

# 程序媛的分布式数据库 开发经验

李霞 @PingCAP

# 自我介绍

## ——前世今生

前 世	京 东	分布式系统程序媛
今 生	PingCAP	分布式系统程序员
来 世	PingCAP	分布式系统程序猿

@紫沐夏\_go

# 程序媛的分布式数据库开发经验

- 分布式数据库现况
  - MySQL 水平扩展
  - NoSQL
  - NewSQL
- TiDB
- 如何给 TiDB 提 PR
- 如何在 github 上提代码

# MySQL 水平扩展

- 主从，一主多从
  - 读写分离，通过队列暂存请求... 以提高读写性能和读库的可扩展性
  - 写入仍然是单点

# MySQL 水平扩展

- 中间件 Sharding

- 常见的开源方案有：Cobar, Vitess, MyCat 等
- 这些方案的思路是拦截 SQL 的请求通过 sharding key 和一定规则，将请求转发/广播到不同的 MySQL 实例上
- 从而实现水平扩展的效果，基本解决了单点写入的问题，整体提升了业务的吞吐
- 缺点如下：
  - 大部分中间件没有解决动态扩容的问题，采用了静态的路由策略，扩容一般还处于人工 x2 的状态，对 DBA 要求较高。
  - 从一定程度上来说都放弃了事务。
  - 对业务不透明，需要指定 sharding key，心智负担较大

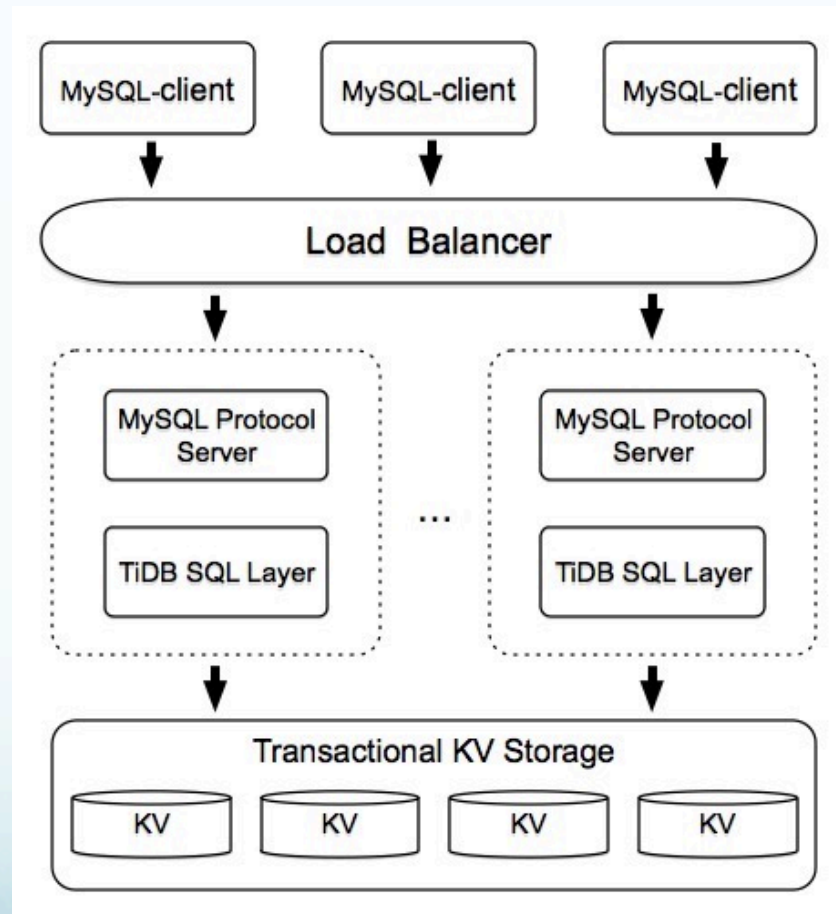
# NoSQL

- 常见的选型方案是：HBase，MongoDB，Cassandra 等
  - HBase 参考的是 Google Big Table 的论文
    - 优点；可扩展性好，比较高的随机读写性能（吞吐），强一致性，支持多版本，支持 Table 和半结构化的数据存储。
    - 缺点：不支持复杂查询，不支持跨行事务。
    - CP 模型，官方不支持 SQL，有一些第三方的公司做了 SQL on HBase。
  - Cassandra 来自 Amazon Dynamo 的分布式架构
    - 明显特点：可以根据业务的需求决定它是 CP 还是 AP。
      - 通过 WRN 模型调整
      - $W + R > N$ ，CP 系统
      - $W + R \leq N$ ，AP 系统
    - 2.0+ 支持了 CAS 之后算是支持单行事务。
    - 读写性能略优于 HBase（<http://www.planetcassandra.org/nosql-performance-benchmarks/>）。

# NewSQL

- NoSQL
  - 优：去中心化、可扩展性、容错能力等属性
  - 劣：很多做不到ACID特性，且不支持SQL。
- NewSQL
  - SQL + NoSQL
  - 例如 Google 的 Spanner 和 F1，被 Apple 收购的 FoundationDB，近年出现的 CockroachDB，TiDB 等，主推能提供 SQL 及分布式事务数据库产品。

# TiDB





# TiDB

- 演示 WordPress on TiDB

# TiDB

- 支持 MySQL 协议
  - 用户从 MySQL 的相关解决方案迁移过来时几乎没迁移成本。
- 支持分布式事务
  - 2PC（二阶段提交），参考 Google percolator 的论文
- 支持动态变更 schema
  - 参考 Google 动态异步变更 schema 的论文实现
- 支持动态扩容

# 如何给 TiDB 提 PR

- 要找到一个 TiDB 代码里面的一个问题
  - 从项目已有的 issue 中找一个
  - 看或者使用 TiDB 项目的时候发现的一些问题
- 把 TiDB 源代码下载到本地
  - 可以在 TiDB 上 fork 一个分支，把这个分支 clone 到本地，在上面建个分支
  - 直接把 TiDB 这个项目 clone 到本地，在上面建个分支
- 根据 issue 或者已经发现的问题，添加或改写代码
  - 添加必要注释，测试代码
  - 本地 make 代码，确定通过
- 提交代码
  - TiDB 的成员会 review 代码，并提出建议
- Merge到主干
  - 集齐2个 LGTM，召唤 TiDB 的文化衫

# 如何给 TiDB 提PR

- 例子，给 TiDB 添加 ADDDATE() 这个函数支持
  - 在 issue#236里有提到，有些尚未支持的时间处理函数
  - 函数具体功能细节，可参考 MySQL 官网文档，如果文档与 MySQL（version 5.7+）运行结果冲突，以后为准
  - 在主干上开分支
    - git clone 或者 go get 源码
    - git branch -b user/add-date （一般分支格式）

# 如何给 TiDB 提 PR

- TiDB 添加 ADDDATE() —— SQL 层代码

- lex 与 yacc 类似的分为三段

% {定义段} %

%%

规则段

%%

用户程序段

- 我们使用了一个开源的 Parser 生成器, cznic/goyacc 和 cznic/ebnf2y

# 如何给 TiDB 提 PR

- TiDB 添加 ADDDATE() —— SQL 层代码
  - scanner.l
    - 定义段

```
abs      {a}{b}{s}  
add      {a}{d}{d}  
adddate  {a}{d}{d}{d}{a}{t}{e}
```

- 规则段

```
{abs}      lval.item = string(l.val)  
           return abs  
{add}      return add  
{adddate} return addDate
```

# 如何给 TiDB 提 PR

- TiDB 添加 ADDBDATE() —— SQL 层代码
  - parser.y

```
%token <item>

/*yy:token "1.%d" */ floatLit      "floating-point literal"
/*yy:token "%c" */  identifier     "identifier"
/*yy:token "%d" */  intLit         "integer literal"
/*yy:token "\"%c\"" */ stringLit    "string literal"
/*yy:token "%x" */  hexLit         "hexadecimal literal"
/*yy:token "%b" */  bitLit         "bit literal"

abs          "ABS"
add          "ADD"
addDate      "ADDBDATE"
```

# 如何给 TiDB 提 PR

- TiDB 添加 ADDDATE() —— 语法分析代码

```
DateArithMultiFormsOpt '(' Expression ',' DateArithInterval')'
{
    $$ = &ast.FuncDateArithExpr{
        Op: $1.(ast.DateArithType),
        Date: $3.(ast.ExprNode),
        DateArithInterval: $5.(ast.DateArithInterval),
    }
}
```

```
DateArithMultiFormsOpt:
    "ADDDATE"
    {
        $$ = ast.DateAdd
    }
|
    "SUBDATE"
    {
        $$ = ast.DateSub
    }
```



# 如何给 TiDB 提 PR

- TiDB 添加 ADDDATE() —— 语法分析

```
DateArithInterval:
  Expression
  {
    $$ = ast.DateArithInterval{
      Form: ast.DateArithDaysForm,
      Unit: "day",
      Interval: $1.(ast.ExprNode),
    }
  }
| "INTERVAL" Expression TimeUnit
{
  $$ = ast.DateArithInterval{Unit: $3.(string), Interval: $2.(ast.ExprNode)}
}
```

# 如何给 TiDB 提 PR

- TiDB 添加 ADDDATE() —— 功能具体实现

- statement

- plan tree

- function

- buildin

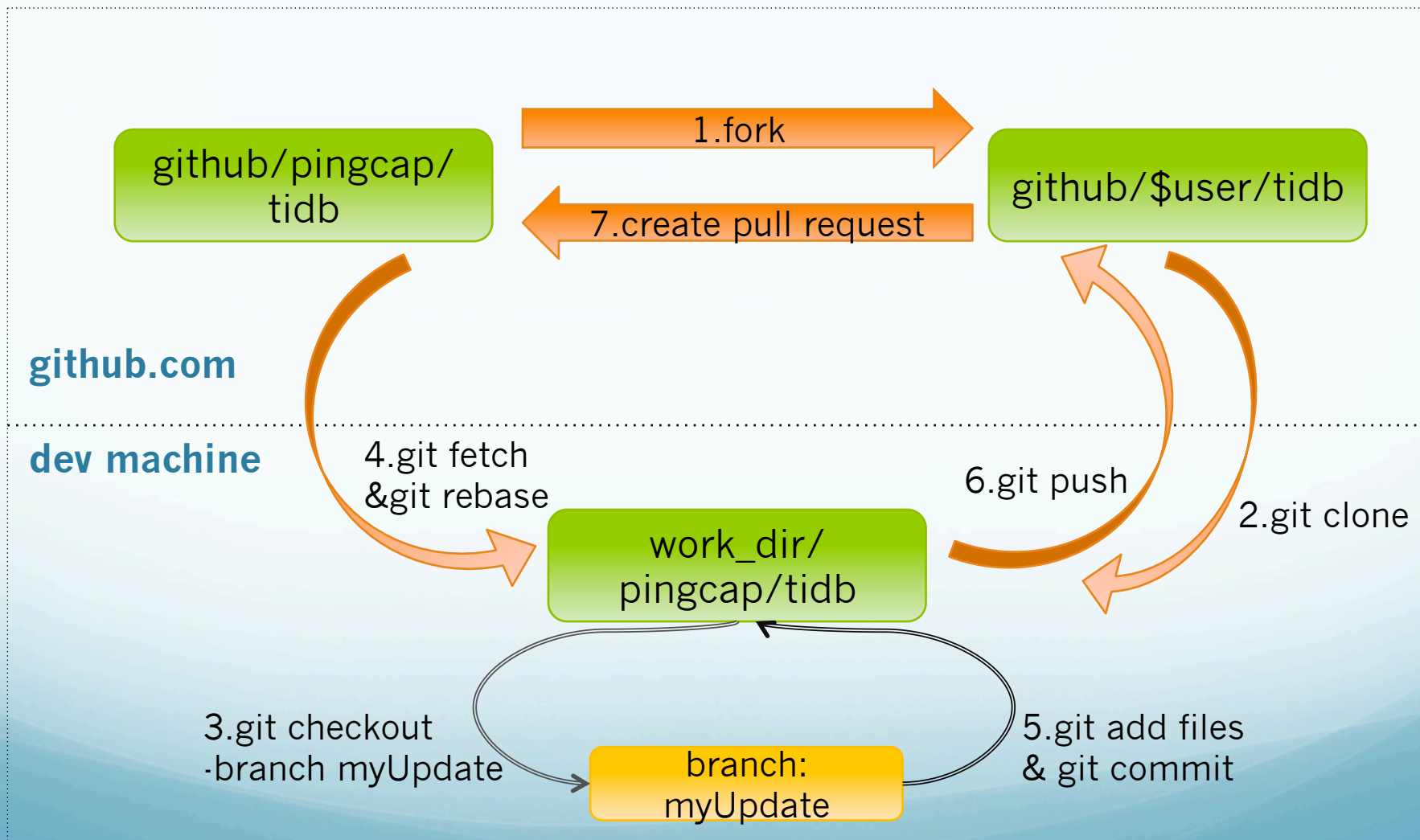
- general

```
+DateArith : struct
    [fields]
    +Date : Expression
    +Form : DateArithType
    +Interval : Expression
    +Op : DateArithType
    +Unit : string
    [methods]
    +Accept(v Visitor) : Expression, error
    +Clone() : Expression
    +Eval(ctx context.Context, args map[int]Expression) : Expression, error
    +IsStatic() : bool
    +String() : string
```

# 如何给 TiDB 提 PR

- TiDB 添加 ADDDATE() ——提交前准备
  - 测试
    - 单元测试
    - 测试覆盖率
      - `go tool cover -html=coverage.out`  
`go test -coverprofile=coverage.out`
  - make
    - `go tool vet`, `go tool vet --shadow`
    - `go lint`
    - `gofmt -s -l`

# 如何在 github 上提代码



# Q&A

@紫沐夏\_go