分布式微服务：

单体架构：

一个程序，向外界提供单一的服务。

一个厨子向外界提供服务。

集群：

两个程序，向外界提供一模一样的单一的服务。

两个厨子向外界提供服务。

分布式架构：

多个程序，整合组合向外界提供服务。

一个厨子，一个服务员，一个财务向外界提供服务。

分布式：

早期的分布式

Ejb

Corba

WebService(SOAP,WSDL)

现在：

Dubbo+ZooKeeper(大型电商 京东，天猫 当当Dubbox)

Spring Cloud+Eureka(中小型企业应用)



我们主要讲3个：

Redis:必须会，必须在你的项目中使用3个点以上。

Spring Cloud：主要是听。

RabbitMQ：主要是听。

MyBatis的分页：

1：在Mybatis xml文件中直接写limit(物理分页).

传入limit 的两个参数。分页是有程序员来完成的。不利于维护。更换数据库，更改逻辑的结果，xml全部得重写。

2：用Mybatis自带的RowBounds进行分页（内存分页）。

1：xml一个字都不改。

2：接口中加一个参数

List<Userinfo> selectByExample(UserinfoExample example,RowBounds rb);

3：就可以用了

List<Userinfo> list = userinfoMapper.selectByExample(null, new RowBounds(5, 5));

3：自己写分页的插件：

原理：

Mybatis的所有数据库的预处理。都是由

StatementHandler对象，的prepare方法来进行处理的。

写一个拦截器，拦截StatementHandler对象的prepare方法。

来判断你是否要进行分页(RowBounds)

如果不分页

放行

如果分页

要把内存分页，更改为物理分页。

更改你要执行的SQL语句。

Mysql:

Select \* from table limit 0,10

Oracle:

Oracle的伪列 rowid rownum

select \* from

(select empno,ename,job,sal,deptno,rownum rn from emp where rownum<10) t

where t.rn>5

索引很重要：

CREATE INDEX myindex ON person(NAME)

有效的利用索引，肯定能够提高程序的查询的速度。

频繁维护(CRUD)的列，适不适合建立索引？

在Mysql5.0 innodb引擎 只有一种索引结构 B+树

在Oracle 11G 只有两种索引结构 B\*树，BitMap索引。

因为B+树的数据结构，你维护数据，绝对不会导致索引表全部重新排列。而只是影响相邻接点的引用。

B+,B\*树索引适合建立在高基数的列中。适合用于频繁的维护的CRUD。比如身高，年龄等。

BitMap索引适合建立在低基数的列中，比如Sex。