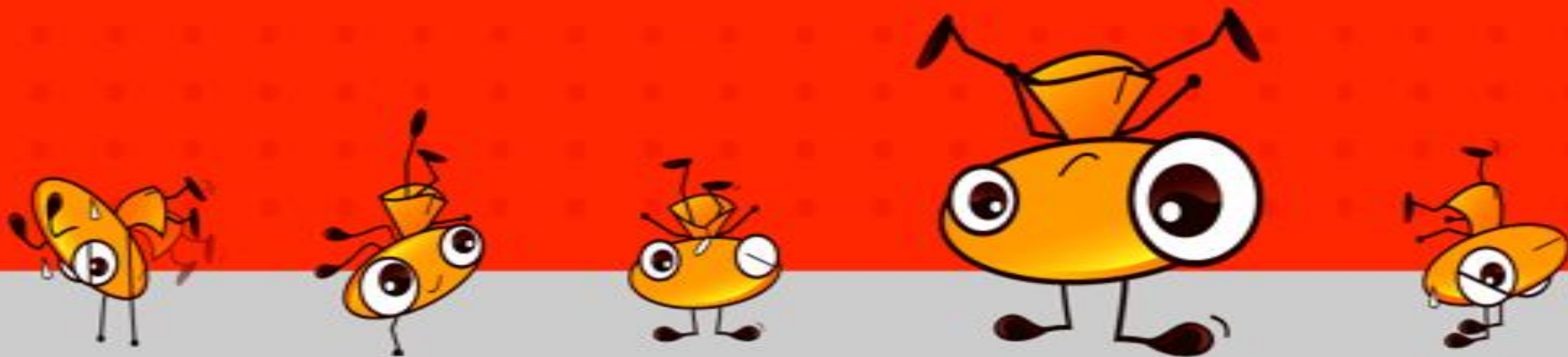


# 数据库架构演变 (2003-2010)

丁原

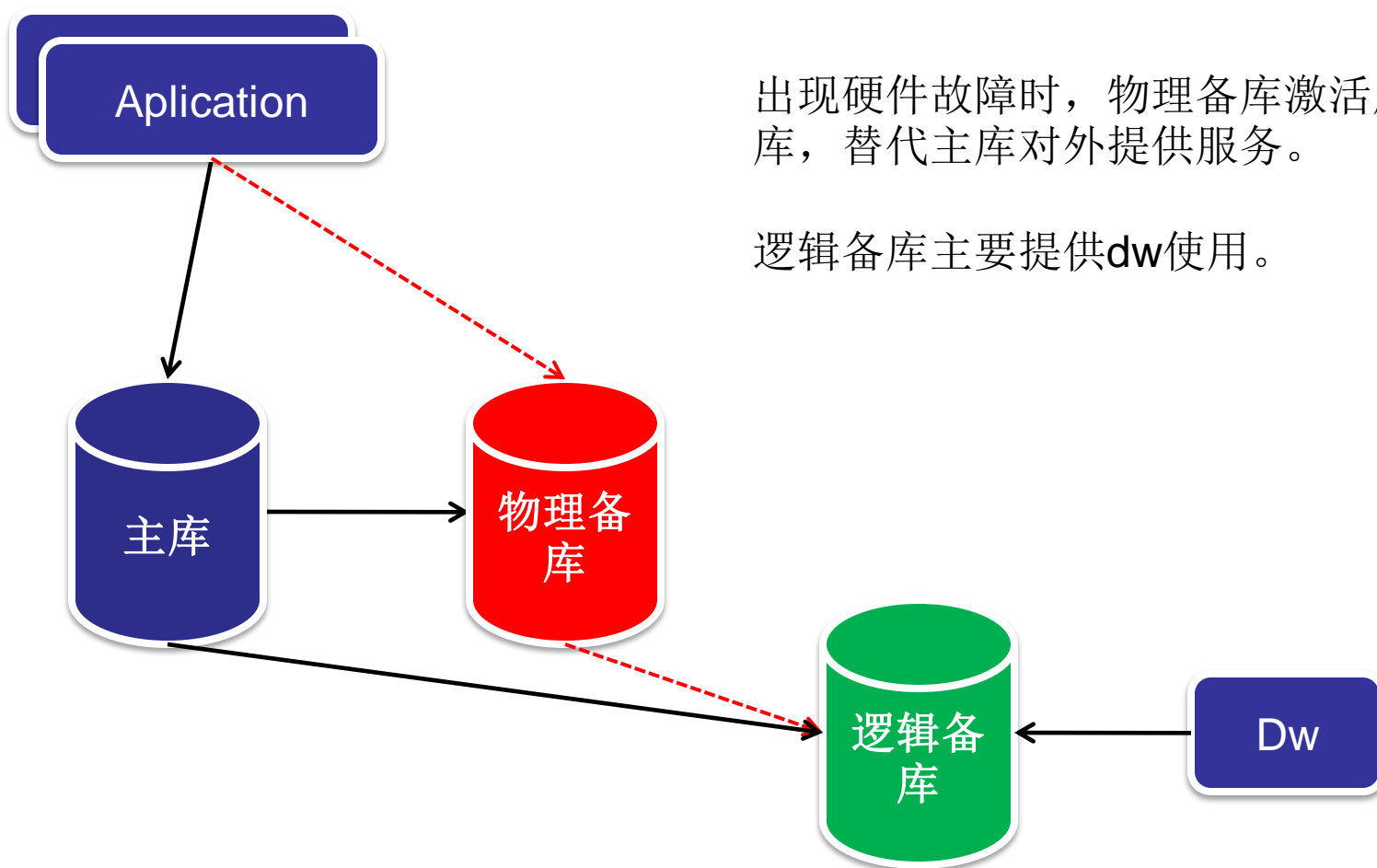
dingyuan@taobao.com

日期: 2010.06



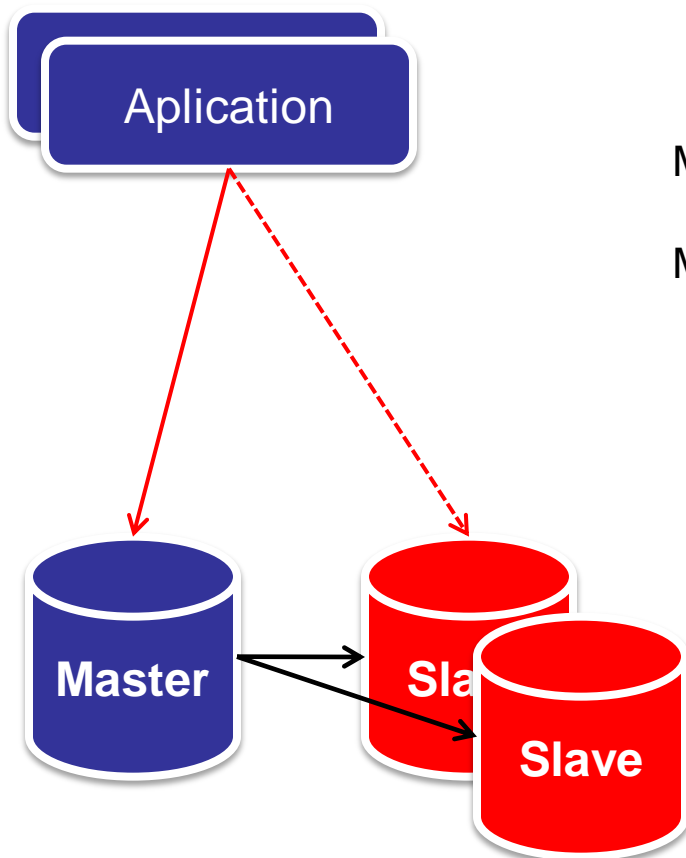


- 数据库基本框架
- 数据库架构演变
- 案例：交易核心数据库演变关键点



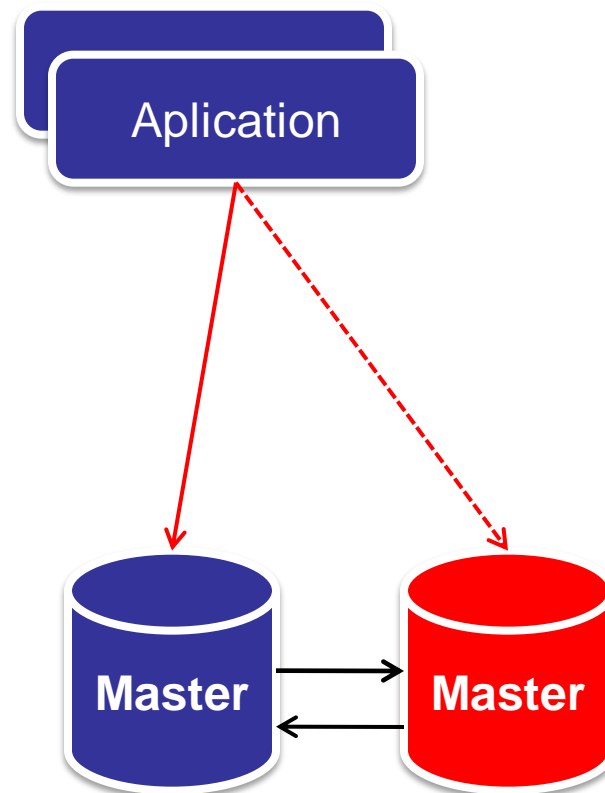
出现硬件故障时，物理备库激活成为主库，替代主库对外提供服务。

逻辑备库主要提供dw使用。



M-S架构:

M-M架构:




















主备切换如何做到不影响到应用，不需要开发人工干预？

```
tbtest =  
  (DESCRIPTION =  
    (failover = on )  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.0.1)(PORT = 1521))  
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.0.2)(PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = tbtest)  
    )  
  )
```



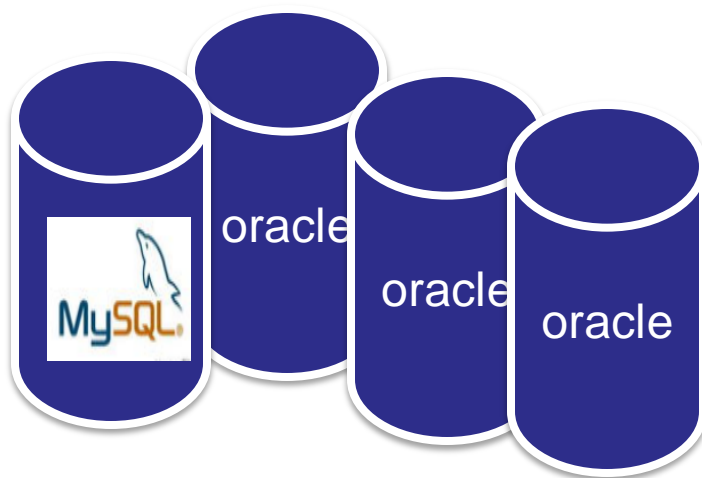
主备数据库进行独立的管理，配置两个数据源。  
数据源中**哪一个是活跃**的，取决于ConfigServer(配置中心)上的配置。

Configserver:

序号	是否修改	名称	SID			
Oracle						
18	<input checked="" type="checkbox"/>	keystore_master	keystore			
19	<input type="checkbox"/>	tbdbetu	ctu			
20	<input type="checkbox"/>	tc_log	tc_log			
21	<input type="checkbox"/>	tesnap	tesnap			
发布所有						



- 数据库基本框架
- 数据库架构演变
- 案例：交易核心数据库演变关键点



2003年：  
快速开发  
Mysql，pc服务器

2004年：  
稳定性，高性能  
逐步开始采用oracle，小型机，  
高端硬件存储

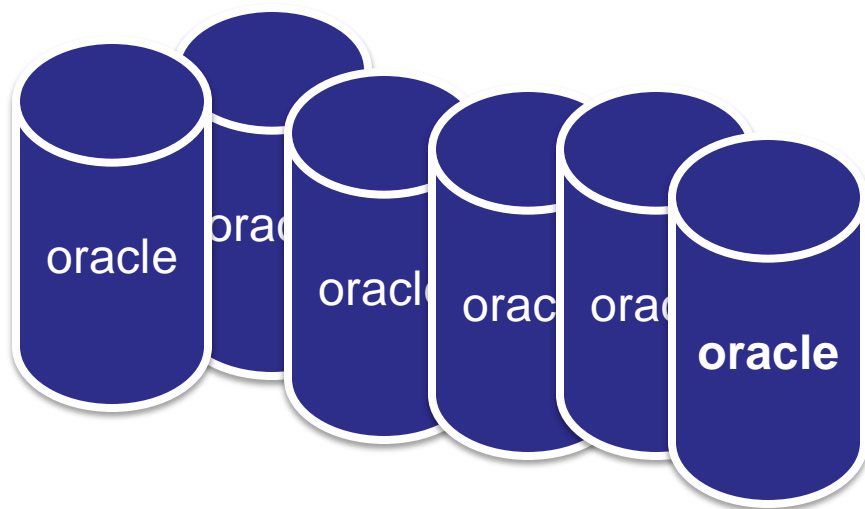
oracle：商品，交易，评价，收藏，用户等（3套oracle环境）

网站迅猛发展，数据库一定要保证高可用性，最稳妥的方法？





找“老板”投钱  
Oracle，小型机，高端硬件存储



2008年：  
业务迅猛发展，单  
台小型机很快就达  
到瓶颈，开始进入  
大规模垂直拆分  
阶段

高可用带来了高成本

Oracle：商品，交易，用户，评价，收藏夹（8-10套数据库）



优点：

减少应用之间的耦合

数据库业务单一，可以针对具体的业务类型优化

缺点：

硬件成本，Oracle license费用

垂直扩展可能带来的问题？



3->8套->16套->?

物理主库，物理备库，逻辑备库

oracle，小型机，高端存储，可怕的硬件投入



找“老板”继续投钱？

可怕的成本，可怕的垂直扩展

垂直扩展并没有打破集中式，可怕的集中式



淘宝不断发展，系统压力增长远远超过2倍/每年，新业务不断上线，在好的硬件也很容易达到瓶颈，**水平拆分？**

问题：

- 1.需要解决拆分带来的成本问题
- 2.我们会增加很多服务器，必须要解决可维护性
- 3.我们也要解决可扩展性



去Oracle  
去小型机  
去高端存储

Mysql, 廉价pc服务器  
应用上做容灾, 不在过度依赖数据库 依赖硬件

分布式, 低成本, 可扩展, 易维护



他能搞定一切吗？



1. 存数据
2. 单条查询 (querybyid)
3. 多表关联sql查询
4. 通过存储过程，函数来处理业务
5. 大量数据实时在线分析 (sum, count, group by)

我们对数据库的定位是什么？





数据库尽量只负责保存数据  
尽量通过应用服务器来分摊复杂计算（order by、sum、group by、count..）

Isearch（搜索，实时搜索）

Tair（基于key value的全内存系统）

Tfs（taobao file system）

Nosql（Cassandra。。。）

Bigtable





- 数据库基本框架
- 数据库架构演变
- 案例：交易核心数据库演变关键点



卖家交易后台管理:

宝贝名称:  成交时间: 从  00:00 到  00:00

买家昵称:  订单状态:  评价状态:

订单编号:  物流服务:  售后服务:

代充类型:  店铺名称:

淘宝网严禁出售2010年上海世博会相关商品

所有订单	等待买家付款	等待发货	已发货	退款中	需要评价	成功的订单	历史订单
宝贝	单价(元)	数量	售后	买家	交易状态	实收款(元)	评价
全选 批量发货 批量备注							
订单编号: 39216166999858 成交时间: 2010-06-09 05:11							
无图	限定在30个汉字内	8.80	2	rico005 成旭华	等待买家付款	17.60 (卖家包邮)	-
订单编号: 39216166349858 成交时间: 2010-06-09 05:11							
无图	限定在30个汉字内	8.80	1	rico005 成旭华	等待买家付款	8.80 (卖家包邮)	-
共有142165条记录   1 2 3 ... 7109 下一页 到第 页 确定							



1. 模糊查询
2. 大量数据count操作
3. list查询分页查询
4. 查询条件复杂，用户可以动态选择

备注：

交易实时性要求非常高，需要实时展示，不能有延迟

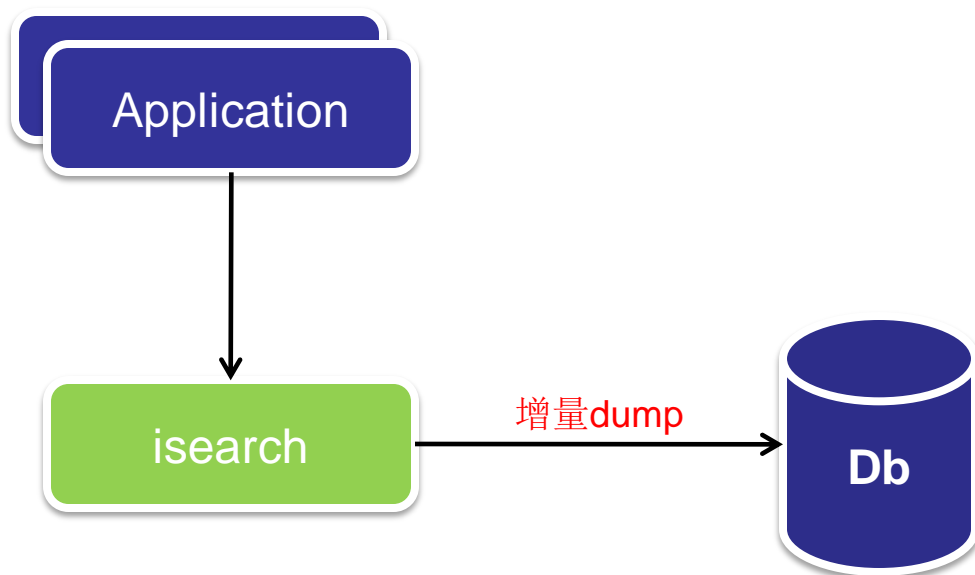


这球怎么踢？





大数据量处理尽量通过搜索来实现。  
相比其他的方案，搜索很好的解决了标题的模糊like查询。







通过监控平台采集了部分sql的执行量（12小时）

AWR快照报表

SQL全文索引

应用会话报表

索引访问报表

记录变更报表

序列监控报表

数据字典报表

表空间报表

段空间报表

SQL全文搜索

SQL全文搜索	<input type="text"/>	提交	SQL区段分析报表	SQL未绑定变量报表	说明: 每天采集1次,execs是sql在1天内的执行次数.
---------	----------------------	----	-----------	------------	--------------------------------

sql_id	sql_fulltext	fsc	exe	ex%	buf	bu%	dsk	di%	ela	el%	cp%	pc%	so%	row	app
792xm9h6pxhk2	select id, title, gmt_modified, star	NO	0.39G	25.62	0.71	3.31	0.47	15.67	0.00	9.35	13.13	1.7	0	0.99	ops
8htqxswwu8d7h	select id, title, gmt_modified, star	NO	0.31G	19.97	1.8	6.42	0.69	18.01	0.00	10.6	14.07	3.06	0	0.60	ops
dy3axn3fxaubd	select id, user_num_id, item_id, auc	NO	0.14G	9.4	0.90	1.54	1.6	20.16	0.01	32.43	33.11	16.47	0	1.6	ite
4n7675hwwcndc	select sku_id, item_id, properties,	NO	77.52M	4.93	5.7	5.15	1.2	7.41	0.00	7.86	8.21	4.08	0	7.1	ite
22fwsj60kkp63	select sku_id, item_id, properties,	NO	71.17M	4.53	5	4.13	0.43	2.53	0.00	3	3.77	0.03	0	0.99	ite
39rbqj3ck76w3	select id, title, gmt_modified, starts,	NO	63.91M	4.07	5.3	3.96	0.01	0.05	0.00	0.13	0.26	2.92	0	0.99	ite
2fbcckh4df0dr	select /*+ rowid(a)*/ auction_id, opt	NO	53.9M	3.43	1	0.63	0.00	0.01	0.00	0	0	1.33	0	0.99	sea
fpnshqn3zagc3	select id, user_id, ypid, token, yem	NO	40.68M	2.59	1.4	0.67	0.18	0.59	0.00	1.03	1.33	0	0	0.21	loq

数据库接近4万次/每秒的查询，每个小时在1.5亿次左右，还在快速增加中。





读太多（单台查询，多条查询）  
更新量太大，每天将近1亿次的更新  
sql执行量增长非常快

备注：

实时性要求非常高

不管是卖家还是买家，肯定不乐意看到付款的成功订单，系统却显示未付款。

如何解决读？



Tair

实时搜索

通过廉价pc实现读写分离

其他

我们选择了什么？



通过廉价pc实现读写分离

场景：

1. 买家查询
2. 卖家查询
3. 单条id查询
4. 关联查询
5. 其他查询

拆分如何兼顾、解决多维度查询呢？

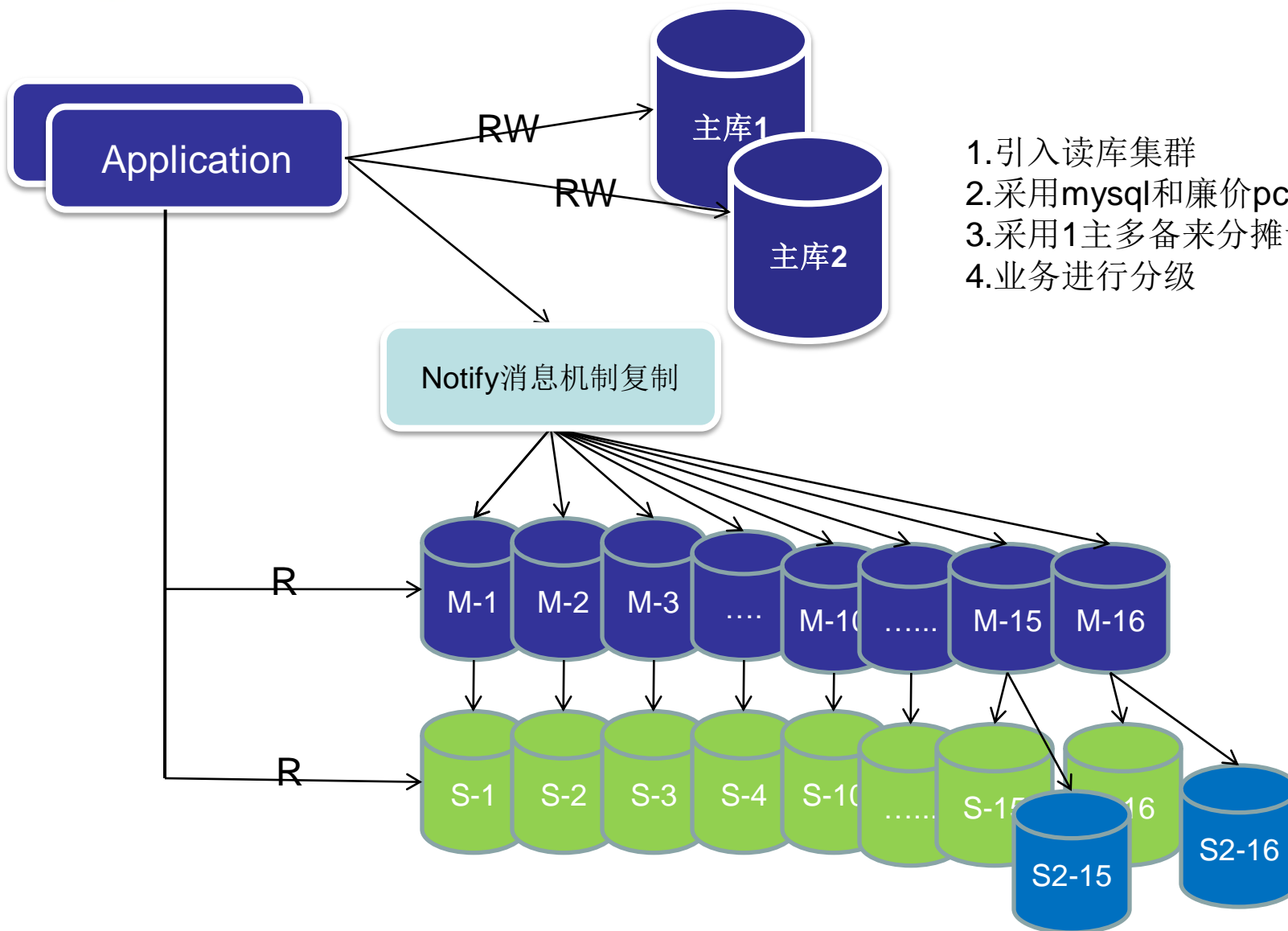


两份数据

按照不同维度拆分，承担各自不同的业务场景。

框架：

两份数据+读写分离



1. 引入读库集群
2. 采用mysql和廉价pc服务器
3. 采用1主多备来分摊读压力
4. 业务进行分级



垂直拆分  
水平拆分  
读写分离

对数据库的定位



# 交流时间