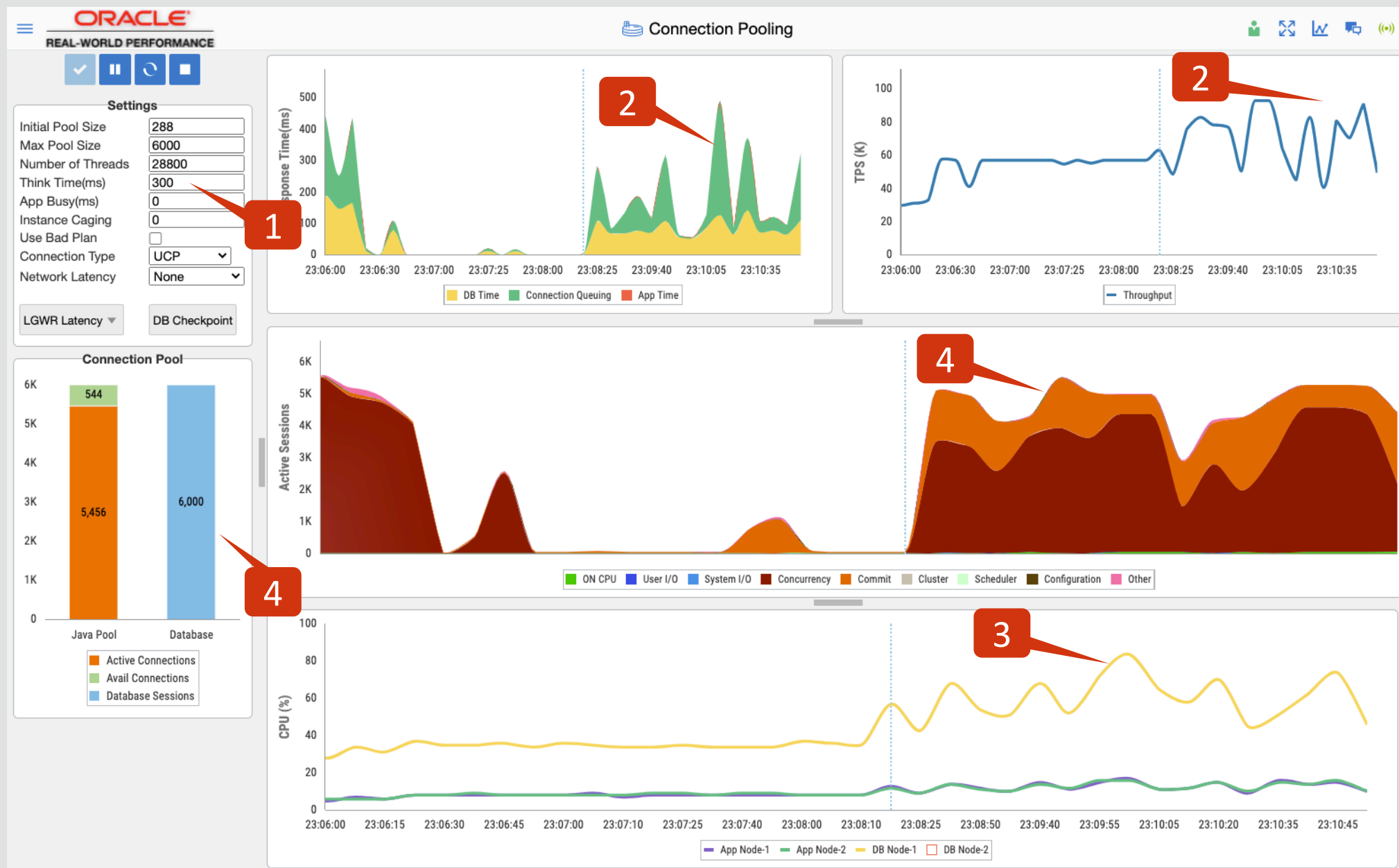
1. Think Time 设成 600 ms, 持续跑一段时间
2. 响应时间, 和 TPS 都非常不稳定
3. 数据库服务器 CPU 使用率大概 30% 多
4. 数据库里的活跃会话数也非常不稳定, 活跃会话数高的时候, 会伴随数据库里面有大量等待事件



1. Think Time 设成 600 ms, 持续跑一段时间
2. 响应时间, 和 TPS 都非常不稳定
3. 数据库服务器 CPU 使用率大概 30% 多
4. 数据库里的活跃会话数也非常不稳定, 活跃会话数高的时候, 会伴随数据库里面有大量等待事件



1. 将Think Time 设成 300 ms, 使得工作负载翻倍
2. 响应时间再次剧烈抖动, TPS 没有线性增长, 且非常不稳定
3. 数据库服务器 CPU使用率非常不稳定, 大概 40-80%, 峰值 80%多
4. 数据库里的活跃会话数居高不下, 大多在 5000左右, 持续有大量等待事件



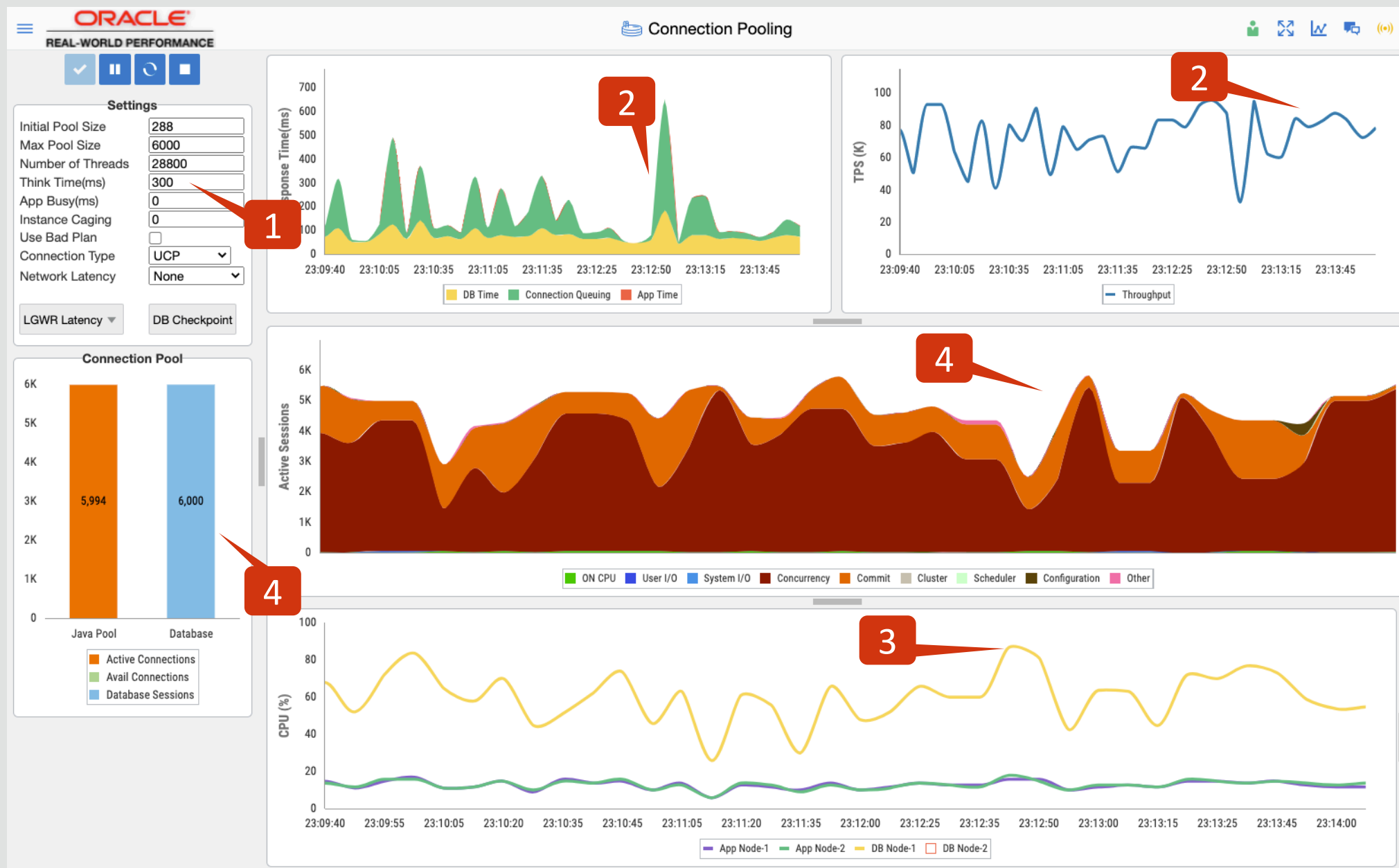
1. 将Think Time 设成 300 ms, 使得工作负载翻倍
2. 响应时间再次剧烈抖动, TPS 没有线性增长, 且非常不稳定
3. 数据库服务器 CPU使用率非常不稳定, 大概 40-80%, 峰值 80%多
4. 数据库里的活跃会话数居高不下, 大多在 5000左右, 持续有大量等待事件



1. 将Think Time 设成 300 ms, 使得工作负载翻倍
2. 响应时间再次剧烈抖动, TPS 没有线性增长, 且非常不稳定
3. 数据库服务器 CPU使用率非常不稳定, 大概 40-80%, 峰值 80%多
4. 数据库里的活跃会话数居高不下, 大多在 5000 左右, 持续有大量等待事件



1. Think Time 设成 300 ms, 持续跑一段时间
2. 响应时间, 和 TPS 都非常不稳定
3. 数据库服务器 CPU 使用率也非常不稳定, 大概 40-80%, 峰值 80% 多
4. 数据库里的活跃会话数居高不下, 大多在 5000 左右, 持续有大量等待事件



1. Think Time 设成 300 ms, 持续跑一段时间
2. 响应时间, 和 TPS 都非常不稳定
3. 数据库服务器 CPU 使用率也非常不稳定, 大概 40-80%, 峰值 80% 多
4. 数据库里的活跃会话数居高不下, 大多在 5000 左右, 持续有大量等待事件



客户抱怨响应时间太慢了  
领导批评TPS太低赚钱太少了  
同事说都赖数据库里大量的等待事件

—  
怎样降低响应时间？

怎样提高TPS？

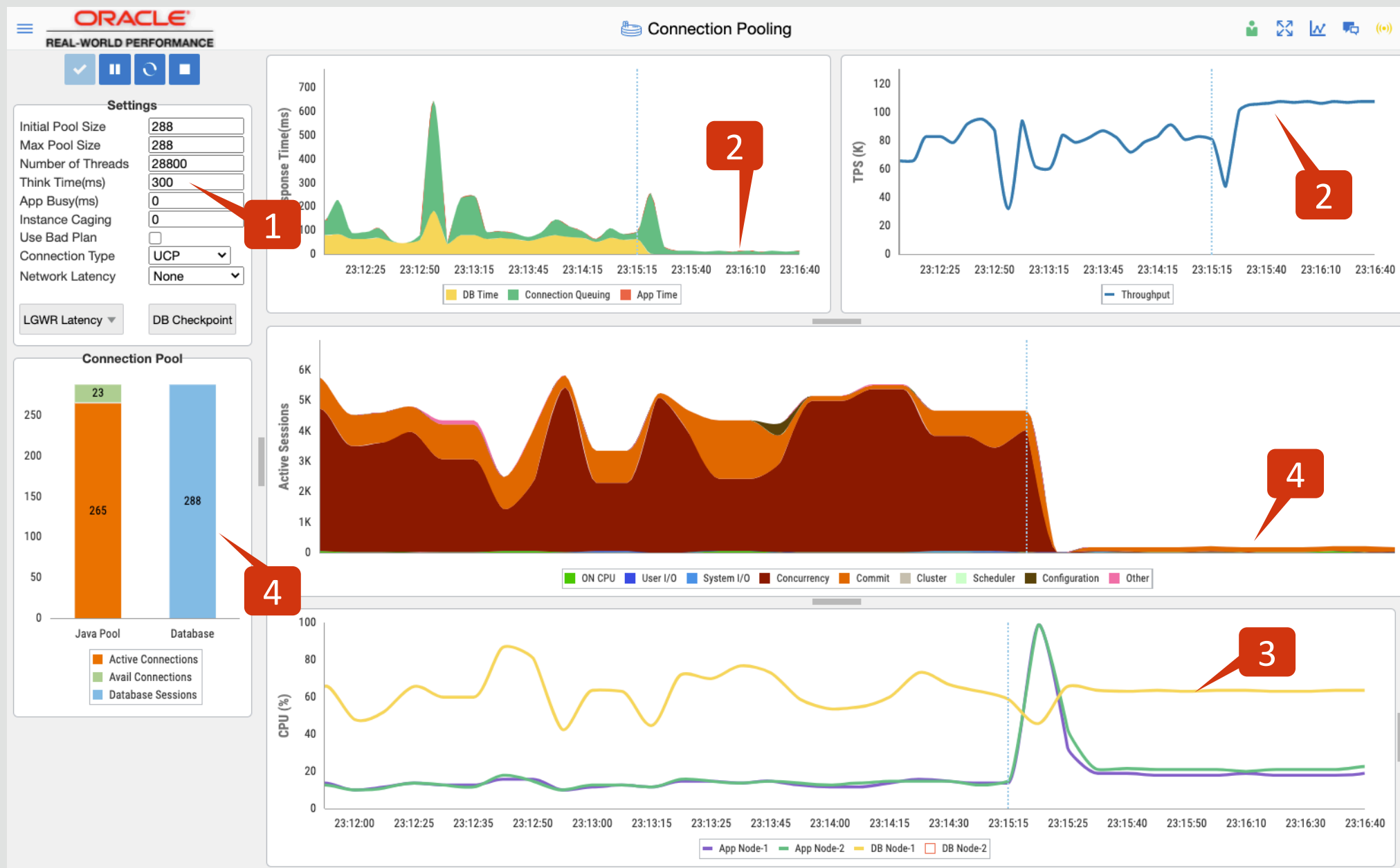
怎样消除数据库里大量的等待事件？

# 用自上而下的方法去解决问题

—  
仅做一个改变：

减少到数据库服务器的连接数，6000 → 288

1. 将Max Pool Size设成288, 即允许最多288个连接连到数据库上
2. 响应时间降低了, TPS升高了, 响应时间和TPS都变的非常稳定
3. 数据库服务器CPU使用率也非常稳定, 大概在60%多, 整体比之前的使用率低
4. 数据库里的活跃会话数大幅降低, 等待事件主要是log file sync和ON CPU



1. 将Max Pool Size设成288, 即允许最多288个连接连到数据库上
2. 响应时间降低了, TPS升高了, 响应时间和TPS都变的非常稳定
3. 数据库服务器CPU使用率也非常稳定, 大概在60%多, 整体比之前的使用率低
4. 数据库里的活跃会话数大幅降低, 等待事件主要是log file sync和ON CPU



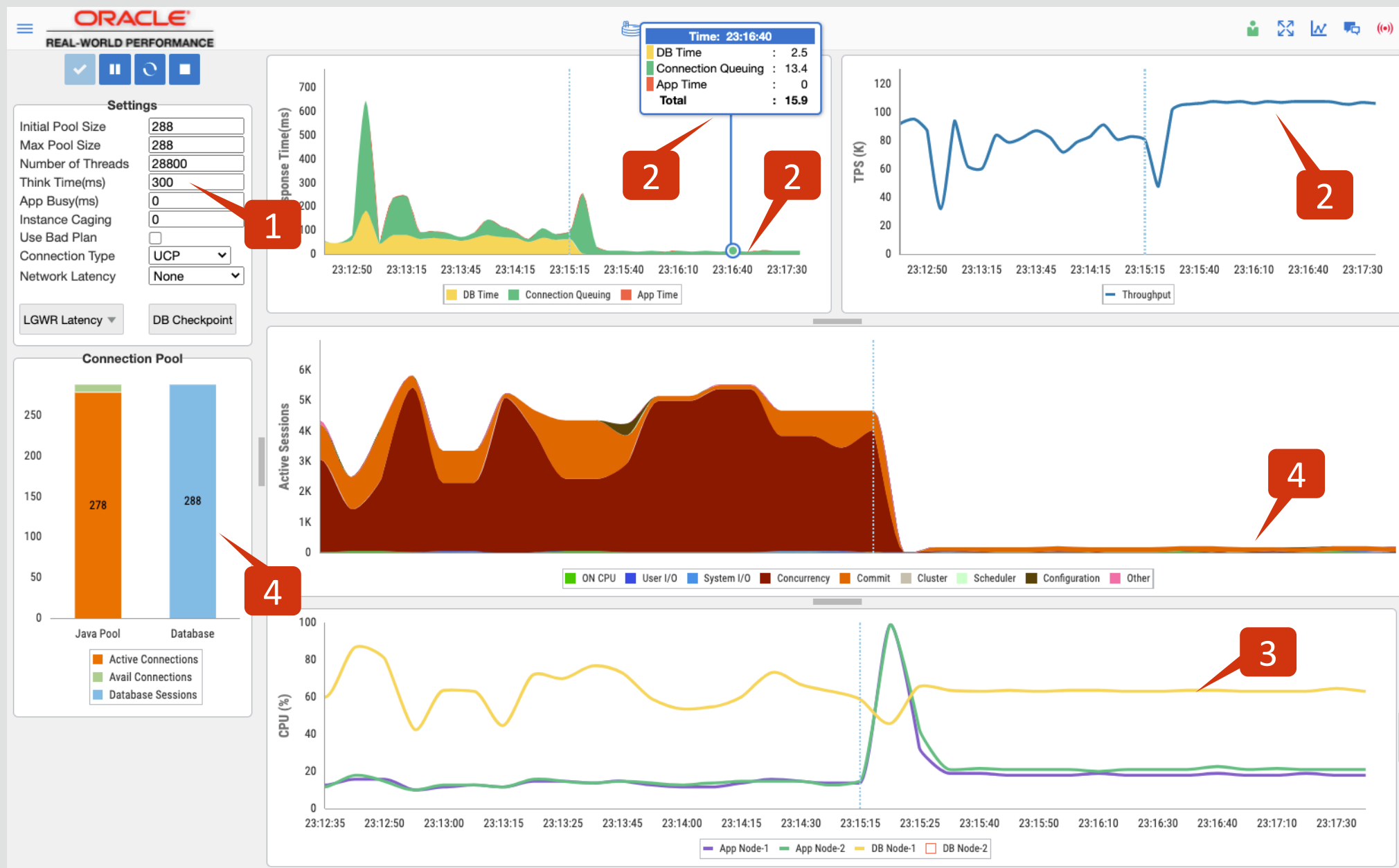
1. 将Max Pool Size设成288, 即允许最多288个连接连到数据库上
2. 响应时间降低了, TPS升高了, 响应时间和TPS都变的非常稳定
3. 数据库服务器CPU使用率也非常稳定, 大概在60%多, 整体比之前的使用率低
4. 数据库里的活跃会话数大幅降低, 等待事件主要是log file sync和ON CPU



1. 将Max Pool Size设成288, 即允许最多288个连接连到数据库上
2. 响应时间降低了, TPS升高了, 响应时间和TPS都变的非常稳定
3. 数据库服务器CPU使用率也非常稳定, 大概在60%多, 整体比之前的使用率低
4. 数据库里的活跃会话数大幅降低, 等待事件主要是log file sync和ON CPU



1. 将Max Pool Size设成288, 即允许最多288个连接连到数据库上
2. 响应时间降低了, TPS升高了, 响应时间和TPS都变的非常稳定
3. 数据库服务器CPU使用率也非常稳定, 大概在60%多, 整体比之前的使用率低
4. 数据库里的活跃会话数大幅降低, 等待事件主要是log file sync和ON CPU



1. Max Pool Size 设成288, 持续跑一段时间
2. 响应时间一直稳定保持在低位, TPS一直稳定保持在高位
3. 数据库服务器 CPU使用率非常稳定, 大概在64%
4. 数据库里的活跃会话数大概200, 等待事件主要是log file sync和ON CPU



此时将Think Time从300改为150，即  
工作负载再次翻倍，系统会怎么样？

会出现连接风暴吗？

会Crash吗？

数据库里会再次出现各种等待事件吗？

响应时间会增加吗？

TPS会下降吗？

此时将Think Time从300改为150，即工作负载再次翻倍，系统会怎么样？

- 数据库服务器上的CPU数量是没有变化的，就是运算能力没变化
- 连接池大小是没变化的，数据库只会给288个连接服务，服务的效率也不会变化，就是说，数据库处理每个用户请求的时间应该是没变化的
- Think Time=300的时候，响应时间里面就已经有了绿色，即排队等待连接的时间，就是说数据库已经到了最大处理能力了，那么Think Time=150的时候，就只意味着排队等待连接的时间更长而已，数据库服务器的CPU使用率应该也是没变化的
- 排队等待连接的时间更长，数据库处理每个用户请求的时间没变化，意味着最终的用户响应时间会变长

1. 将Think Time 设成 150 ms, 使得 workload 翻倍
2. 响应时间增加了, 但是都是增加在了排队等待连接的时间, TPS和之前基本一致
3. 数据库服务器 CPU使用率和之前基本一致, 大概在64%
4. 数据库里的活跃会话数和之前基本一致, 大概200, 等待事件主要是log file sync和ON CPU



1. 将Think Time 设成 150 ms, 使得 workload 翻倍
2. 响应时间增加了, 但是都是增加在了排队等待连接的时间, TPS和之前基本一致
3. 数据库服务器 CPU使用率和之前基本一致, 大概在64%
4. 数据库里的活跃会话数和之前基本一致, 大概200, 等待事件主要是log file sync和ON CPU



1. 将Think Time 设成 150 ms, 使得 workload 翻倍
2. 响应时间增加了, 但是都是增加在了排队等待连接的时间, TPS和之前基本一致
3. 数据库服务器 CPU使用率和之前基本一致, 大概在64%
4. 数据库里的活跃会话数和之前基本一致, 大概200, 等待事件主要是log file sync和ON CPU



合适的连接池策略, 能够确保系统的平稳运行  
 即使再减少Think Time, 增加 workload

- ✓ TPS也不会降低
- ✓ 数据库服务器CPU使用率也不会升高
- ✓ 数据库里也不会出现大量因竞争而导致的等待事件
- ✓ 只是响应时间会变的更长, 因为需要排更长的队等待数据库连接

# 如果此时对响应时间不满意，怎么办？

---

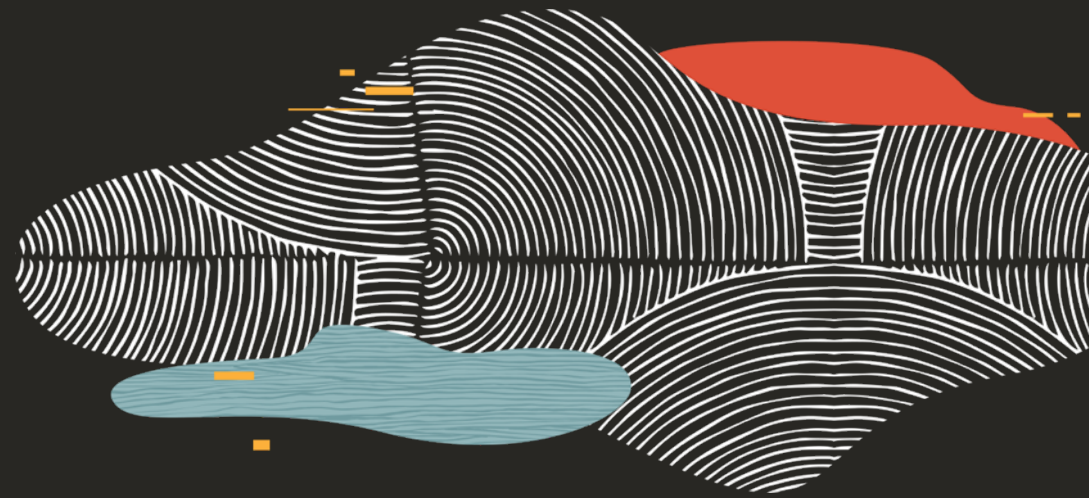
# 如果此时对响应时间不满意，怎么办？

有一个经验是，OLTP系统想要性能好，数据库服务器的CPU使用率最好不要超过60-65%

上限已至，该扩容啦！

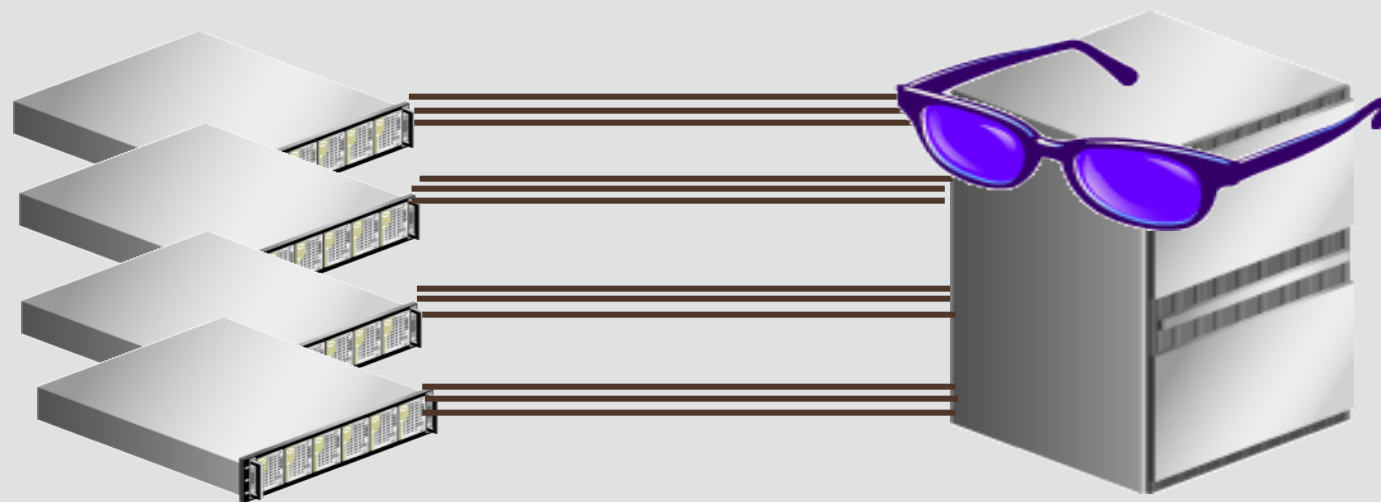
# 议程

- 1 连接池demo
- 2 连接池策略分析
- 3 总结

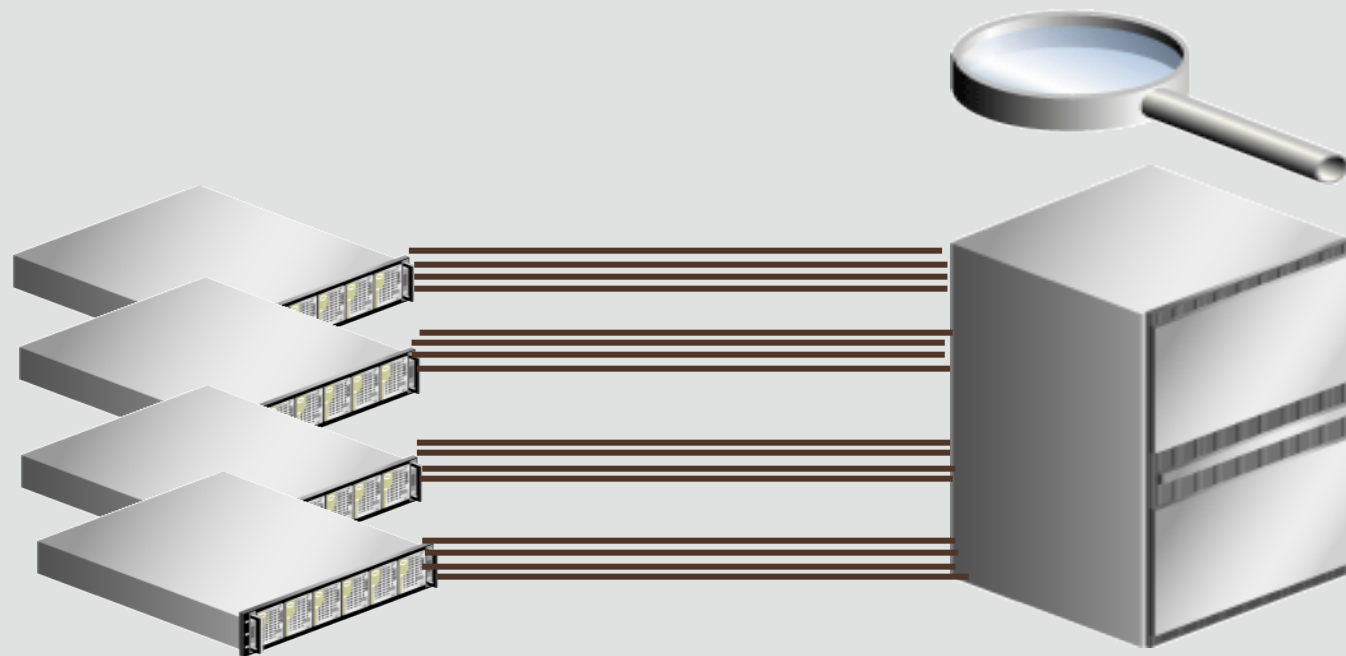
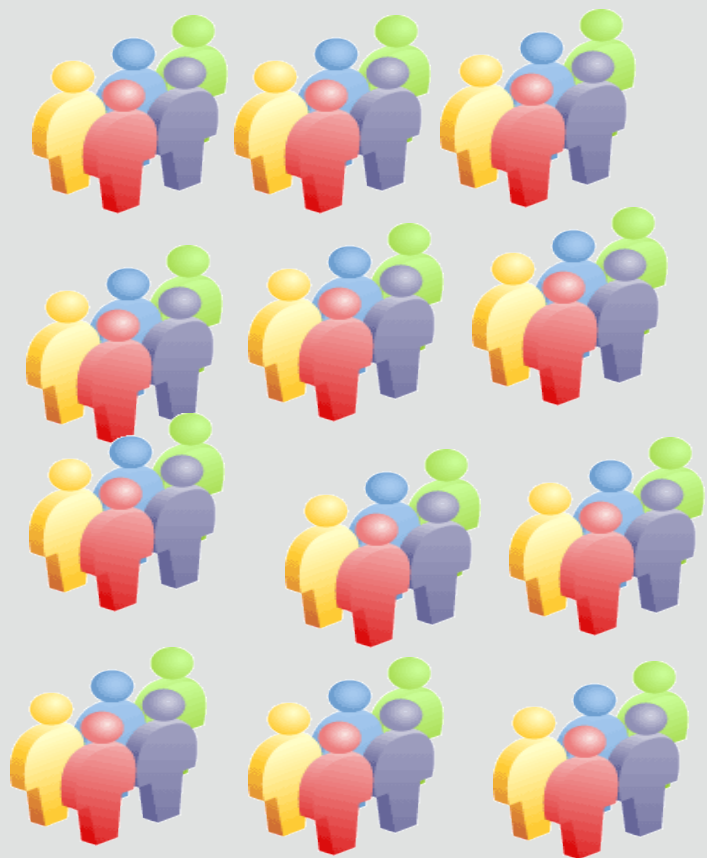


- 连接风暴是指，在短时间内，连接数成百上千的增长
- 连接风暴可能导致数据库服务器不稳定或者不可用
  - Logon/Logoff是非常消耗资源的（创建新连接，分配资源，map SGA等）
- 产生连接风暴的根本原因是动态连接池策略，即连接池的最小值和最大值不等，且差值很大（几千）
- 如果使用动态连接池策略，执行计划变差可能成为连接风暴的导火索，但大量的进程创建和logon活动可能掩盖导火索，从而导致诊断过程更长

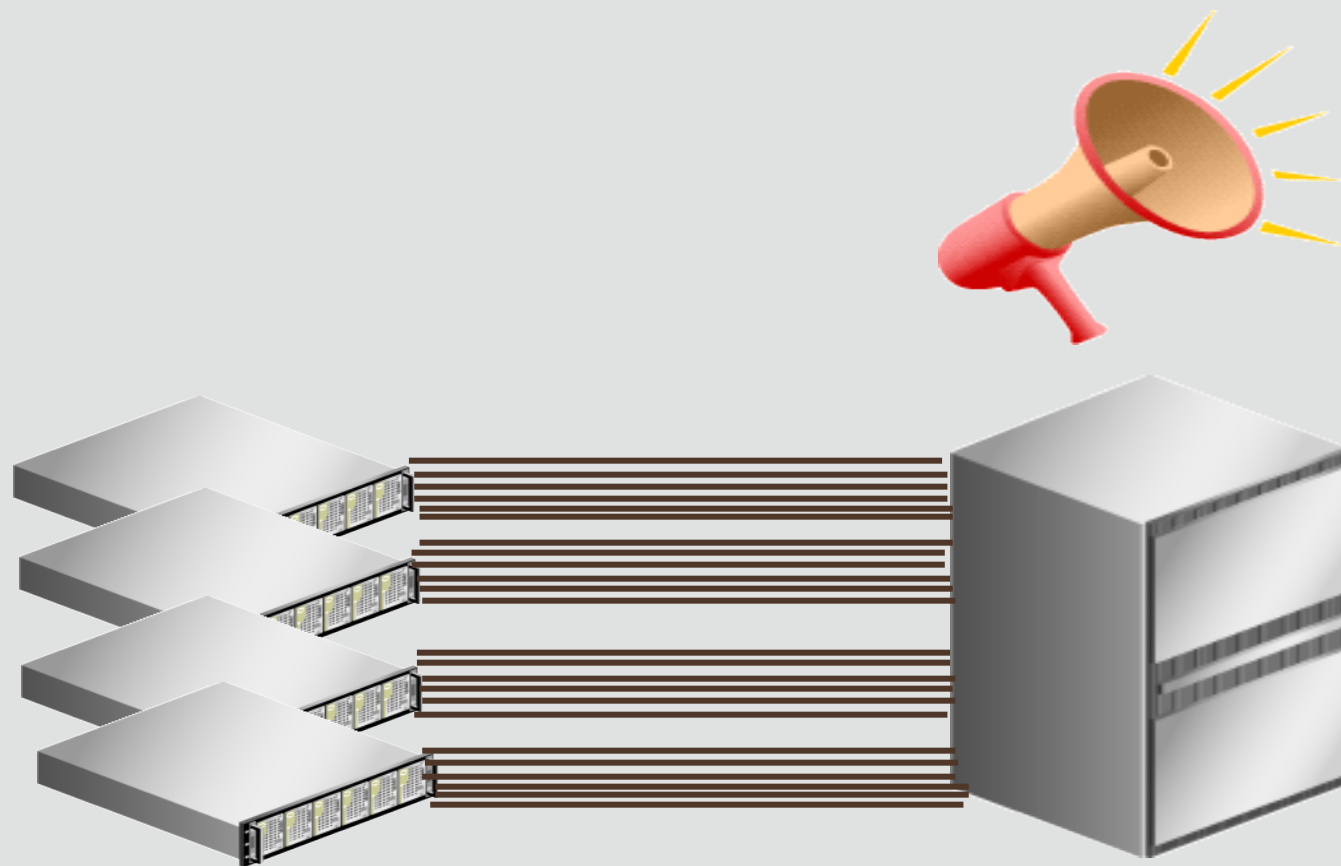
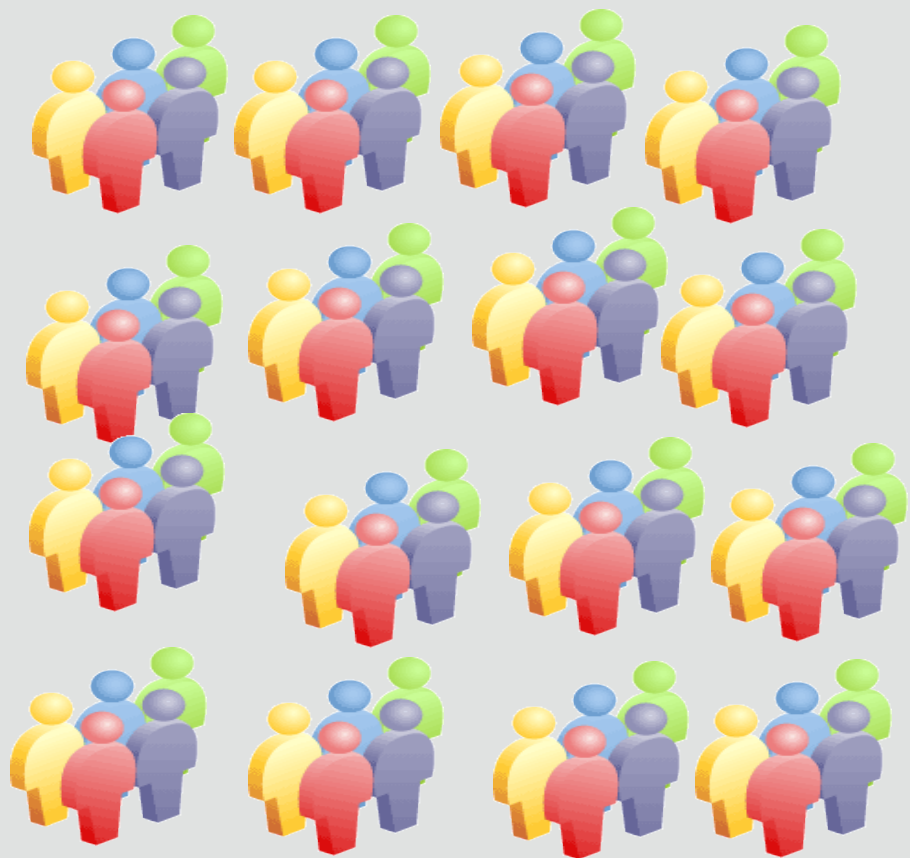
## 连接池策略分析 - 连接风暴示例



## 连接池策略分析 - 连接风暴示例



## 连接池策略分析 - 连接风暴示例



- Oracle数据库是process-based architecture

很多情况下，到数据库的连接会使用dedicated server方式，这意味着，对于每一个连接，在数据库服务器端有一个进程给这个连接服务

- 过多的连接，意味着过多的进程，但是并不意味着更高的TPS

Demo里面，使用288个连接的时候，与使用6000个连接相比，数据库服务器使用了更少的CPU，却使得响应时间变短了，TPS升高了

- CPU的使用效率才是关键

连接数过多会导致CPU cache的命中率下降，从而使得CPU使用效率下降

连接数过多时，latch的spin机制可能导致更多的空耗的CPU，从而使得CPU使用效率下降

## • 有一种消除大量等待事件的方法，叫 **减少连接数**

这是自上而下的分析问题的方法

关系型数据库都要能够做到ACID，为了ACID，数据库里面有各种串行机制，比如latch，lock，等等，连接数多了，不同的数据库会话争抢有限的资源，自然就会有大量的等待事件

所以连接数过多会导致数据库里面出现的大量等待事件，是表象

宏观思维，跳出等待事件的细节，从系统架构层面考虑，减少到数据库服务器的连接数，就能减少对有限资源的竞争，提高CPU的使用效率，就能消除那些大量的等待事件

## 连接池策略分析 - 连接数过多的问题

- 指的是连接池的最大值
- 连接池最大值设的很高，但是平时active连接数很少，这样有没有问题？

在连接池里的连接没有都active的时候，没有问题

但是连接数设大，就存在所有连接都active的可能，就存在连接数过多导致的问题的可能，也会有连接风暴的风险

如果有一次写日志操作很慢，或者有一条SQL的执行计划出了问题，都可能导致连接被占用的时间更长，从而导致连接风暴，大量的连接都变成active的

如果知道active连接数总是很少，连接池设小就行了，不需要设大，也不推荐设大

连接池策略分析 - Exadata最多能支持多少个连接?

## • Exadata最多能支持多少个连接?

这是个假问题。经常会有类似的问题出现

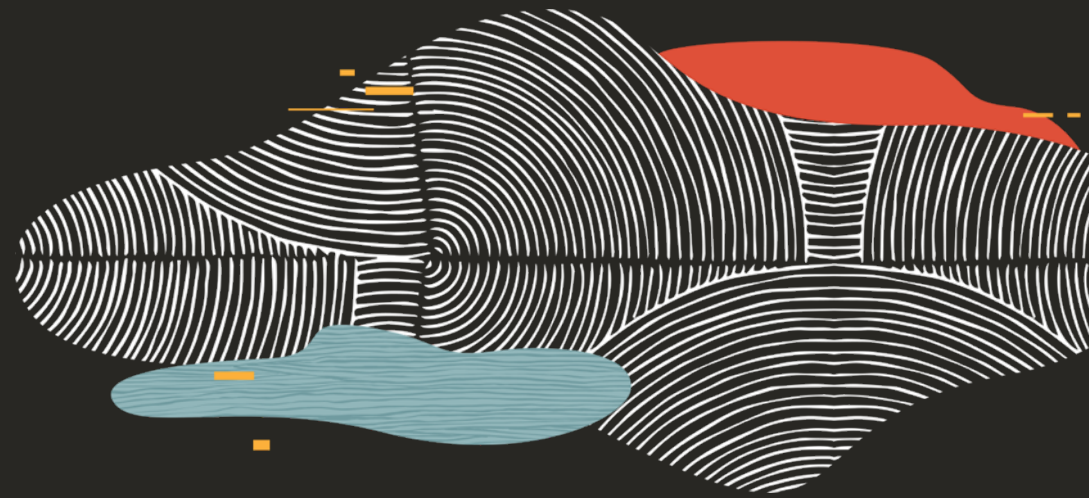
问这个问题的原因是什么呢？有一种声音：连接数越多 = 性能越好。有人说，想要性能好，就必须一请求连接到数据库，马上就能拿到连接。

而真正的问题是：系统跑在Exadata上，性能好不好？

所以更贴切的提问应该是：系统跑在Exadata上，事务的响应时间能在什么水平？TPS能达到多少？分析报表，批处理的业务多长时间能完成？

# 议程

- 1 连接池demo
- 2 连接池策略分析
- 3 总结



## 连接池策略常见的错误

### 1. 使用动态连接池

会有连接风暴的风险

### 2. 参照应用服务器上的排队时间来决定连接池大小

数据库服务器变忙的时候，应用服务器上的排队时间变长是正常的

### 3. 只参照一个应用服务器来决定连接池大小

决定连接池大小的时候，所有的应用服务器都应该考虑进去

### 4. 参照init.ora里面的CPU\_COUNT来决定连接池大小

超线程？虚拟机？CPU\_COUNT这个参数并不可靠！

## 总结

1. 建议，对于OLTP类型的系统，所有应用到数据库服务器的  
**最大连接数 = 最小连接数 = 数据库服务器CPU core\*数的1-10倍\*\***

\*注释1：CPU core，指的是物理的core数，而不是超线程之后的。

\*\*注释2：1-10倍只是一个参考值，具体适合的数值与具体的工作负载有关，需要测试才能确定。如果工作负载做的I/O相对多一些，则可能更接近于10倍是适合的数值，如果工作负载做运算耗CPU相对更多一些，则可能更接近于1倍是适合的数值。

2. 如果连接池策略合适，系统其他方面也已经很优化了，SLA还是不能满足要求，怎么办？ 扩容

# RWP团队谈性能优化之大开眼界篇 后续内容

- |                            |                   |
|----------------------------|-------------------|
| (二) 应用程序算法                 | 2020年6月12日上午11:00 |
| (三) 资源管理                   | 2020年6月19日上午11:00 |
| (四) AWR实例分析                | 2020年7月 3日上午11:00 |
| (五) SQL Monitor Report实例分析 | 2020年7月10日上午11:00 |

想RWP团队分析你的系统的AWR报告吗？请在2020年6月24日之前将你的系统的AWR报告及问题描述发送至 [rwp\\_training\\_admins@beehiveonline.oracle.com](mailto:rwp_training_admins@beehiveonline.oracle.com)

想RWP团队分析你的SQL Monitor Report吗？请在2020年7月1日之前将你的系统的SQL Monitor Report及问题描述发送至 [rwp\\_training\\_admins@beehiveonline.oracle.com](mailto:rwp_training_admins@beehiveonline.oracle.com)

资料会在各群同步发布，已入群客户请勿重复入群！

扫码加入：  
19c新特性讲座群



欢迎关注：  
甲骨文云技术公众号

