

2010 Esri 中国区域用户大会



# ArcGIS Server 部署策略 与性能优化

吴泳锋

GIS-让人类认知世界



# ArcGIS Server 系统中的性能影响因素



- 硬件
- 数据源
- ArcGIS Server 配置
- 服务
- 开发框架

# 硬件的影响

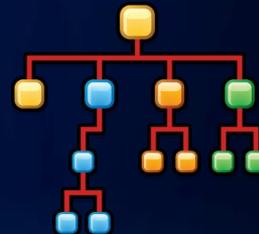
## ● 影响响应速度

- CPU性能
- 存储系统读写速度
- 网络延迟



## ● 影响负载能力

- CPU (核) 数量
- 存储系统读写速度
- 网络带宽



- 服务器集群规模



# 数据源的影响

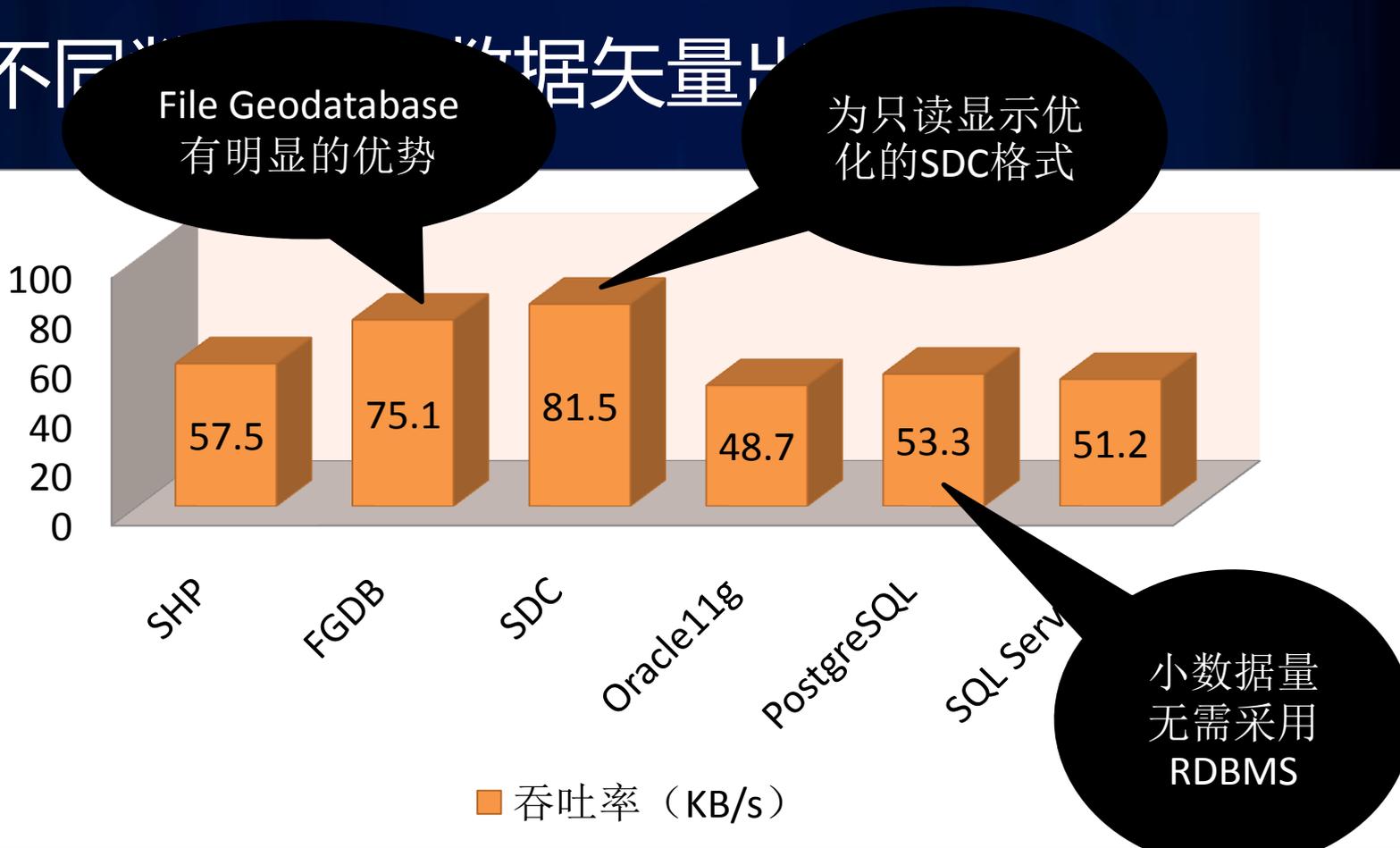
## ● 矢量数据

- 数据来源：ArcSDE, FileGDB, 矢量数据文件等
- 存储类型：ST\_Geometry, Oracle SDO\_Geometry, SQL Server Geometry, PG Geometry

## ● 栅格数据

- 数据来源：ArcSDE, FileGDB, Image Server, 栅格数据文件等
- 压缩：NONE, LZ77, JPEG

# 不同数据格式的数据吞吐量

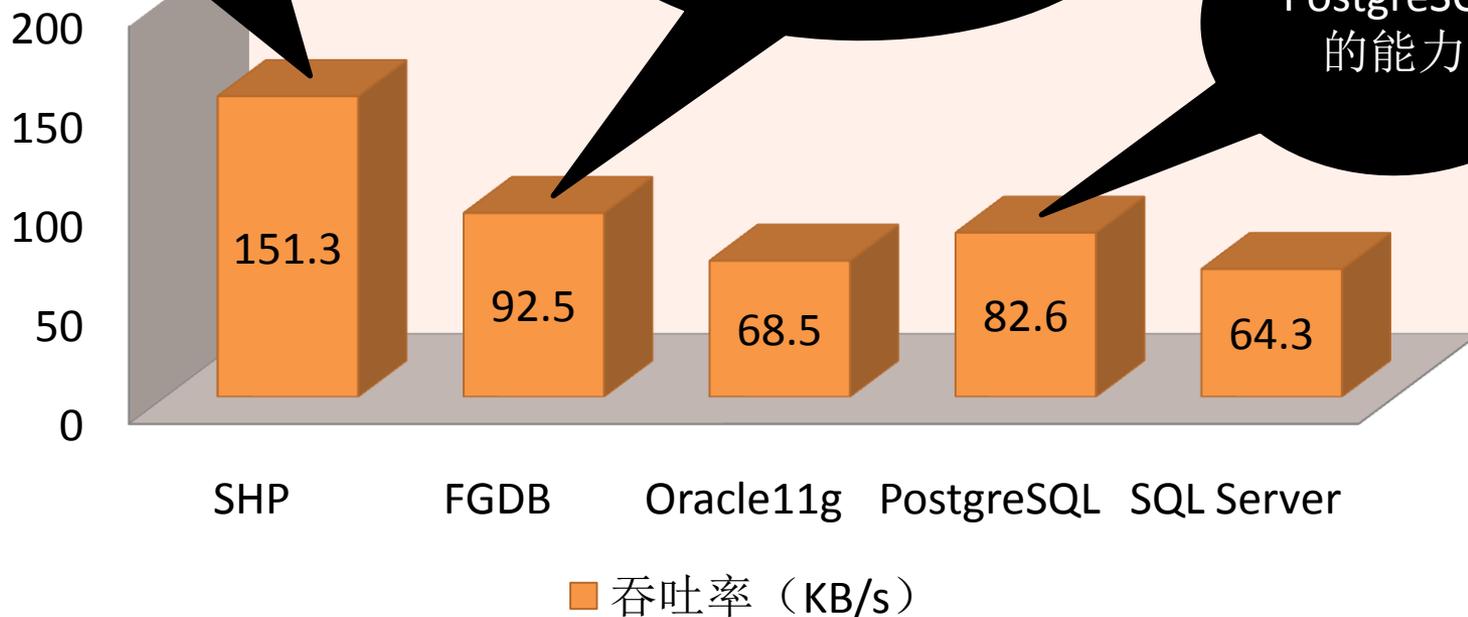


# 大数据性能对比图

shapefile在不使用标注的情况下出图性能很好

File Geodatabase的性能依然非常出色

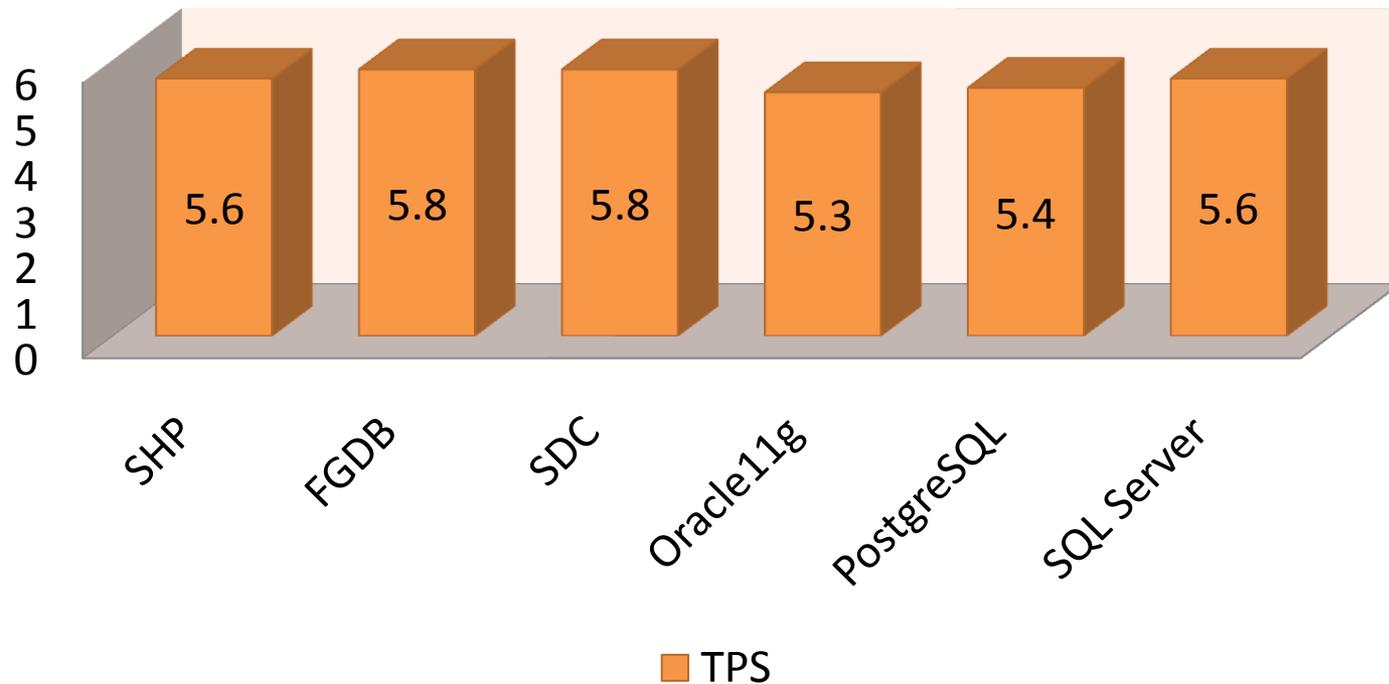
PostgreSQL的能力



- 更大数据量的矢量数据，有更多测试结果和比较，有兴趣可以参与明天的讲座——《海量空间数据库实施策略》

# ● 不同数据库

数据量较小时基本没有区别

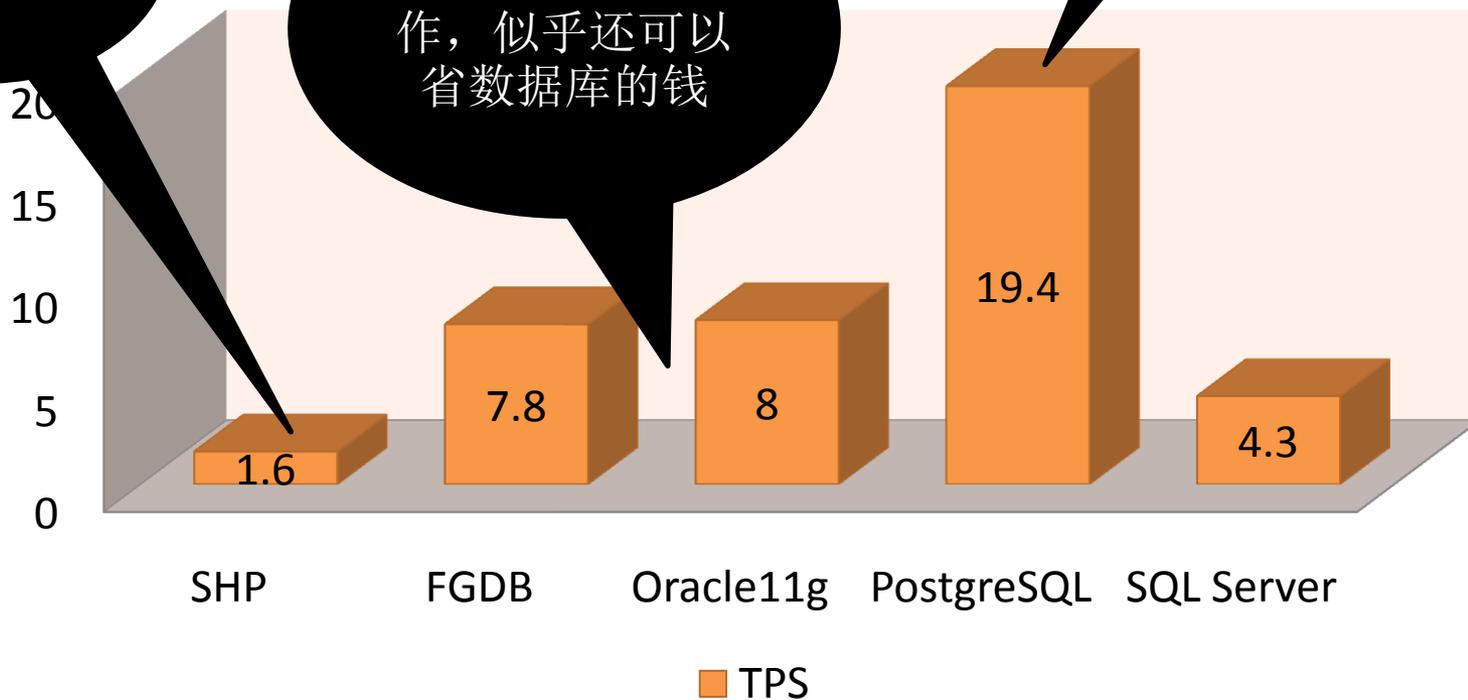


# 同数据源下的数据库矢量查询

Shapefile  
劣势明显

如果是只读操作，似乎还可以省数据库的钱

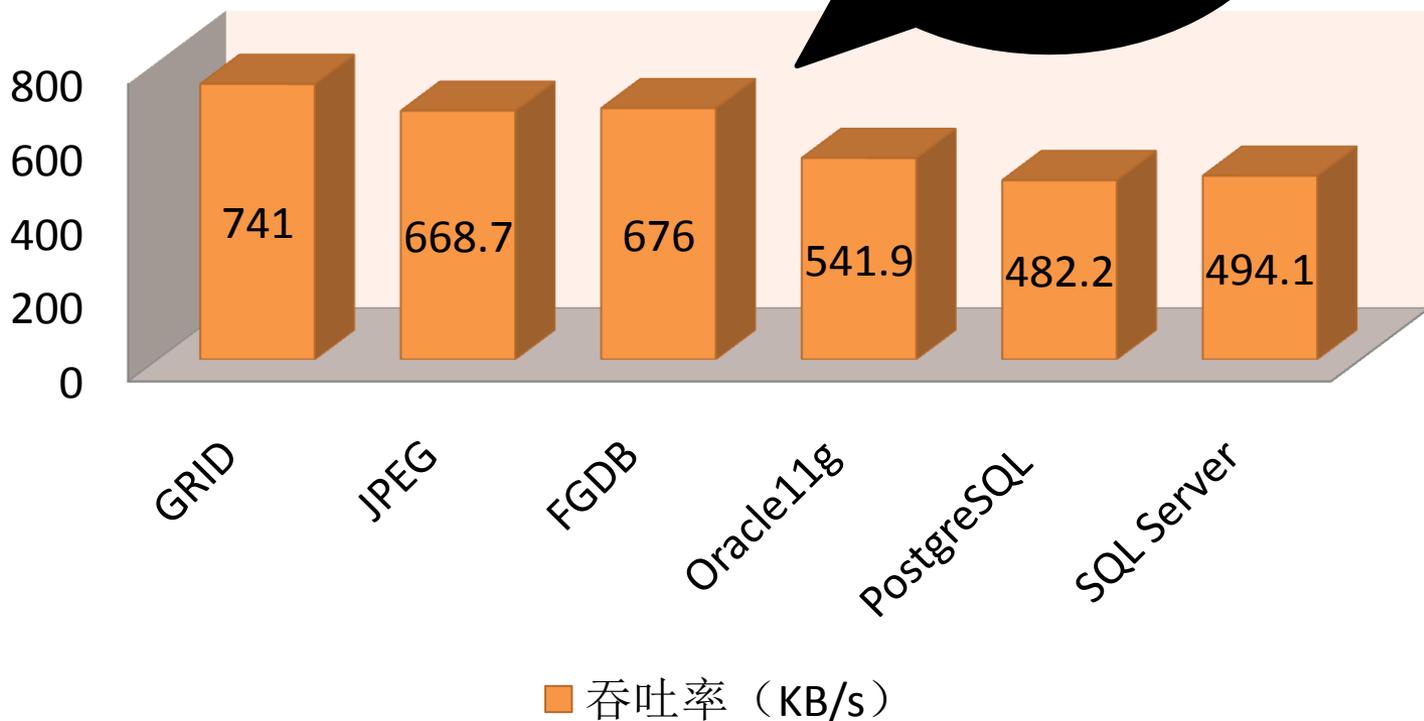
PostgreSQL  
免费量又足



# ● 不同数据源栅格出图

## ➤ JPEG2000压缩

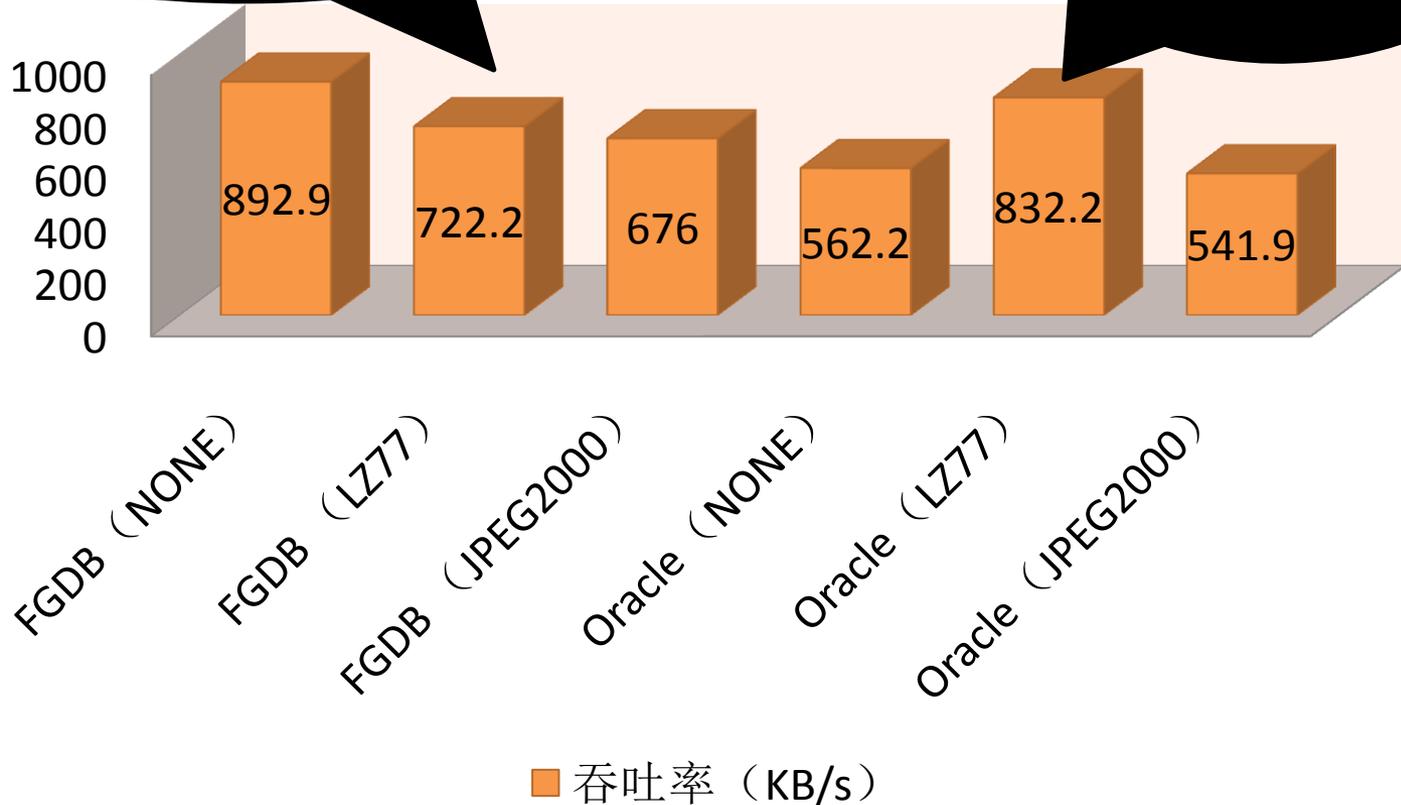
文件存储普遍  
比入库后要快



文件形式的存储越压缩  
性能越差

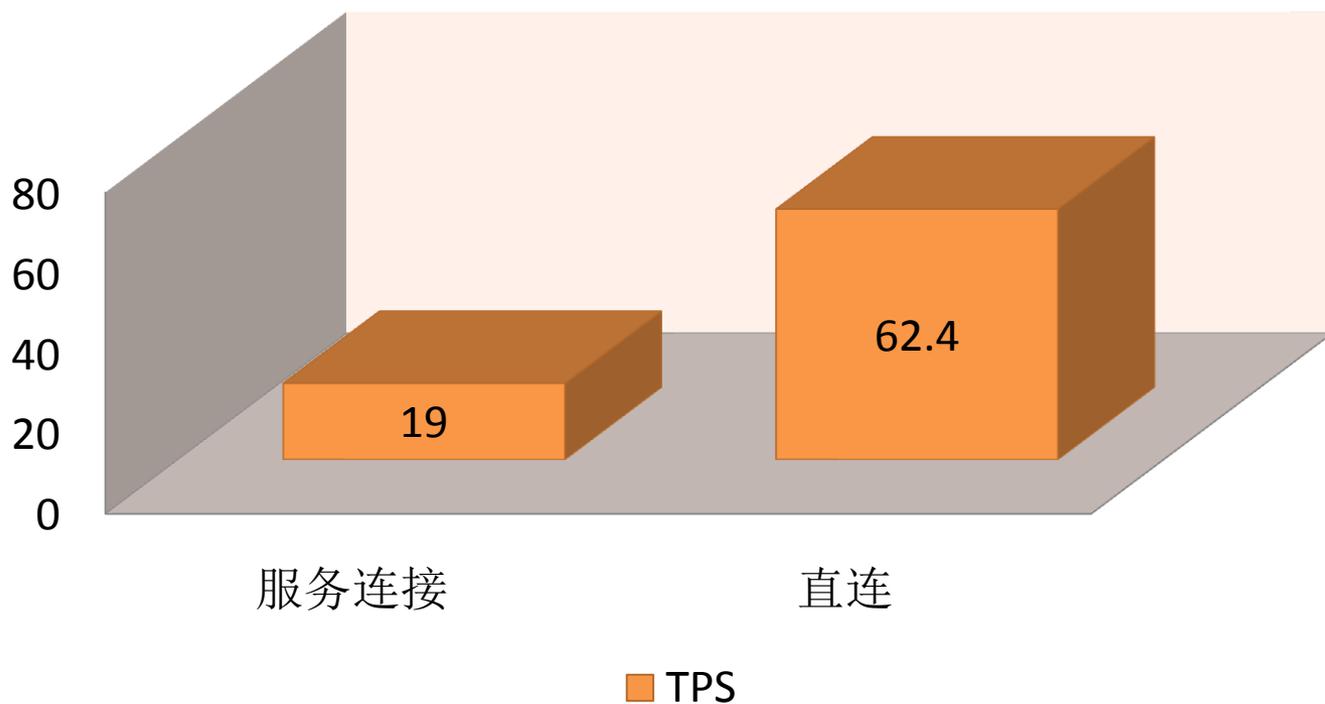
## 压出图

数据库中可能在  
I/O和压缩结算间  
达到较优值



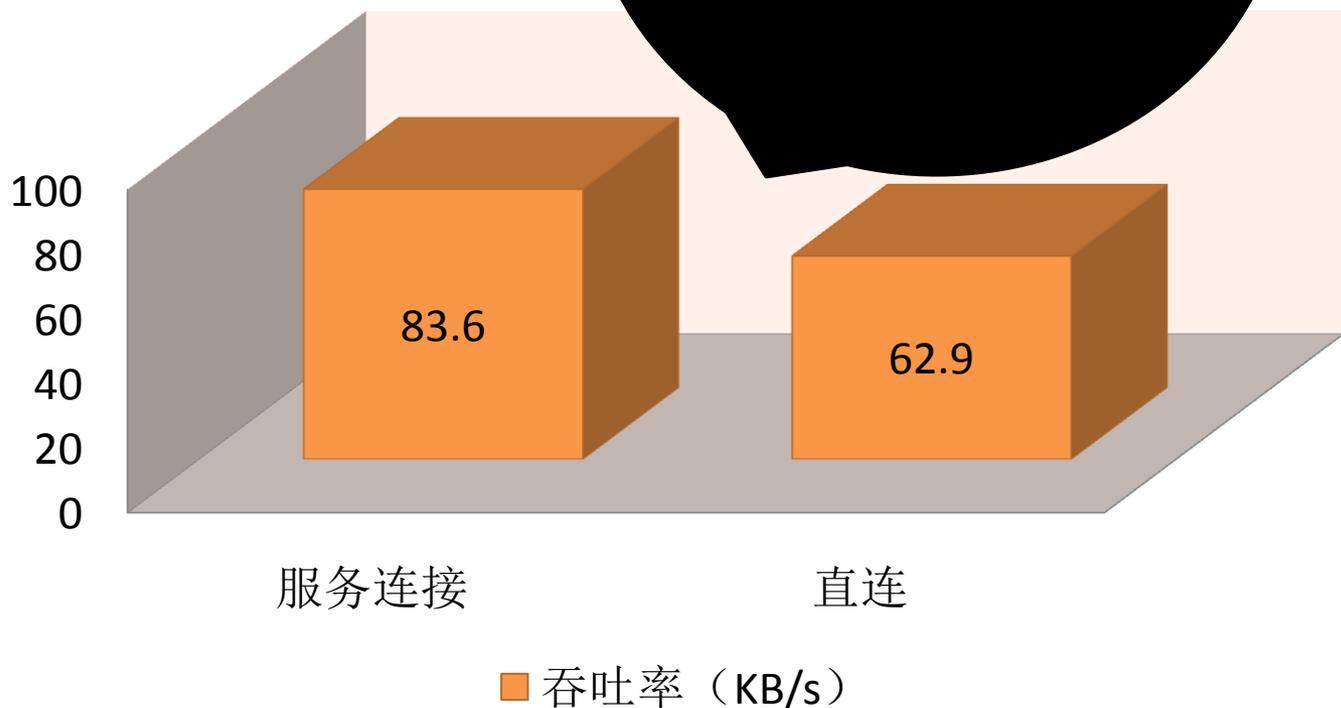
- 更大数据量的栅格数据，情况和这里还  
有所区别，有兴趣可以参与明天的讲  
座——《海量空间数据库实施策略》

# ● SDE服务连接 vs. 直连 矢量查询



## ● SDE服务连接 vs.

但是这个结果并非绝对，在某些情况下可能结果也会相反。一般还是推荐直连。



- 数据库参数调整
- 创建/维护 属性索引
- 更新数据库统计信息
- 版本压缩
- 检查空间索引

- 数据源种类太多，无法穷尽。
- 在可选的基础上，模拟业务情况，测试+比较。
- 明天的讲座《海量空间数据库实施策略》中会有一些供参考的基准结果。

# ArcGIS Server 配置的影响

## ● MIME 瓶颈

➤ REST export image/SOAP returnMimeData

➤ 网络带宽

## ● URL 瓶颈

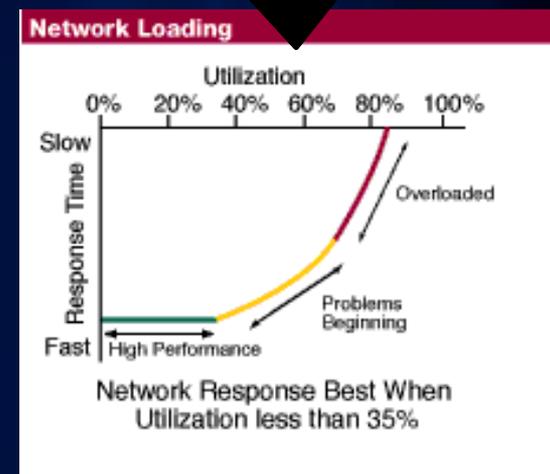
➤ REST export html/SOAP returnURL/ADF/WMS

➤ 网络带宽

➤ 磁盘性能

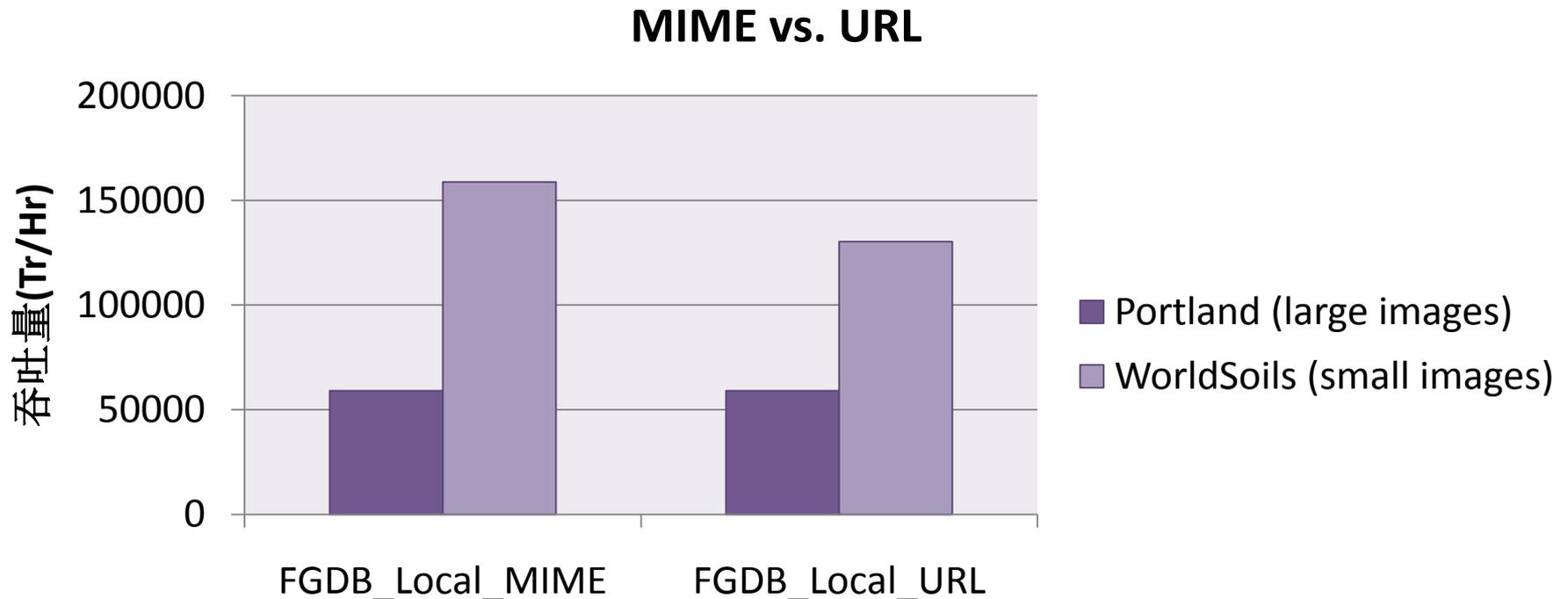
➤ 不同机器不要使用共享虚拟目录

网络负载超过  
约35%响应时  
间急剧下降



## ● MIME vs. URL

- 一般磁盘总是先于网络成为系统瓶颈
- MIME伸缩性较好
- 输出小图片MIME更有优势



## ● 安全与认证

### ➤ SSL

### ➤ 本地安全验证服务LSASS ( Win2003 )

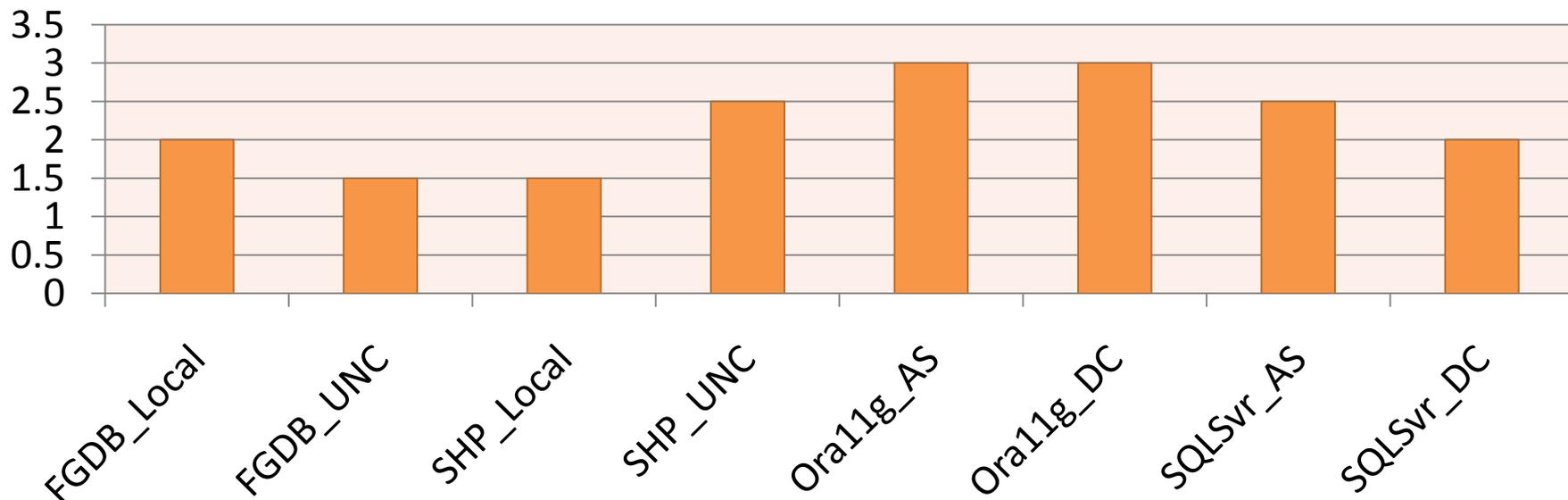
<http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=32620>

### ➤ Web层的安全机制

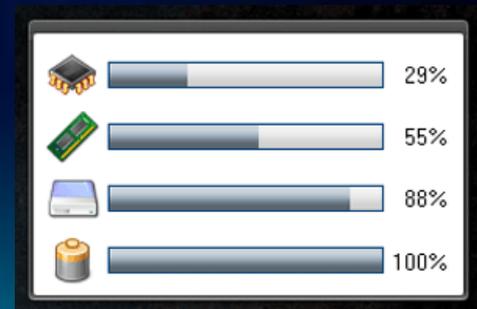
## ● 实例数

- 实例数的优化和服务类型、数据源有着比较复杂的关系
- 经验公式：每个CPU核配2-4个实例

不同数据源类型性能峰值时每CPU核配置的实例数



- 这是一个经验参考，不要拘泥
- 有些情况下很可能不同
- 建议在这个经验值的基础上进行调整
- 模拟业务情况，查看系统资源使用情况



# 服务的影响

## ● Map Service

### ➤ 优化MXD

- 设置有利于绘制的符号
- 避免使用动态投影
- 优化标注（使用Annotation/避免Maplex）
- 避免跨库的Join操作
- 避免使用小波压缩的栅格数据（JPEG2000等）
- 简化数据
- 设置显示比例尺范围



## ● Map Service

- 给底图创建切片缓存
- 如果业务图层改变并非十分频繁，也创建切片缓存
- 在客户端进行业务数据绘制  
( Flex/Silverlight/JavaScript API )

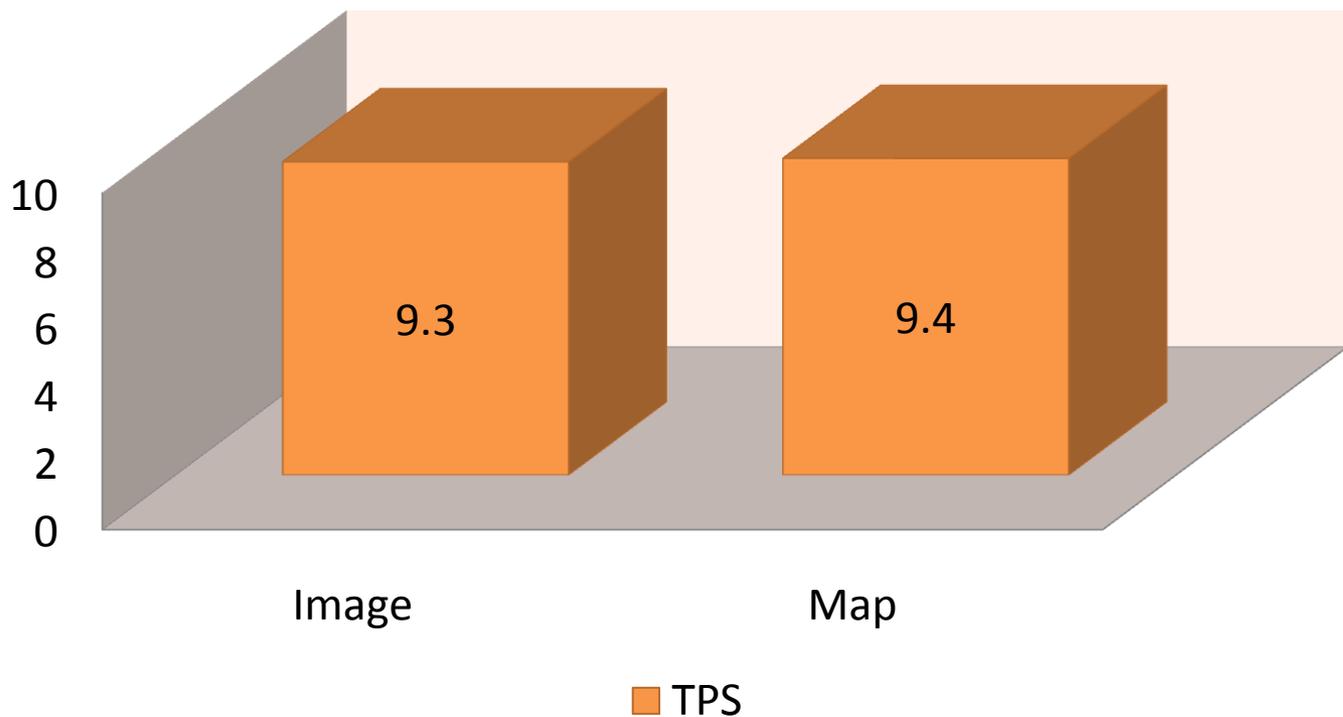
## ● GP Service

- 使用同步交互模式
- 使用本机的Jobs目录
- 对大栅格数据进行分幅
- 使用原生未压缩的栅格类型，比如TIFF
- 尽量通过地图图层去访问数据库中的数据
- 中间/最终结果尽量使用内存进行操作

## ● GP Service

- 可以预先计算的环节首先进行预处理
- 使用本地的数据和资源
- 避免不必要的坐标转换
- 给使用到的属性字段添加索引
- 简化数据

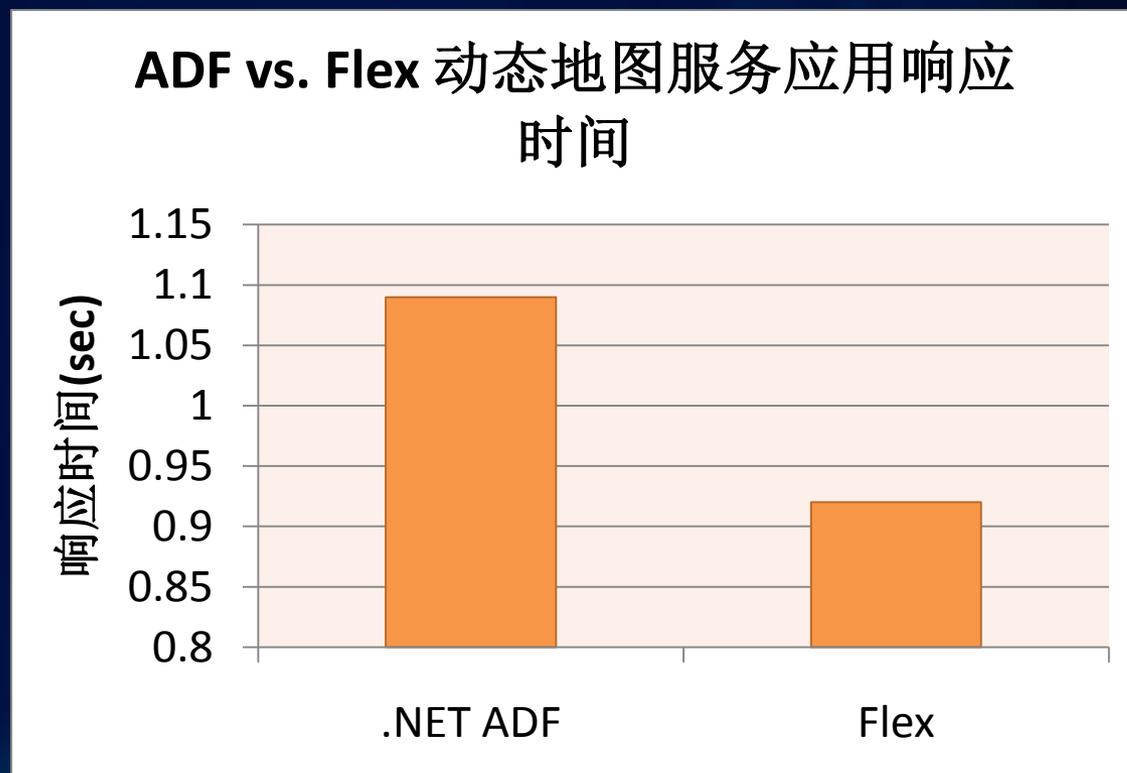
# ● Image Service vs. Map Service



# 开发框架的影响

- 判断是否能满足应用的功能需求
- 能否尽量提升性能和伸缩性
- 技术成本
- 可选的开发框架：
  - .Net/Java ADF
  - JavaScript/Flex/Silverlight API
  - SOAP SDK
  - ArcObjects API

## ● ADF vs. Flex



# ArcGIS Server 的部署策略



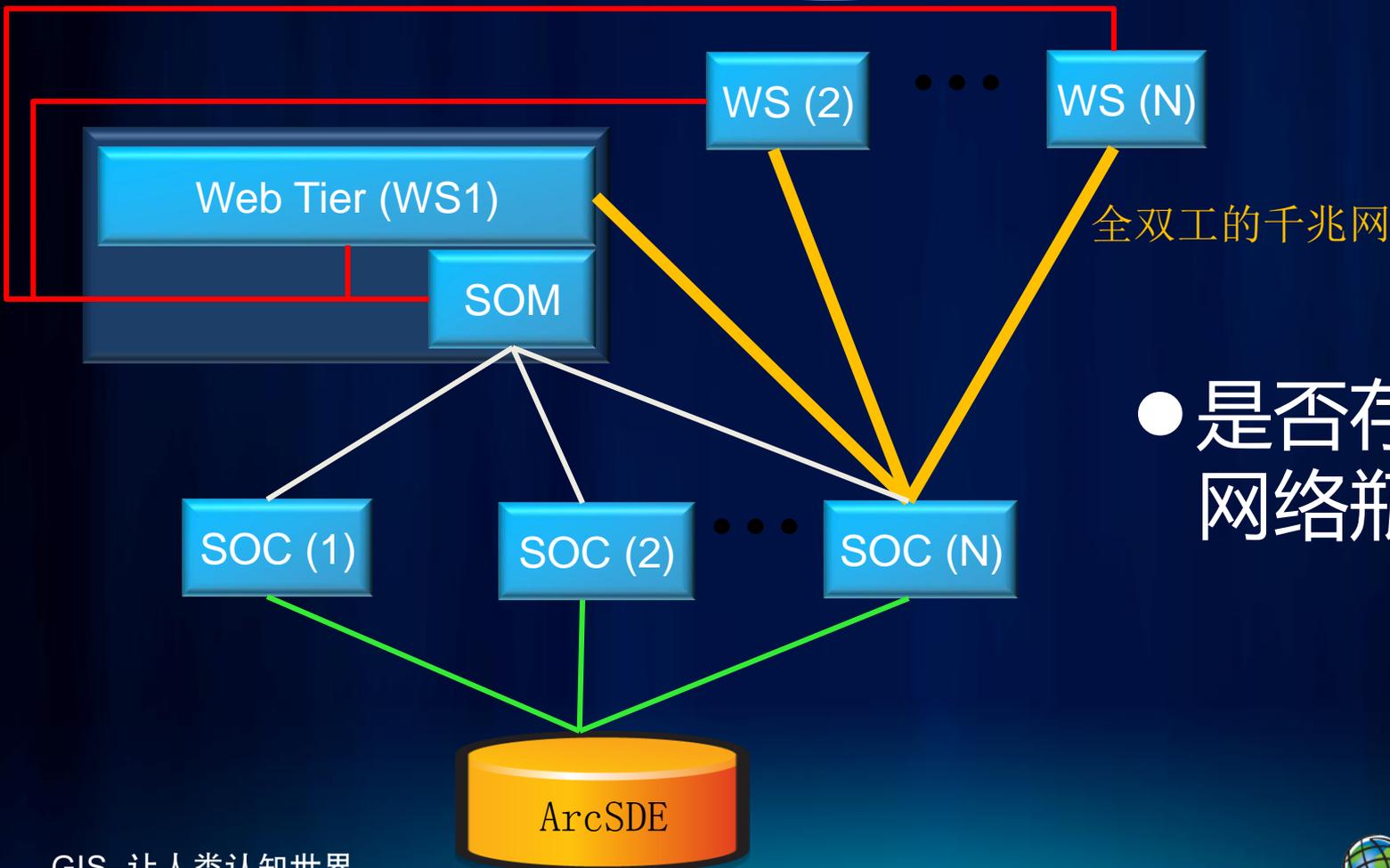
# 单ArcGIS Server部署



- 对于ArcSDE数据源，尽量保证一个单独的应用服务器运行数据库
- 对于文件数据源，尽量保证与ArcGIS Server在同一机器
- ArcGIS Server 的Output 目录尽量与之在同一机器

# 多SOC部署 (MIME)

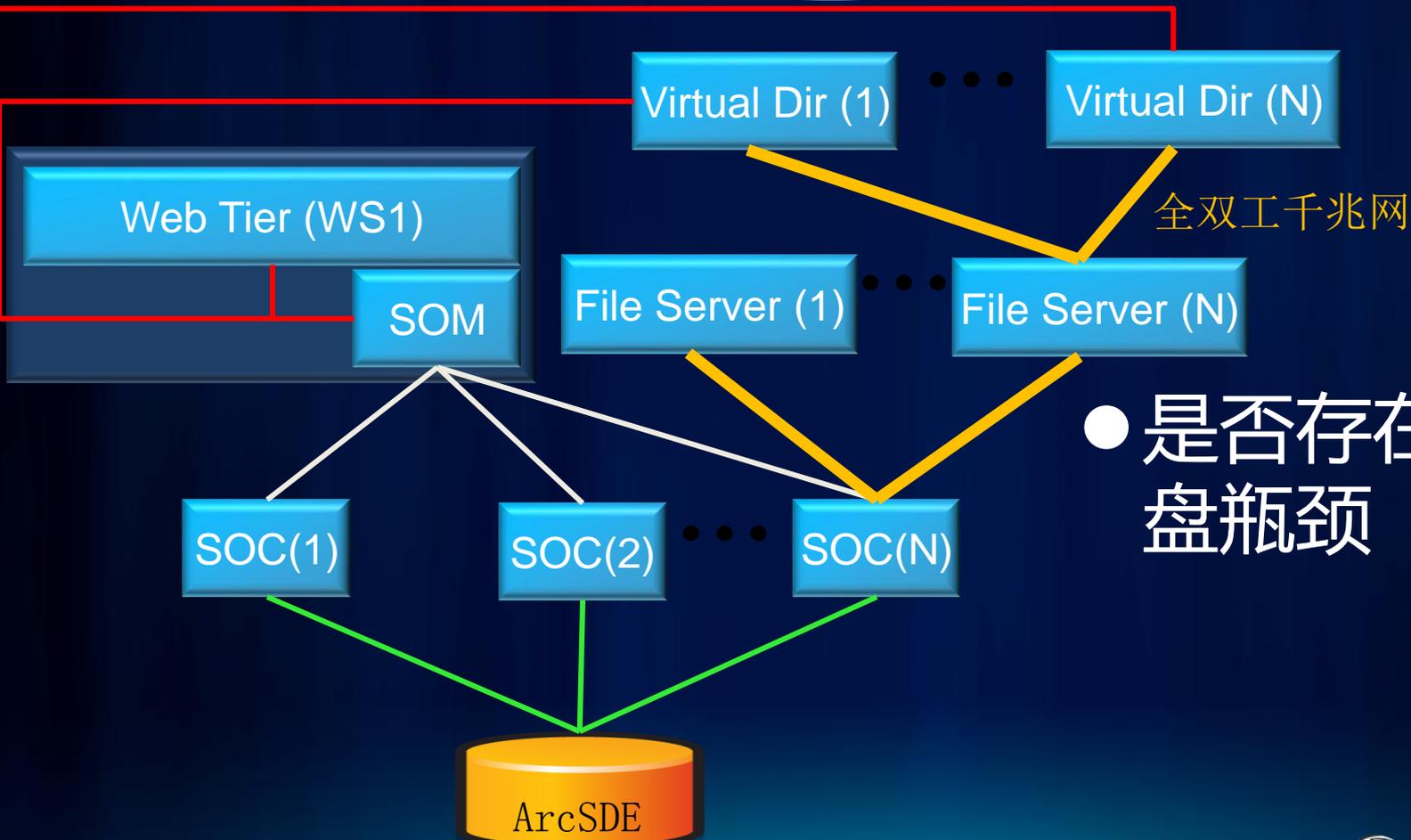
网络负载均衡



● 是否存在网络瓶颈

# 多SOC部署 (URL)

网络负载均衡



● 是否存在磁盘瓶颈

# 多ArcGIS Server部署（简易部署）



- 较少的瓶颈
- 部署简单，伸缩性强
- 重复配置服务的缺点

谢谢！

