



13 | 架构设计流程：详细方案设计
李运华



- 00:00 / 08:58

完成备选方案的设计和选择后，我们可以长出一口气，因为整个架构设计最难的一步已经完成了，但整体方案尚未完成，架构师还需继续努力。接下来我们需要再接再厉，将最终确定的备选方案进行细化，使得备选方案变成一个可以落地的设计方案。所以今天我来讲讲[架构设计流程第4步：详细方案设计](#)。

架构设计第4步：详细方案设计

简单来说，详细方案设计就是将方案涉及的关键技术细节给确定下来。

- 假如我们确定使用Elasticsearch来做全文搜索，那么就需要确定Elasticsearch的索引是按照业务划分，还是一个大索引就可以了；副本数量是2个、3个还是4个，集群节点数量是3个还是6个等。
- 假如我们确定使用MySQL分库分表，那么就需要确定哪些表要分库分表，按照什么维度来分库分表，分库分表后联合查询怎么处理等。
- 假如我们确定引入Nginx来做负载均衡，那么Nginx的主备怎么做，Nginx的负载均衡策略用哪个（权重分配？轮询？ip_hash？）等。

可以看到，详细设计方案里面其实也有一些技术点和备选方案类似。例如，Nginx的负载均衡策略，备选有轮询、权重分配、ip_hash、fair、url_hash五个，具体选哪个呢？看起来和备选方案阶段面临的问题类似，但实际上这里的技术方案选择是很轻量级的，我们无须像备选方案阶段那样操作，而只需要简单根据这些技术的适用场景选择就可以了。

例如，Nginx的负载均衡策略，简单按照下面的规则选择就可以了。

- 轮询（默认）

每个请求按时间顺序逐一分配到不同的后端服务器，后端服务器分配的请求数基本一致，如果后端服务器“down掉”，能自动剔除。

- 加权轮询

根据权重来进行轮询，权重高的服务器分配的请求更多，主要适应于后端服务器性能不均的情况，如新老服务器混用。

- ip_hash

每个请求按访问IP的hash结果分配，这样每个访客固定访问一个后端服务器，主要用于解决session的问题，如购物车类的应用。

- fair

按后端服务器的响应时间来分配请求，响应时间短的优先分配，能够最大化地平衡各后端服务器的压力，可以适用于后端服务器性能不均衡的情况，也可以防止某台后端服务器性能不足的情况下还继续接收同样多的请求从而造成雪崩效应。

- url_hash

按访问URL的hash结果来分配请求，每个URL定向到同一个后端服务器，适用于后端服务器能够将URL的响应结果缓存的情况。

这几个策略的适用场景区别还是比较明显的，根据我们的业务需要，挑选一个合适的即可。例如，比如一个电商架构，由于和session比较强相关，因此如果用Nginx来做集群负载均衡，那么选择ip_hash策略是比较合适的。

详细设计方案阶段可能遇到的一种极端情况就是在详细设计阶段发现备选方案不可行，一般情况下主要的原因是备选方案设计时遗漏了某个关键技术点或者关键的质量属性。例如，我曾经参与过一个项目，在备选方案阶段确定是可行的，但在详细方案设计阶段，发现由于细节点太多，方案非常庞大，整个项目可能要开发长达1年时间，最后只得废弃原来的备选方案，重新调整项目目标、计划和方案。这个项目的主要失误就是在备选方案评估时忽略了开发周期这个质量属性。

幸运的是，这种情况可以通过下面方式有效地避免：

- 架构师不但要进行备选方案设计和选型，还需要对备选方案的关键细节有较深入的理解。例如，架构师选择了Elasticsearch作为全文搜索引擎，那么就需要让架构师自己

对Elasticsearch的设计原理有深入的理解。比如索引、副本、集群等技术点；而不能道听途说Elasticsearch很牛，所以选择它，更不能成为把“细节我们不讨论”这句话挂在嘴边的“PPT架构师”。

更多一手资源请添加QQ/微信1182316662

- 通过分步骤、分阶段、分系统等方式，尽量降低方案复杂度，方案本身的复杂度越高，某个细节推翻整个方案的可能性就越高，适当降低复杂性，可以减少这种风险。
- 如果方案本身就很复杂，那就采取设计团队的方式进行设计，博采众长，汇集大家的智慧和经验，防止只有1~2个架构师可能出现的思维盲点或者经验盲区。

详细方案设计实战

虽然我们上期在“前浪微博”消息队列的架构设计挑选了备选方案2作为最终方案，但备选方案设计阶段的方案粒度还比较粗，无法真正指导开发人员进行后续的设计和开发，因此需要在备选方案的基础上进一步细化。

下面我列出一些备选方案2典型的需要细化的点供参考，有兴趣的同学可以自己尝试细化更多的设计点。

1. 细化设计点1：数据库表如何设计？

- 数据库设计两类表，一类是日志表，用于消息写入时快速存储到MySQL中；另一类是消息表，每个消息队列一张表。
- 业务系统发布消息时，首先写入到日志表，日志表写入成功就代表消息写入成功；后台线程再从日志表中读取消息写入记录，将消息内容写入到消息表中。
- 业务系统读取消息时，从消息表中读取。
- 日志表表名为MQ_LOG，包含的字段：日志ID、发布者信息、发布时间、队列名称、消息内容。
- 消息表表名就是队列名称，包含的字段：消息ID（递增生成）、消息内容、消息发布时间、消息发布者。
- 日志表需要及时清除已经写入消息表的日志数据，消息表最多保存30天的消息数据。

2. 细化设计点2：数据如何复制？

直接采用MySQL主从复制即可，只复制消息存储表，不复制日志表。

3. 细化设计点3：主备服务器如何倒换？

采用ZooKeeper来做主备决策，主备服务器都连接到ZooKeeper建立自己的节点，主服务器的路径规则为“/MQ/server/分区编号/master”，备机为“/MQ/server/分区编号/slave”，节点类型为EPHEMERAL。

备机监听主机的节点消息，当发现主服务器节点断连后，备服务器修改自己的状态，对外提供消息读取服务。

4. 细化设计点4：业务服务器如何写入消息？

- 消息队列系统设计两个角色：生产者和消费者，每个角色都有唯一的名称。
- 消息队列系统提供SDK供各业务系统调用，SDK从配置中读取所有消息队列系统的服务器信息，SDK采取轮询算法发起消息写入请求给主服务器。如果某个主服务器无响应或者返回错误，SDK将发起请求发送到下一台服务器。

5. 细化设计点5：业务服务器如何读取消息？

- 消息队列系统提供SDK供各业务系统调用，SDK从配置中读取所有消息队列系统的服务器信息，轮流向所有服务器发起消息读取请求。
- 消息队列服务器需要记录每个消费者的消费状态，即当前消费者已经读取到了哪条消息，当收到消息读取请求时，返回下一条未被读取的消息给消费者。

6. 细化设计点6：业务服务器和消息队列服务器之间的通信协议如何设计？

考虑到消息队列系统后续可能会对多种不同编程语言编写的系统，为了提升兼容性，传输协议用TCP，数据格式为ProtocolBuffer。

当然还有更多设计细节就不再一一列举，因此这还不是一个完整的设计方案，我希望可以通过这些具体实例来说明细化方案具体如何做。

小结

今天我为你讲了架构设计流程的第四步骤：详细方案设计，并且基于模拟的“前浪微博”消息队列系统，给出了具体的详细设计示例，希望对你有帮助。这个示例并不完整，有兴趣的同学可以再详细思考一下还有哪些细节可以继续完善。

这就是今天的全部内容，留一道思考题给你吧，你见过“PPT架构师”么？他们一般都具备什么特点？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。（编辑乱入：精彩的留言有机会获得丰厚福利哦！）

更多一手资源请添加QQ/微信1182316662



正是那朵玫瑰

2018-05-28

可以完善的细节：
1、发送端和消费端如何寻址
利用zookeeper做注册中心，把broker的地址注册到zk上，发送端和消费端只要配置注册中心的地址即可获取集群所以broker地址，当有broker下线时，发送端和消费端能及时更新broker地址。
2、发送端消息重试
当发送消息发生网络异常时（不包括超时异常），可以重新选择下一台broker来重试发送，重试策略可以自定义。
3、消息消费采用pull还是push？
考虑push模式会更复杂，故放弃，采用pull模式，消费端主动去拉。为了达到与push模式相同的低延迟效果，可以采用长轮询的方式，消费端轮询拉取消息费，当有消费可消费时，返回消息，如果没有可消费的消息，挂起当前线程，直到超时或者有可消费的消息为止。
4、消息重复问题
消息中间件不解决消息重复的问题，有业务系统自己根据业务的唯一id去重。
5、顺序消息
发送端在发生顺序消息时，只发送到相同broker的相同队列，消费端消费时，顺序消息只能由同一个消费端消息。
6、定时消息
发送端指定消息延时多长时间消费，broker端定时扫描定时消息，达到延时时间的消息加入到消费队列。
7、事务消息
发送端分两步，先预发送消息，broker端只记录消息为预发送状态，再执行本地事务，然后再根据本地事务的成功或者失败发送确认消息（回滚还是提交），这一步如果发生异常，broker启动定时任务，把未确认的消息发送给发送端回查事务状态（需要发送端提供回查接口）。

目前就想到这么多。

作者回复

2018-05-28

厉害，基本上重点都涵盖了

稳健的少年

2018-05-26

PPT架构师以脱离一线时间较久的领导居多吧。很多技术他们没有真正使用过，只是从各种途径得知很强大，其他公司（BAT）都在用，于是就在PPT中写入这种技术。缺点就是很容易做出貌似可行，深究其细节全是坑的设计。

作者回复

2018-05-27

BAT三个字是设计捷径，但也很多坑

ant

2018-05-26

很有幸我们现在的架构师就是PPT架构师，我觉得他的优点就是懂了很多的概念，能说话到，可以忽悠住老板。缺点也很明显，就是他知道的都不是很深，比如曾经我们的搜索引擎原型，他并不能说出ES和solr的优点（当然我也不知道，平时用solr多点），最后我们选了ES，他给的原因就是朋友说的ES比solr好，后面搜索这里就交给我来搞了。我们是互联网项目，在重构的项目的时候他选择了jpa，这就导致变化需求的时候，查询这块比较麻烦，不灵活。
其实就像前面说的，每个技术存在就是合理的，只是每个有每个技术的使用点，架构师应该对常见的技术栈原理非常清楚，知道什么时候应该使用什么技术。
我理解的PPT架构师的特点就是知识点多，知道概念，能虎住老板，缺点就是技术栈不咋滴，特别是细节上。我觉得架构师应该帮助员工成长，而不是遇到问题就说这个问题我遇到过，你上网搜索下解决方案。
初次留言，欢迎板砖

作者回复

2018-05-27

架构师确实需要技术面宽，但别只知道技术名词，基本使用，关键原理，优缺点都需要了解

蜗牛

2018-05-26

就一些新的技术引入，架构师需要做哪些技术验证，或者研究到什么深度以后，才认为该技术适合呢？

作者回复

2018-05-26

基本原理，优缺点，关键设计点，架构师至少要安装过，编写demo体验过，确定选型后，要进行性能和可用性测试例如es的索引设计就是关键设计点

bluefantasy

2018-05-28

请教华仔：您说的“日志表是尾部追加，性能高”，这个日志表是用myisam存储引擎吗？我刚刚查了文档innodb只有系统表空间和日志文件是序列化写呀，普通表文件都是随机写。请指教。

作者回复

2018-05-29

这个日志表是我们自己设计的，不管是innodb还是myisam都可以，我说的尾部追加是指我们只会往日志表插入数据，类似kafka的文件尾部追加一样

Jolie Liang

2018-05-26

对于PPT架构师，总结有以下几个特点：
1）整体思路附庸
2）不能准确定位业务需要解决的痛点
3）对所需技术的原理解释不深

更多一手资源请添加QQ/微信1182316662

4) 执行到后期，返工及补坑的工作比较多 …… 作者回复	2018-05-27
PPT可以造火箭，实际实现只能造鞭炮	
燃烧的阿布	2018-05-26
业务系统发布消息时，首先写入到日志表，日志表写入成功就代表消息写入成功；后台线程再从日志表中读取消息写入记录，将消息内容写入到消息表中。 业务系统读取消息时，从消息表中读取。	
这里的设计是不是冗余了？？ 作者回复	2018-05-27
日志表是尾部追加，性能高	
空档滑行	2018-05-26
真见过PPT架构师，1.自己基本不写代码， 2.对某一项技术比较精通所以架构设计时尽量往这上面靠，比如对mysql很熟悉所以设计出的系统大量用到mysql 存储过程。 3.设计出来的架构完全不考虑老系统兼容或者迁移难度 作者回复	2018-05-27
架构师手里有一把锤子，然后所有的问题都是钉子◆◆	
李志博	2018-05-31
认识一个工作10多年的架构师，天天只会说代码命名和注释的问题，从来没谈过什么高大上的架构，也从来没见过他写过代码，有一次我来他来帮我看一个问题，他装没听见，走了…… 作者回复	2018-06-01
你们公司还要架构师么◆◆◆◆	
我的名字叫浩仔、	2018-05-30
我们的架构师连PPT都懒得做。	
crazyone	2018-05-29
华哥，“日志表是尾部追加，性能高”，这个具体实施细节能否讲解下。 作者回复	2018-05-29
其实高性能的很多存储方案都是这样设计的，MySQL有Binlog，HBase有HLog，道理都是相通的。 在这个备选方案中，我们设计一个日志表，假设名称叫MQ_LOG，包含ID, time, queueName, message, publisher等几个字段，每次收到消息发布请求时先写这个表，每次都是表尾部追加。 如果不用自己设计的日志表，mysql的binlog也类似尾部追加，性能也不错，缺点就是没法自己灵活实现各种刷盘机制了。	
Fenlon	2018-06-15
一楼事物消息 还有种方案就是本地事物解决方案。 消息和业务一块存在本地，只要保证本地消息能被发到broker就好了	
Sylar.	2018-05-28
我也没有明白作者说的日志表尾部追加性能好，具体是什么操作，还望赐教，多谢了！ dilei	2018-05-28
看来ppt架构师还是很多的啊，应该推荐他们来看看老师的文章o(^o^)o	
Garwen	2018-05-28
老师，您讲的详细设计示例特别好，我非常想看到一些比较完整的详细设计文稿，不知可否有推荐。 作者回复	2018-05-28
可以自己尝试，我没有这几个备选方案的详细文档，内部的方案设计和这个方案不同，而且资料也不能泄露，望谅	
itperson	2018-05-27
这种架构师 不了解技术细节 只了解大体使用哪种技术 技术有哪些限制或者坑了解的不透彻 这样在细化阶段会产生很多问题 这种架构师忽悠领导还凑活 落地时大多就会出状况 由此我想问 如何快速提升架构硬实力 尤其是当下技术发展迅速 人的精力有限 如何更好的进行技术了解与选型	
卡莫拉内西	2018-05-27
ppt架构师共同的特点是： 1. 方案设计大多从网上照抄 2.回避别人提出的技术细节问题 3.口头禅：别人都是这么用的 4.只会众多技术名称的单词拼写，而且还经常在交谈中时不时的说出一两个 作者回复	2018-05-27
细节不讨论，哈哈哈◆◆◆◆	
Hook	2018-05-27
请老师指导下： 同一个队列的消息数据是不是被分散到了不同的分区上？此时假如要考虑消息顺序性的话，是不是就不满足了？ 作者回复	2018-05-27
是的，保证消息顺序很复杂	
公众号：歪脖贰点零	

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

架构师对PPT的出发点，和PPT面，如何体系化理清，这就是技术不仅要有深度的问题，平常说打杂也不为过，什么都要懂，但不一定都得上手做，要能指引他人按着蓝图去实施，当然碰到棘手问题，还是要深入进去解决。	2018-05-27
架构师是体现了一个团队的技术强度，PPT架构师有时候还是需要做一做，但不能只停留在PPT里，落地实战同样不能逊色。	
作者回复	
我的习惯是最好安装运行，然后写demo体验一下，单纯看文档理解还是不够	2018-05-27
行用	
	2018-05-27
现在很多架构师只关注主流程，大方向，根本不关注细节。导致做出来的东西用户体验不好	
作者回复	
	2018-05-27
两者需要兼顾，把握平衡	
Tom	
	2018-05-26
怎么保证日志表所有数据都写入消息表？日志表是不是要加状态字段，后台线程写入消息表后更新状态值。	
作者回复	
	2018-05-27
细节可以有很多实现方式，限于篇幅无法展开，详细的设计文档有几十页	
蔡	
	2018-07-15
基本原理，优点缺点，关键技术点。亲身体验下	
Enjoystudy	
	2018-06-28
补充几点: 1.应该确保同一个分区只有一个消费者，否则消费的时候还得解决多个消费者拉到相同的消息，而且这块应该做到一个负载均衡，多个分区应该分给不同的消费者 2.队列应该做个优先级存储，也就是日志表应该有多个 3.应该有个ack的机制，否则没法做重试 4.应该支持延时队列 5.消息的状态以及下一条消费的游标要区分不同的group	
轩辕十四	
	2018-06-27
日志表尾部写入的细节还没理解透。请问如果没有日志表，直接写各消息表，不也是尾部追加吗？这里是否涉及MYDQL写文件的细节？	
作者回复	
	2018-06-28
各个消息表在磁盘上存储的位置不同，会导致磁盘寻址时间较长	
jasudd	
	2018-06-19
请问老师，架构设计和技术方案设计的界限是什么？感觉架构师很详细的设计完，开发人员只是功能实现会没有成就感	
作者回复	
	2018-06-21
简单来说，架构师说用单进程多线程的网络模型，开发人员要具体实现这个模型，里面细节还是很多的	
vintin	
	2018-06-15
MySQL中为什么要设计一张日志表呢？直接根据队列表名称来路由写入到对应的消息分表性能岂不是更高？这样省去了从日志表到消息表的中间环节。如果是为了读写分离，转移线程在写消息表的时候业务也在拉消息，也没有避免。请华仔指教	
作者回复	
	2018-06-15
日志表是尾部追加，写入性能高	
LouisLimTJ	
	2018-06-10
这可能离题了，主要是管理问题。经过内部的评审机制，阿里的P9还存在不PPT 架构师？	
SHLOMA	
	2018-06-04
李老师有没有可以开源的架构设计文档之类的资料，见识见识大厂出品	
作者回复	
	2018-06-06
这是公司机密资料，如果你要看开源的，推荐kafka，es，disruptor的技术资料	
Mars	
	2018-06-04
华哥，这个专栏讲解架构设计的概念、设计原则、设计规则等主谓宾语理论上的设计，对很多没有亲身经历过架构设计的程序猿又想踏入架构师之旅，文章阅读之后文中讲的理论基本也都能懂，但要自己真正落地恐怕非常难？要是这个专栏之后华哥能再写一个专栏，就针对“前浪微博”的消息队列从设计前期、详细设计、框架搭建、开发实现、最后测试部署的一套流程的专栏。让我们亲自动手感受从系统设计到最终系统上线的一套完整流程。相信会受很多人欢迎及订阅。	
Geek_8242cb	
	2018-06-02
多个消费端并发的取消息，如何保证消息的顺序性？当前上一个消息的offset需要保存在zookeeper 中吗？	
作者回复	
	2018-06-02
需要的，可以用zookeeper，也可以用数据库，redis等，就看你要求多高的性能	
c@ini@o	
	2018-06-02
传统企业及提供服务的架构师就是PPT架构师，这个没啥办法，因为和传统企业的组织机构对应。我认为应该提供一个平衡的方法来衔接。按目前分享的内容肯定是说服不了现有组织接受设计的。	
allwmh	
	2018-06-01
采用mysql存储 读取消息 的话 应该不需要状态了吧？	
作者回复	
	2018-06-01
也需要记录消费信息	

更多一手资源请添加QQ/微信1182316662

更多一手资源请添加QQ/微信1182316662

小龙

架构师难道把框架搭起来吗？还是只是以文档形式把方案写出来，团队根据文档搭框架与代码？现在的专栏技术都是讲解互联网电商行业的技术及架构，您专栏讲的架构知识是否适用于工业领域？比如自动化码头调度监控所有自动化设备的软件系统。

作者回复

1. 架构师不需要搭框架，但要保证方案可行
2. 这套架构理论不局限于互联网软件系统，各行业都可以适应

成功

售前架构师也是需要，作者大人估计不怎么打标或推新

作者回复

以前在菊花厂答过标，转移动互联网就没做过了

正直O令狐勇

看了评论学习到很多，还有不解的地方。如何记录消费者的消费状态，比如正在处理哪些消息，成功处理了哪些消息，哪些消息处理失败了。这些保存在消费者端好，还是在消息队列系统中保存。

作者回复

是的，限于篇幅无法完整展开，你正好可以根据课程内容，针对这个问题来练练手

孙晓明

在做架构设计时，为了使甲方觉得系统规模、技术与合同成本是对等的，不得不进行复杂的架构设计。有的技术可能根本用不到或无需用到，但都要写到架构设计中，什么技术新这什么，什么技术看起来高大上这什么。如果不这样甲方可能觉得钱花多了，这个怎么破？

作者回复

通常情况下，这些要求算架构设计的约束，只能遵守，不能选择，架构师的任务是如何以最小成本来满足甲方需求，这也是复杂度之一

fuhuoyeyou

我一般用户session是不是单独存储呀？如果直接放在业务服务器上，是不是有很多局限性？

作者回复

可以持久化或者独立存储，也可以直接放业务服务器，没了就重新生成，看你的业务场景下用户的接受程度了

钰涵

随着做t时间久了，发现好架构真的太少了，上得了厅堂下得了厨房，我们开发团队挺大的，每个团队最多也就培养出一两个人，找个替代很难。

作者回复

好的架构师比好的程序员难找多了

王磊

设计上是考虑一致性问题，即连接到不同业务节点，他们连到不同数据库节点，看到的信息是否可能不同？不知道mysql集群是怎么处理的，例如只写入一个数据库节点就返回？是否需要只有master负责写，所有节点可以读？

作者回复

master负责读写，slave只在master故障的情况下提供读操作

刘应涛

发消息时，消息写入日志表，然后通过canal把日志表的binlog导出，处理线程去监听binlog事件去写消息表如何？当然需要引入Kafka

作者回复

不需要导出binlog，直接从日志表读取记录就可以

YMF_WX1981

应该共事过，ppt形式完美好看，没人能复制，包括美工。

作者回复

然而你来实现就有很多坑等着你填

曹铮

我理解的ppt架构师和大家不同，我理解的ppt架构师往往类似咨询公司的那种，要充分熟悉领域业务，有成熟方案经验，只要方案成功过就行，至于怎么落地就是你的事了。这种会有市场，主要还是对某些领域的整体理解比较熟。
今天这个方案，有一个日志消息表的目的，是为了把消费和写入操作带来的压力分开嘛？考虑到写和读的比例是11比10，所以日志表每个库只有一个？
文章似乎没提具体如何记录消费者读取消息位置，这也是个关键trade off的点，决定了qos。下期会说？
另外有个吹毛求疵的想法，自研发sdk往往比看起来的难，需要设计功力，有bug了如何及时分发新版本都是问题。选型时需要开发sdk的都是减分项。怎么看这都是消息队列擅长的场景，使用成熟三方实现成本会非常非常低，不能和运维大哥商量一下么？_-_-

作者回复

1. 你说的一般叫系统分析师
2. 日志表是尾部追加，写入性能高，日志表一个库只有一个
3. sdk写稳定确实需要一段时间
4. 运维大哥不怕每天维护，最怕出问题几个小时甚至几天都搞不定

木木

大佬，我以前是个系统集成的售前，没有研发背景，想学习架构，有好的路线推荐么？◆◆

作者回复

售前技术支持的架构一般称为业务架构，和软件的技术架构不太一样，如果你想转行编程的话，我的专栏就是最好的入门，以后还需要自己在工作中去锻炼。

黄金的太圆

更多一手资源请添加QQ/微信1182316662

2018-05-26

可以继续迭代的设计点思考:
1.消息写入/读取SDK, 支持阻塞/非阻塞模式调用两种, 非阻塞模式还需设计回调通知机制
2.消息写入/读取的状态反馈机制, 是否要支持消息的事务机制, 由于存在生产者->消息服务器->数据库三个环节, 至少两次通信, 消息存储功能的高可用设计, 存储状态如何反馈至生产者, 以及失败后的处理机制设计(消费者模式同理)

目前就想到这么多

作者回复

2018-05-27

集群路由也是重点

更多一手资源请添加QQ/微信1182316662