



# DTCC

## 2016中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2016

数据定义未来

SequeMedia  
盛拓传媒

IT168.com

ChinaUnix

ITPUB

# GemFire如何应对电商或移动APP平台的高负载挑战



**DTCC**

**2016年中国数据库技术大会**  
DATABASE TECHNOLOGY CONFERENCE CHINA 2015

SequeMedia  
数据传媒

IT168

ChinaUnix

mpub

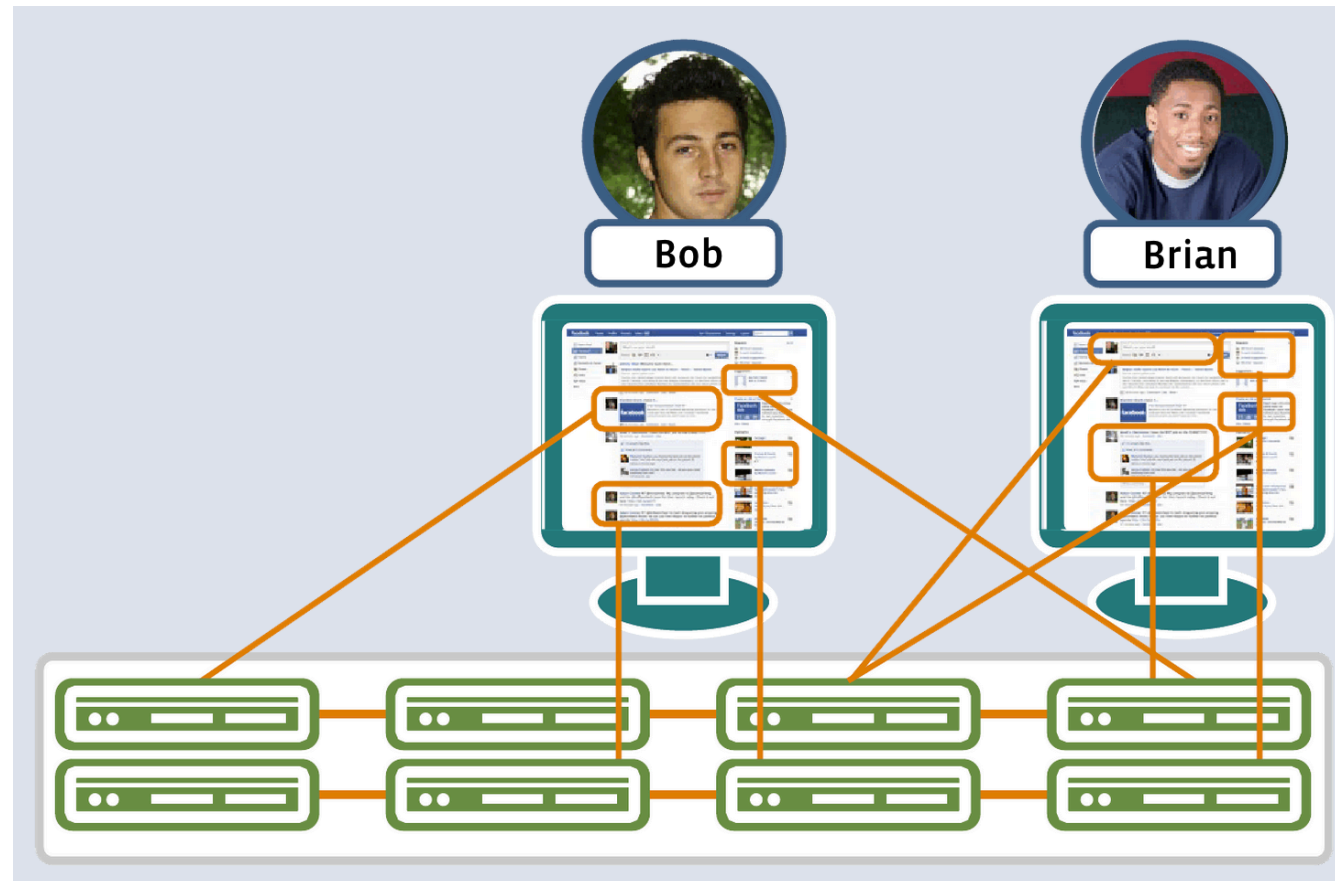
# 议题

---

- 拆分应用互联数据
- Web Server与GemFire进行集成
- Memcached、Redis与GemFire进行集成
- GemFire利用HDFS进行日志保存和处理

# 大规模扩展分布式缓存系统

借用FaceBook  
的架构方式，  
将页面的访问  
进行拆分，保  
存数据到多个  
缓存集群中



# 应用间互联数据

---

- 拆分主页和其他页面的点击访问
- 数据不保存在单台服务器上，保存于多台服务器集群中
- 跨所有集群的服务器快速拉取数据到前端页面

# 应用大并发查询数据

---

- 为了在数据并发操作的情况下获得良好的性能，需要并行地转发get请求
- 切换到异步 I/O 进行访问
- 不同的Object有不同的大小和访问方式。构建memcached pools拆分不同的objects类型，达到高效的内存利用率

# 解决前端连接拥塞问题

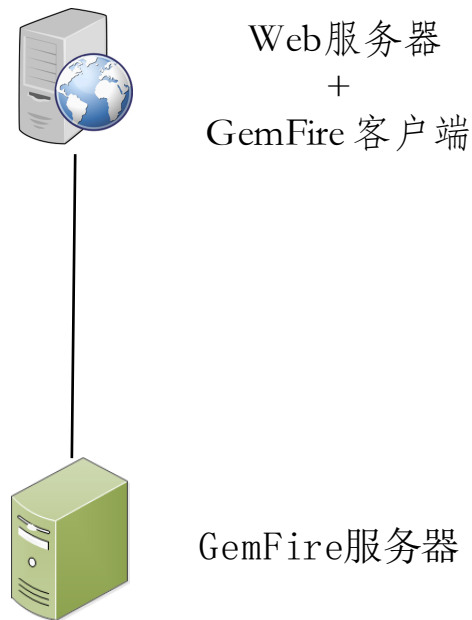
---

- web服务器连接到memcache服务器
- 每个web 服务器运行50-100个进程
- 每个memcache有100K+ TCP 连接(UDP 能够减少连接数量)
- 使用gzcompress压缩序列化的字符串

# Web Server 与 GemFire 集成

---

1. GemFire采用C/S架构与Web Server进行交互，将GemFire客户端嵌入到Web Server中，Web App 从GemFire客户端读写数据，然后客户端将读写操作同步给GemFire Server

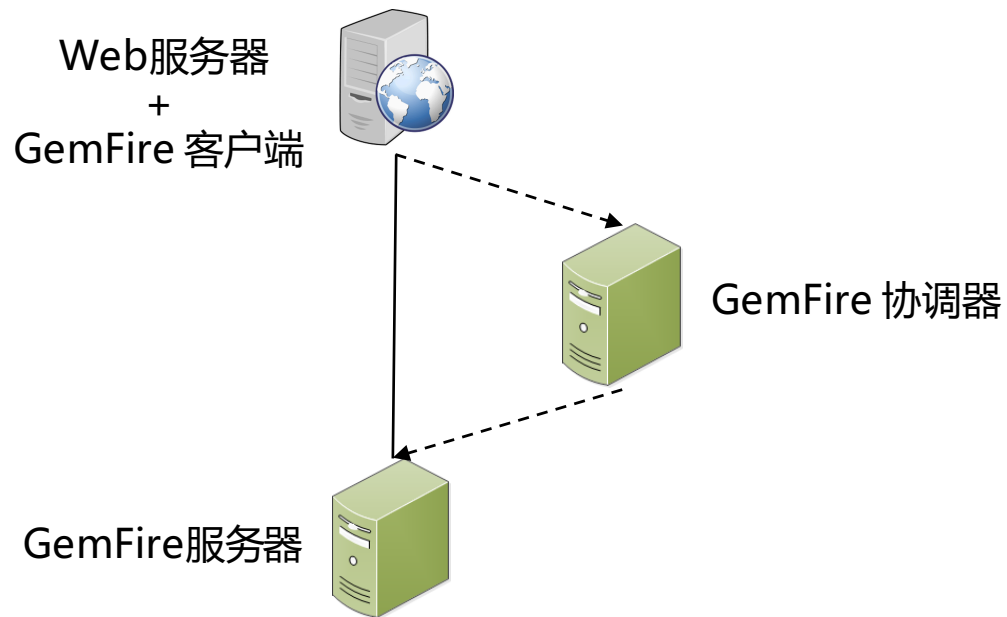




# Web Server 与 GemFire 集成

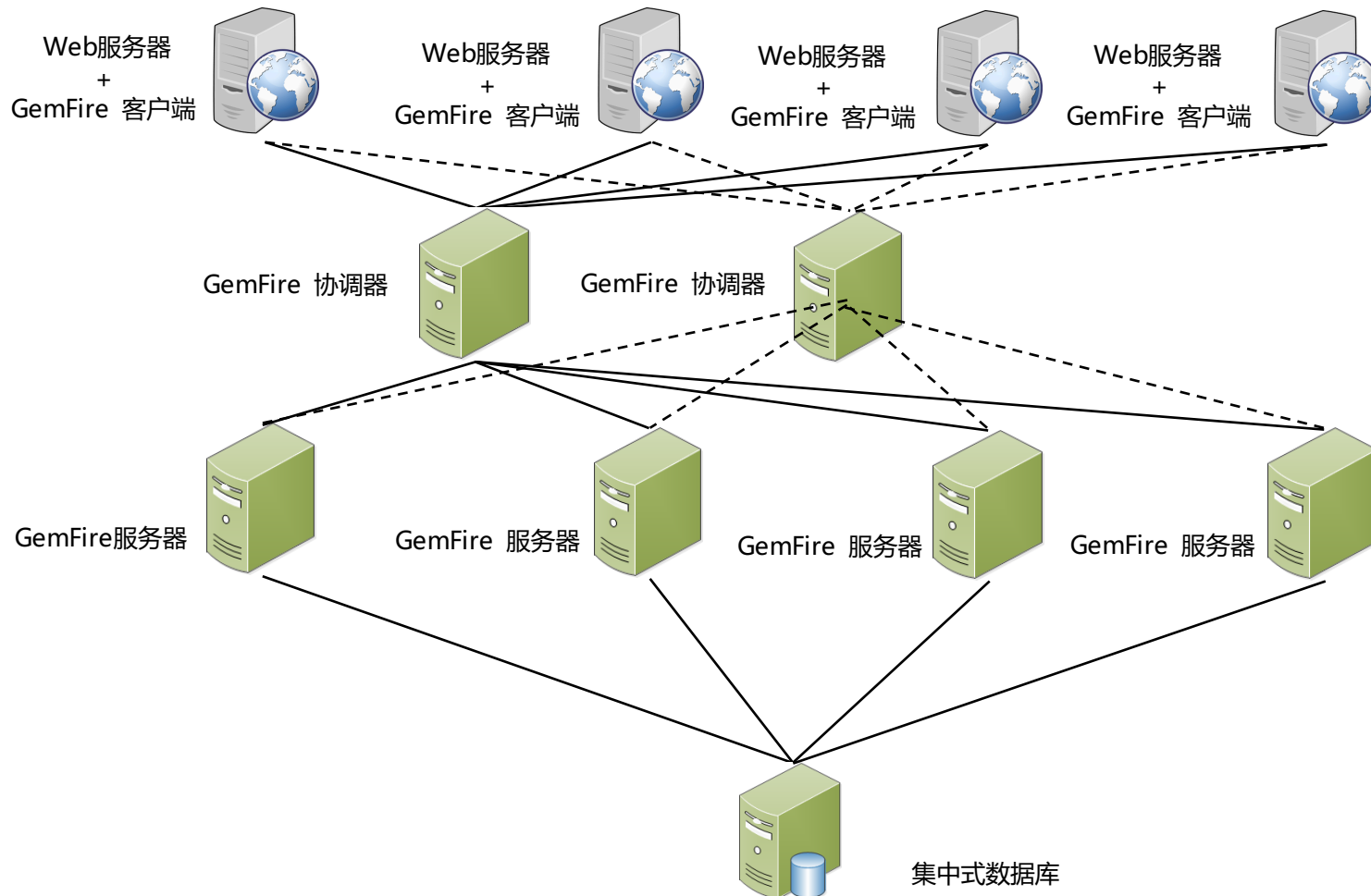
---

2. GemFire 客户端通过协调器来连接合适的GemFire服务器，当建立连接之后，客户端便可以直接与服务器进行交互



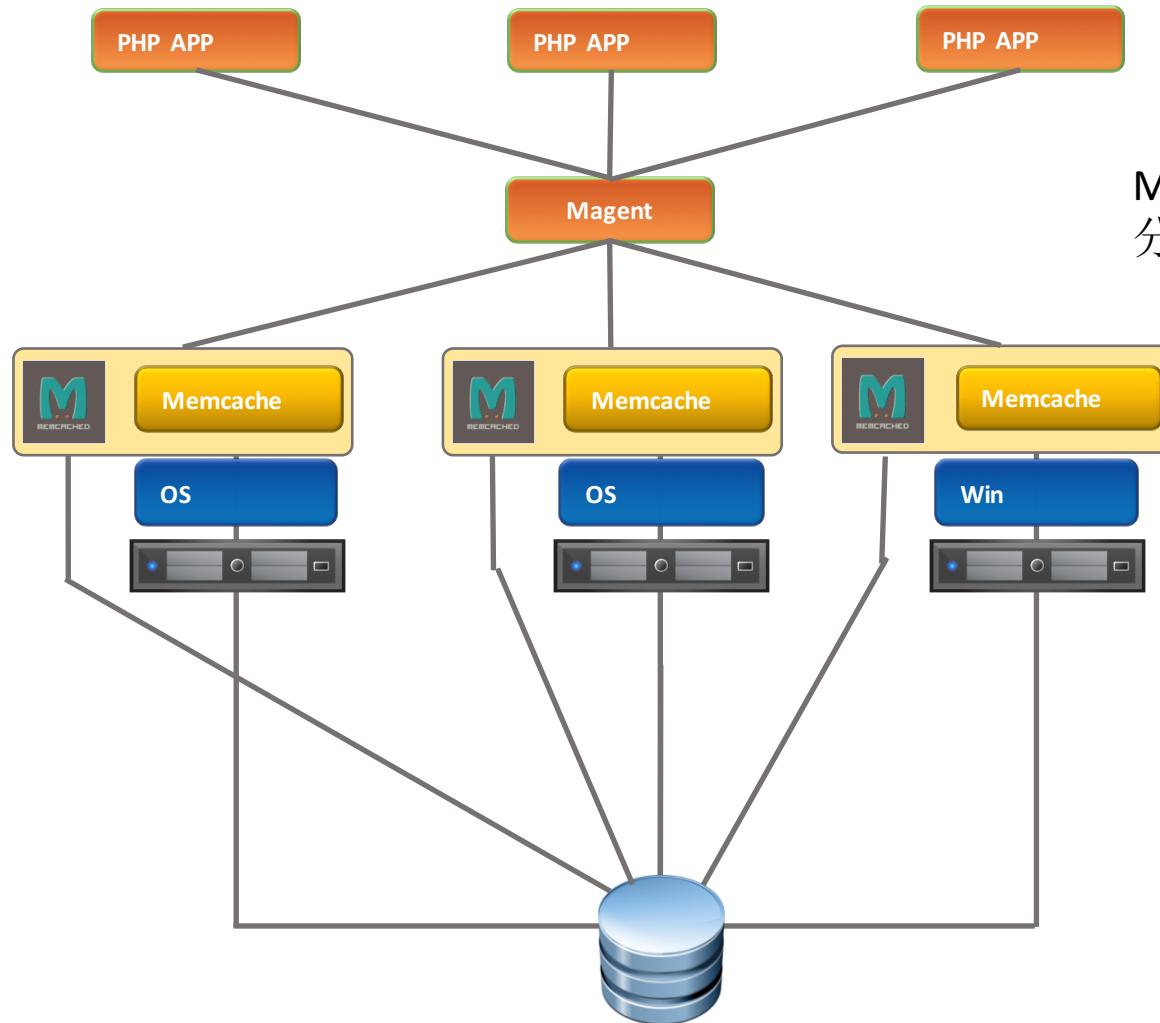
# Web Server 与 GemFire 集成

3. GemFire 协调器采用HA高可用架构，防止协调器宕机导致客户端与服务器之间连接出现故障，GemFire集群整体架构如下：



# Memcached大规模集群扩展

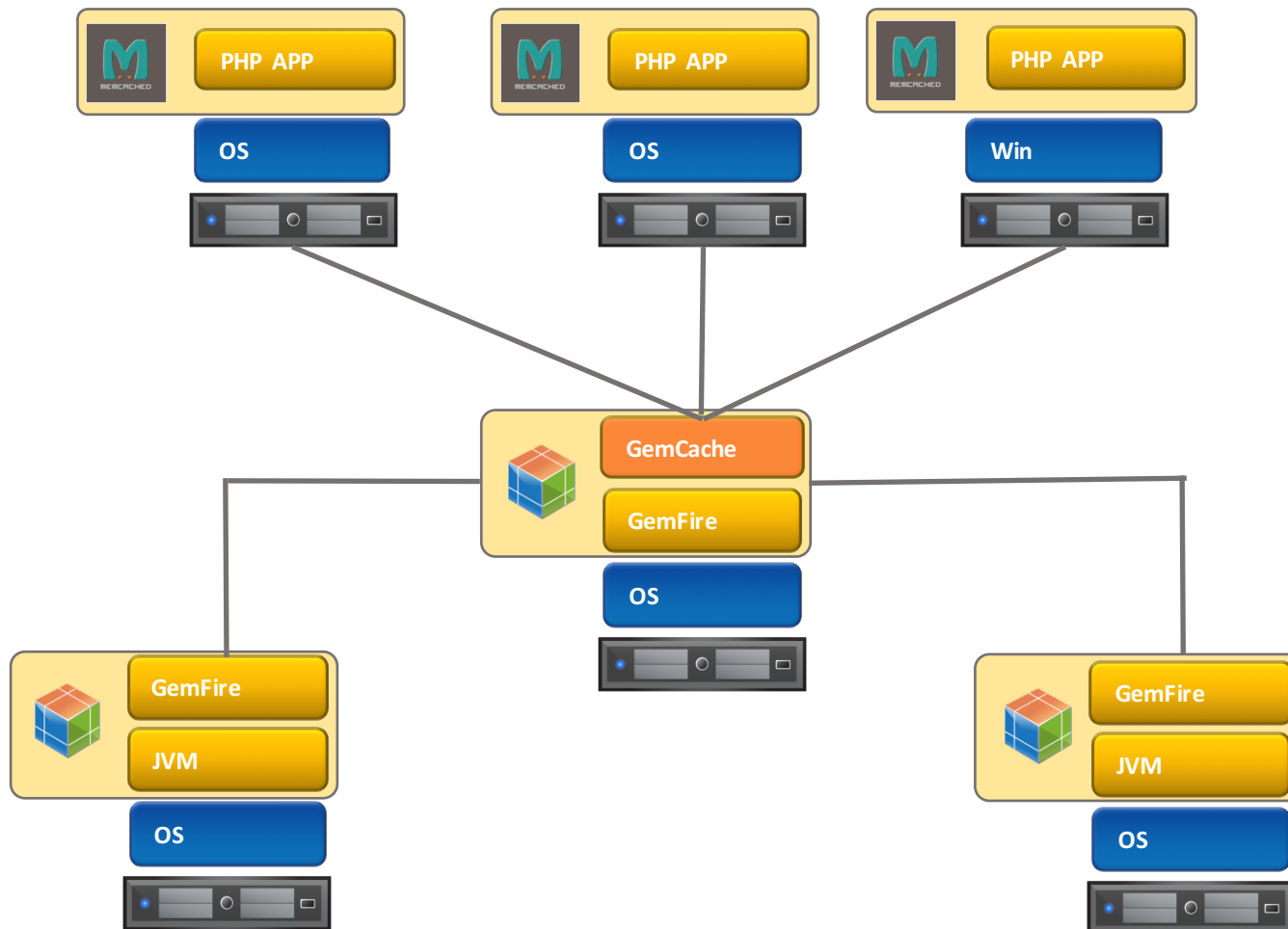
- 传统memcache分布式集群主要靠客户端程序库实现，“服务端”没有分布式
- 当memcache集群环境发生变化时，如加入、离开节点会严重影响缓存的命中率



Magent使用一致性哈希  
分配值到memcache节点

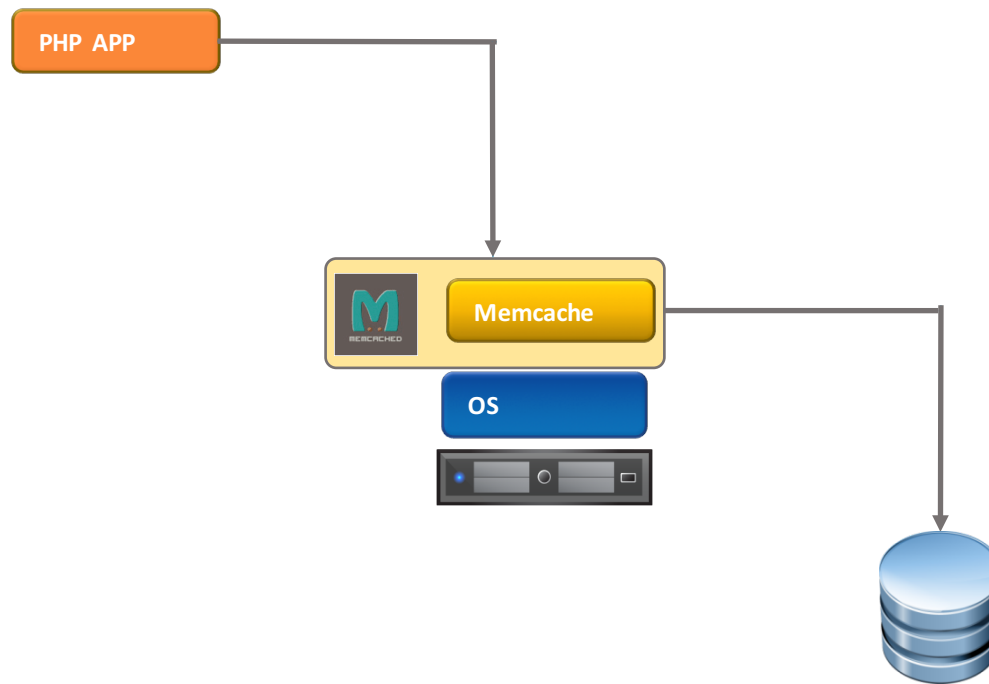
# Memcached大规模集群扩展

- **GemFire 嵌入 Gemcache**服务器与memcache客户端进行交互，将形成与memcache类似的动态分布式缓存集群，实现memcache应用 与 GemFire联动



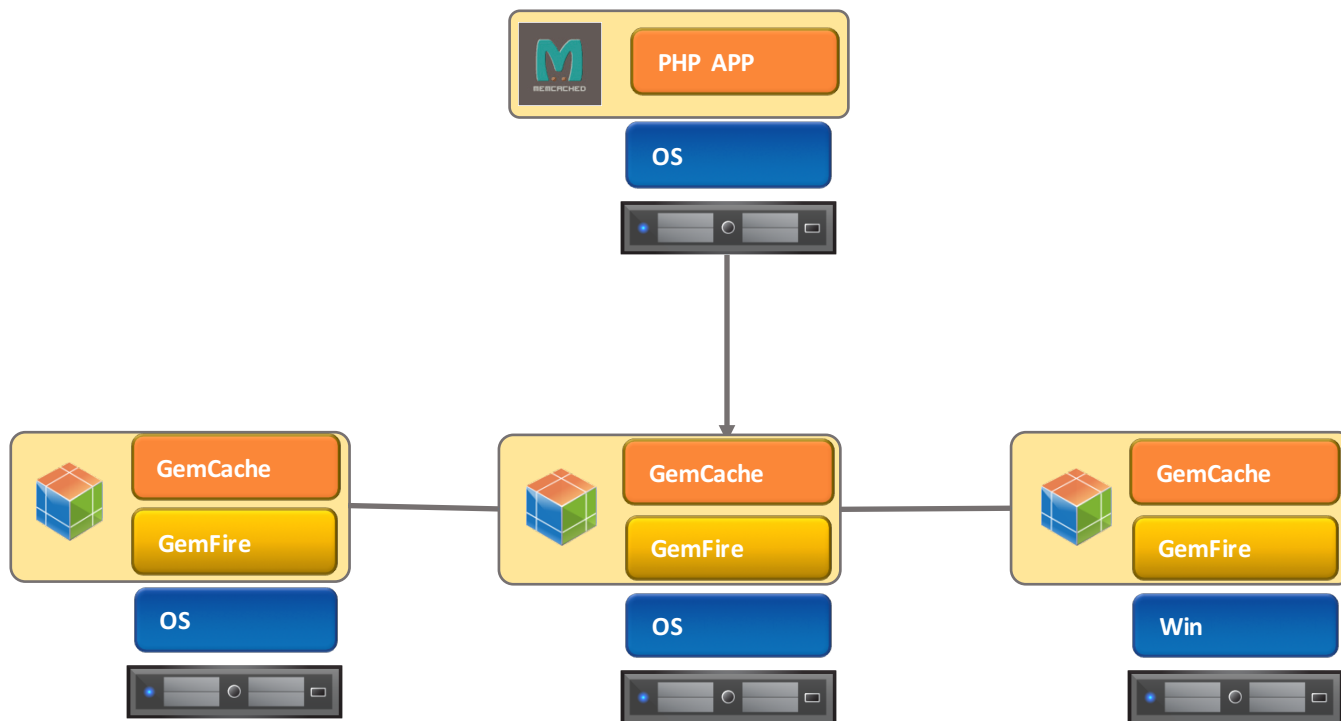
# Memcached迁移至GemFire

- 迁移动机: memcached 最根本的问题在与它仅支持“cache-aside”，应用既负责更新缓存和也负责更新数据库。其结果是：
  - 会导致缓存和DB数据出现不一致的风险
  - 破坏每个应用的业务逻辑



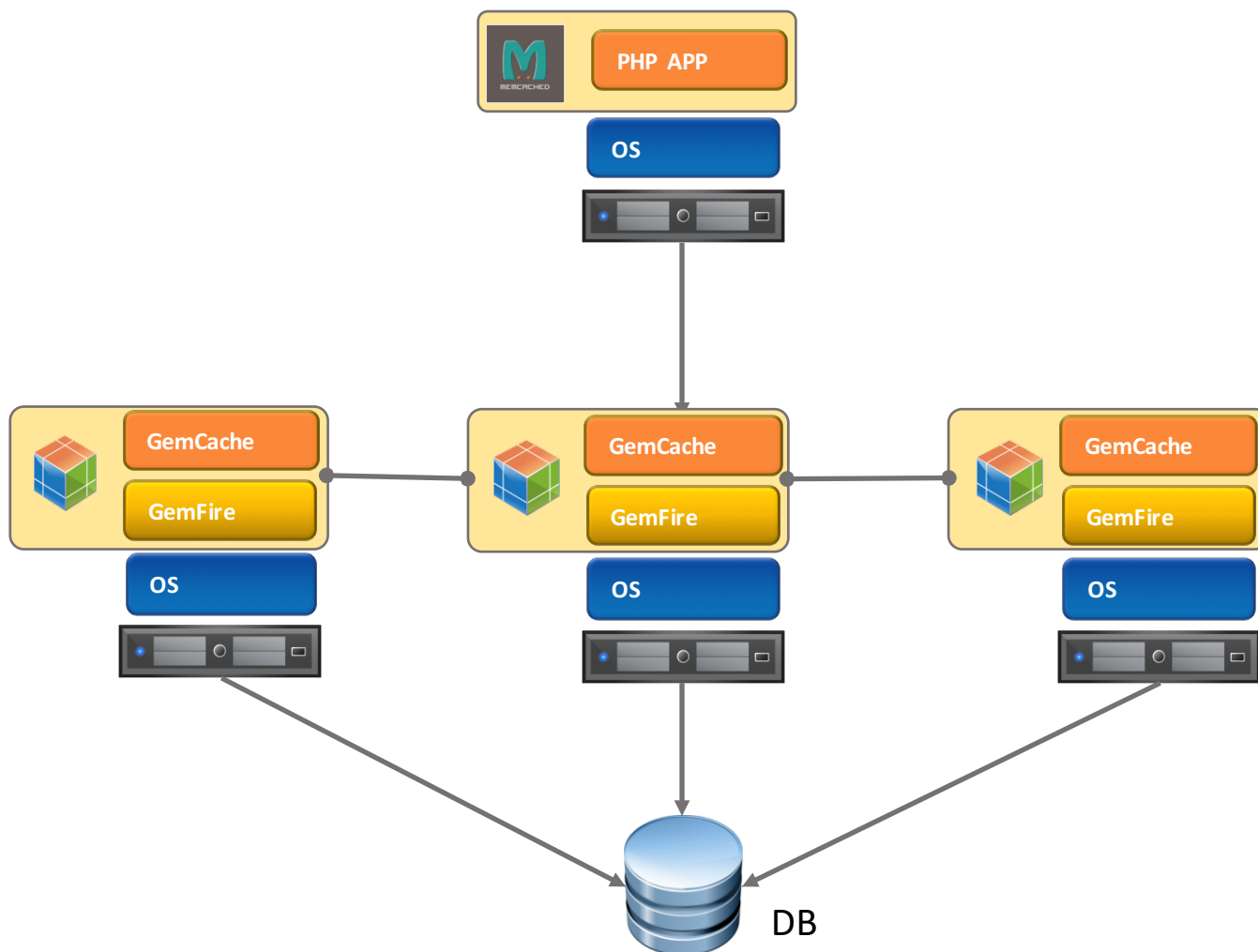
# 有效避免惊群效应

- 目前大多互联网应用前端内容更新地非常快。当这台服务器出现故障后，所有的客户端请求服务器将得到缓存丢失错误，那么所有的客户端都向数据库访问数据，数据库有被瞬间压瘫的风险。



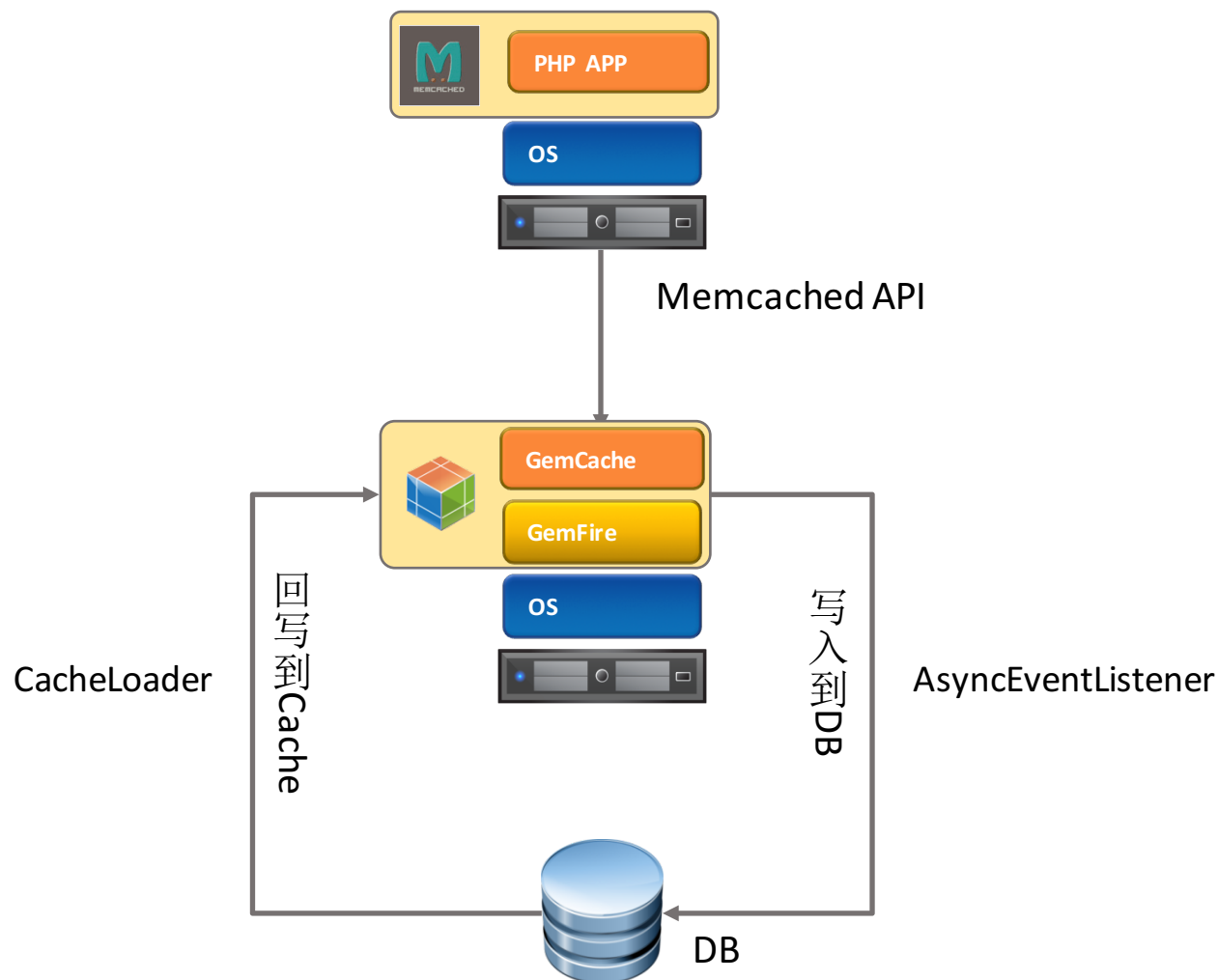
# 有效避免惊群效应

- Memcached 客户端使用 Memcached API 连接 GemCache，实现读写 GemFire 中的数据，维护着一个 GemCache 服务器的列表。



# 有效避免惊群效应

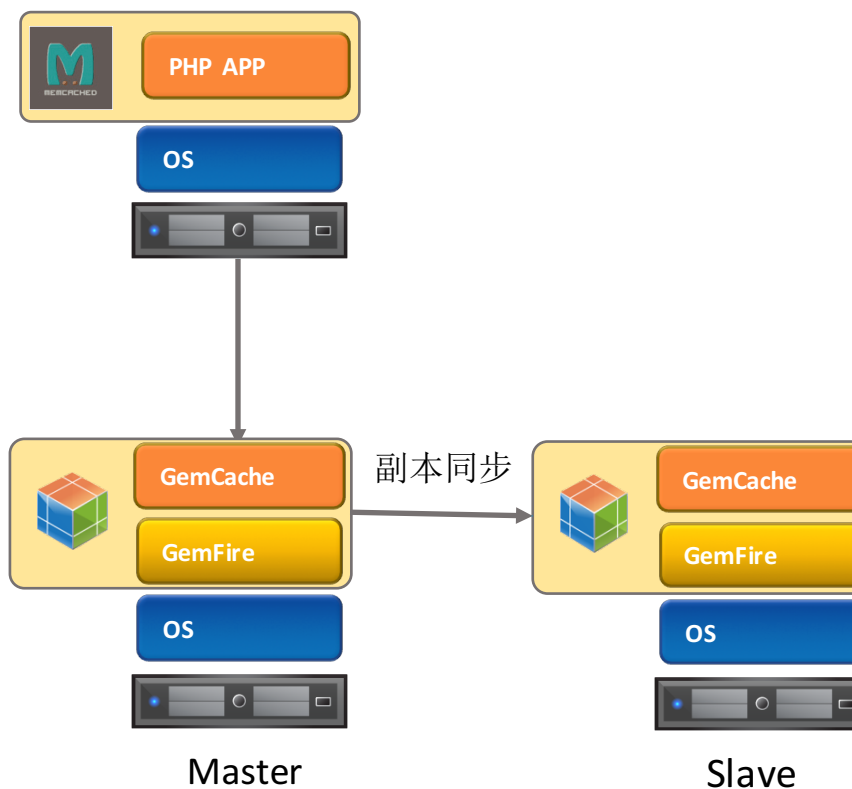
- 前端应用不再直接与DB交互，简化了应用代码。所有的数据库都通过GemFire来读写数据。为了从DB中读取数据，你能够使用GemFire CacheLoader，同时可以使用AsyncEventListener将数据写回到DB中。





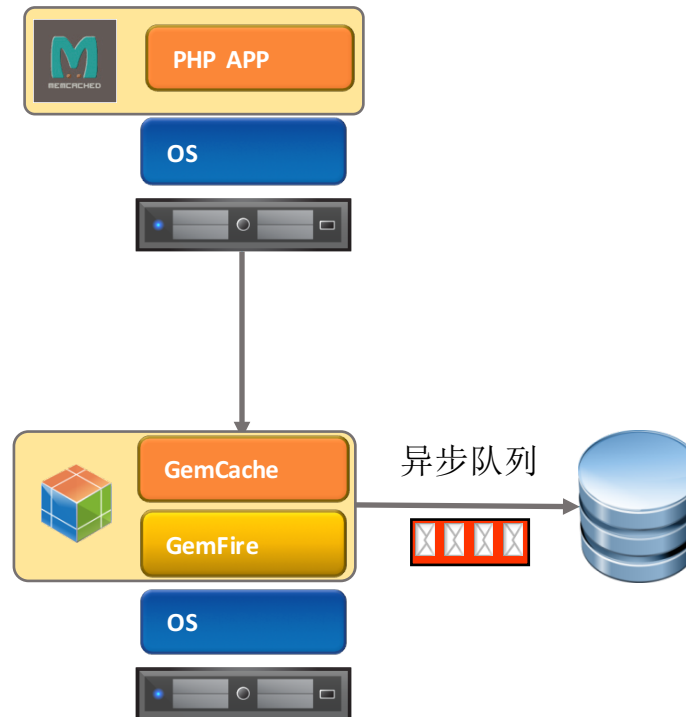
# 有效避免惊群效应

- 客户端只写到GemFire，当写入完成时，GemFire将保障写入已经同步到了冗余的副本。在同步到冗余副本之前，即使主 bucket 死掉，其他客户端和数据库都不会看到数据更新。



# 数据一致性保证

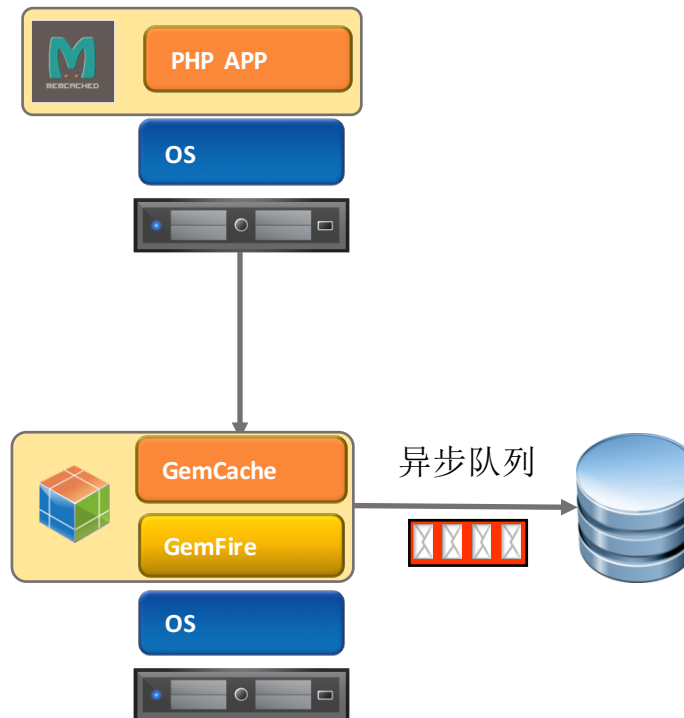
- 来自客户端的所有数据更新最初先进入到GemFire，然后持久化到DB，它们经常是一致的。所有更新进入DB，即时在这种情况下GemFire节点出现故障失效，AsyncEventListener队列将同步，保证数据也是一致的。



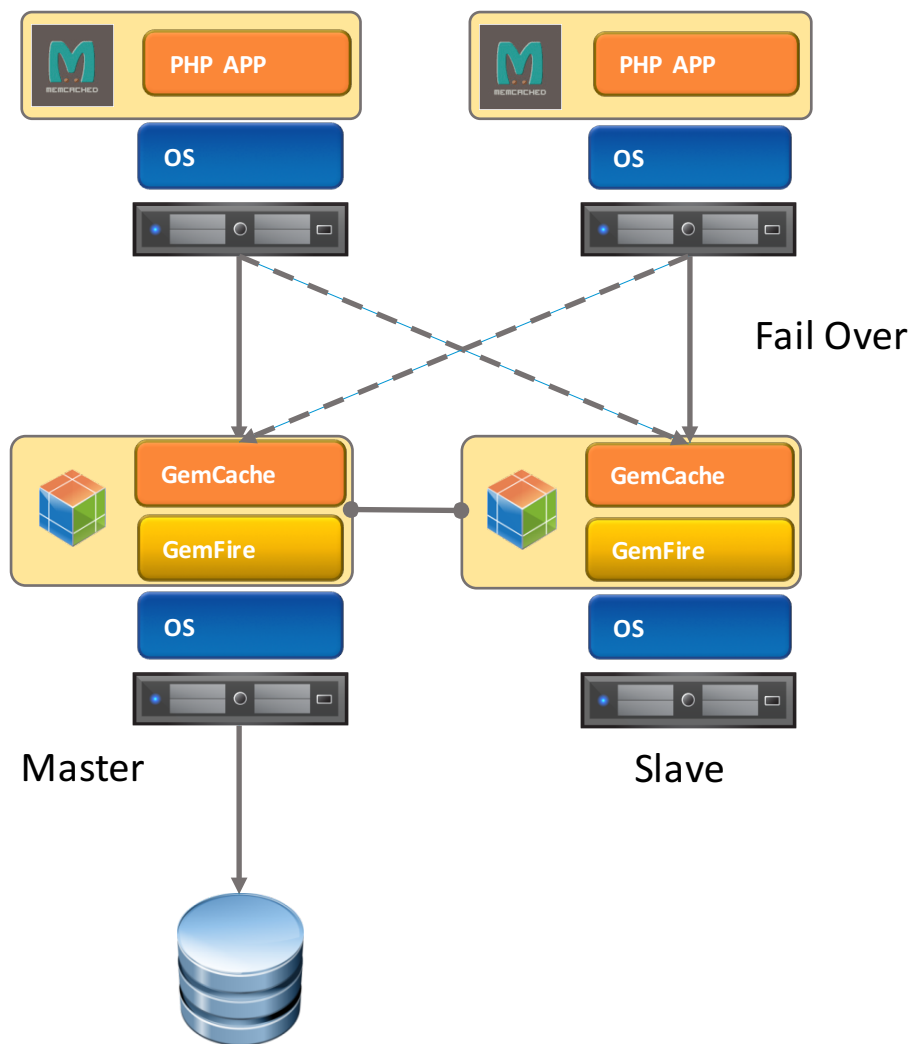
# 交互协议配置

- Memcache 客户端能够使用两种协议与GemFire 服务器进行通信，客户端应用不需要改动任何代码。

—WIRE 协议： ACSII 或 BINARY

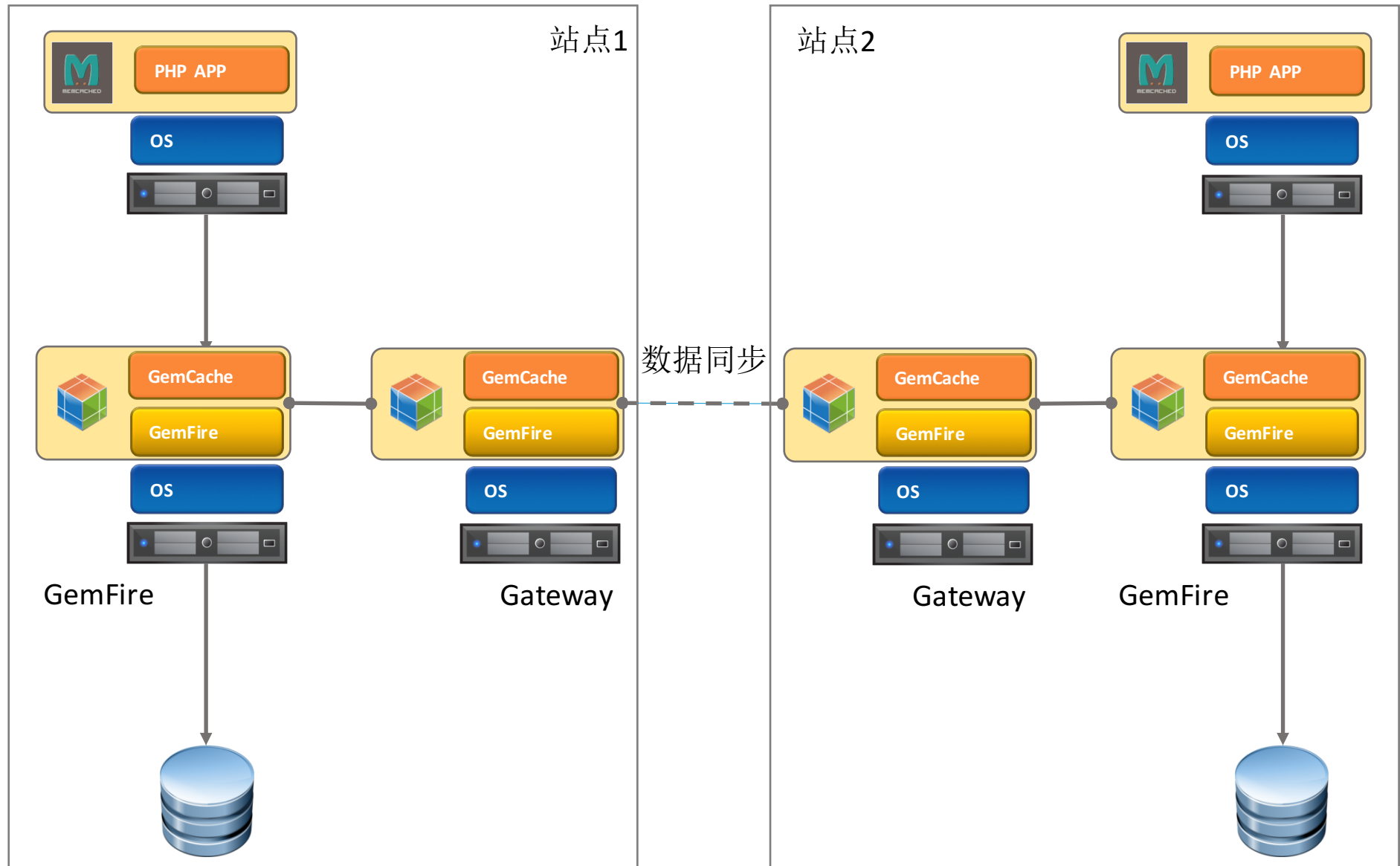


# Single DC Cluster



- 并发 I/O
- 流控
- 故障切换
- 整流

# Multi DC Cluster



# GemCache的优势

---

- **高可扩展性：** GemCache 可以自动添加节点到集群中，Memcache 客户端不需要维护服务器列表。一个Memcache 客户端需要访问多份保存在多个节点的数据，导致服务器对每个客户端都建立TCP连接，而Memcache 客户端只需要连接一个GemCache 服务器即可，大幅减小TCP连接数。
- **数据一致性：** Memcache客户端须维护一个服务器列表，一旦某客户端的列表不正确，则给客户端返回的脏数据。
- **HA高可用：** 当一个Memcache服务器挂掉后，会导致Memcache 集群故障或性能降级，客户端直连后端DB。所有故障处理都必须应用来做，而GemFire可自动解决这一问题。
- **集群热启动：** 当一个Memcache集群故障后，数据必须重新加载和分布到各集群成员，而GemCache可以从其他节点或本地磁盘加载数据，大幅节省时间消耗。
- **断网处理：** 几百台Memcache集群部署，由于网络故障，客户端不可能连接所有的节点，必须从数据库拉取数据，避免出现脏数据。而GemCache自身能够处理断网情况，保障响应数据的一致性。

# GemFire与Redis

在Apache Geode工程中，**GemFire** 目前正在与**Redis**做集成。

工程地址为：

<https://github.com/apache/incubator-geode/tree/a781843b160de7b751b8d32990a163fe31ef798c/geode-core/src/main/java/com/gemstone/gemfire/redis>

Tree: a7818... ▾


New file

Upload files

Find file


History

[incubator-geode](#) / [geode-core](#) / [src](#) / [main](#) / [java](#) / [com](#) / [gemstone](#) / [gemfire](#) / **redis** /

 **sboorlagadda** GEODE-52: Remove @author tags from Java source ...

Latest commit 7d944f6 on Mar 3

..

 [GemFireRedisServer.java](#)

GEODE-52: Remove @author tags from Java source

19 days ago

# GemFire与Redis

---

在 GemFire 内核中，新添加一个GemFireRedisServer服务器，此服务器可以解析 Redis 协议，当有命令发送到这个服务器上时，每个命令都会被中断、执行并响应给客户端。

默认的连接端口为6379，也可以修改为其他端口。

每个 Redis 数据类型（String和HyperLogLog）都会被保存在单独的Region中，默认的Region类型为RegionShortcut#PARTITION。

在执行事务时，事务只能作用在本地，或开启事务处理的持久化 Region。另外，默认情况下看不到Key 键（在 GemFire 事务中是能够看到的）



# GemFire与Redis

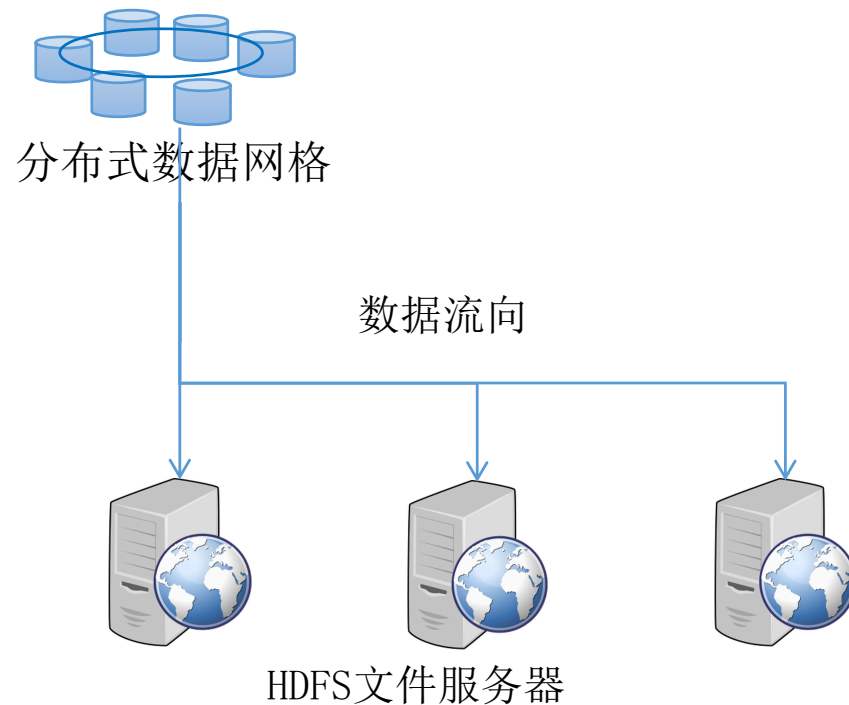
GemFire支持Redis命令如下：

| 命令类型                 | 命令内容  |
|----------------------|---|
| <b>String命令</b>      | APPEND, BITCOUNT, BITOP, BITPOS, DECR, DECRBY, GET, GETBIT, GETRANGE, GETSET, INCR, INCRBY, INCRBYFLOAT, MGET, MSET, MSETNX, PSETEX, SET, SETBIT, SETEX, SETNX, STRLEN                            |
| <b>List命令</b>        | LINDEX, LLEN, LPOP, LPUSH, LPUSHX, LRANGE, LREM, LSET, LTRIM, RPOP, RPUSH, RPUSHX   |
| <b>Hash命令</b>        | HDEL, HEXISTS, HGET, HGETALL, HINCRBY, HINCRBYFLOAT, HKEYS, HMGET, HMSET, HSETNX, HLEN, HSCAN, HSET, HVALS  |
| <b>Set命令</b>         | SADD, SCARD, SDIFF, SDIFFSTORE, SINTER, SINTERSTORE, SISMEMBER, SMEMBERS, SMOVE, SREM, SPOP, SRANDMEMBER, SCAN, SUNION, SUNIONSTORE   |
| <b>SortedSet命令</b>   | ZADD, ZCARD, ZCOUNT, ZINCRBY, ZLEXCOUNT, ZRANGE, ZRANGEBYLEX, ZRANGEBYSCORE, ZRANK, ZREM, ZREMRANGEBYLEX, ZREMRANGEBYRANK, ZREMRANGEBYSCORE, ZREVRANGE, ZREVRANGEBYSCORE, ZREVRANK, ZSCAN, ZSCORE |
| <b>HyperLogLog命令</b> | PFADD, PFCOUNT, PFMERGE   |
| <b>Keys命令</b>        | DEL, DBSIZE, EXISTS, EXPIRE, EXPIREAT, FLUSHDB, FLUSHALL, KEYS, PERSIST, PEXPIRE, PEXPIREAT, PTTL, SCAN, TTL  |
| <b>Transaction命令</b> | DISCARD, EXEC, MULTI  |

# GemFire利用HDFS进行日志处理

---

GemFire在生产环境运行过程中会产生大量日志，那么通过定期将日志文件批量传输到HDFS文件服务器中，通过 Hadoop 集群对日志进行分析处理，进一步加强分布式数据网络的运维手段。



# 日志传输方式

---

1. GemFire定期备份日志，将日志传到FTP服务器，FTP定期向HDFS分布式文件系统传输。
2. GemFire 通过 Flume 或 Kafka准实时传输日志文件到HDFS分布式文件系统
3. GemFire通过脚本设置定期将日志文件通过SCP传输给HDFS分布式文件系统



THANKS

SequeMedia  
盛拓传媒

IT168.com

ChinaUnix

ITPUB