Devlink
第二届PHP全球开发者大会

LIANJIA.com 链家

PHP做大数据量实时分析

基于日志数据的实时分析系统

Devlink 第二届PHP全球开发者大会

LIANJIA 链家 .com

吕毅

链家网平台服务架构师

一名在路上的程序员

云计算浪尖时参与建设国内第一家PaaS平台SAE，

移动浪尖时作为初期成员与团队打造了国内Top3的SuperAPP手机百度，

重度垂直时代加入链家网负责平台化服务建设。

吕毅

链家网平台服务架构师

一名在路上的程序员

云计算浪尖时参与建设国内第一家PaaS平台SAE，

移动浪尖时作为初期成员与团队打造了国内Top3的SuperAPP手机百度，

重度垂直时代加入链家网负责平台化服务建设。

链家网平台服务组

Platform Services Group

Sam

存储方向负责人

通用存储服务提供标准的文件操作类接口，完美兼容AWS S3协议。基于存储服务之上，提供图片处理、视频转码服务，存储整体架构内包含CDN、ATS缓存能力、多机房数据同步与互备能力。

Biwing

话务平台负责人

话务平台快捷接入十余家电信服务供应商，持有号码资源1k万。平台通过内部调度、号码回收与复用机制，支持公司经纪人周通话30+万通，满足链家10+万经纪人多重号码需求，并管控通话。

Monika

数据方向负责人

基于实时分析计算框架，构建日志实时分析服务面向技术角色提供服务监控、预警能力，搭建统计平台服务面向产品、运营角色提供产品数据支撑，并用Go解决数据并发分析能力。

Rambone

研发工程师

擅长PHP，精通FE开发，具有数据分析公司工作背景，格式化数据精准展现。现参加话务、数据方向工作，并负责Services Panel的开发维护。

Skylee

研发工程师

擅长C++，具备Go开发能力，曾在中科院负责网络流量分析工作，擅长网络包处理与流量监控，目前投身于存储方向研发。

Kingho1d

实习工程师

C/C++基本功扎实，大学三年参加ACM竞赛，熟悉PHP开发，掌握一定的FE技能，目前参加话务、数据方向工作。

链家网平台服务组

Platform Services Group

Sam

存储方向负责人

通用存储服务提供标准的文件操作类接口，完美兼容AWS S3协议。基于存储服务之上，提供图片处理、视频转码服务，存储整体架构内包含CDN、ATS缓存能力、多机房数据同步与互备能力。

Biwing

话务平台负责人

话务平台快捷接入十余家电信服务供应商，持有号码资源1k万。平台通过内部调度、号码回收与复用机制，支持公司经纪人周通话30+万通，满足链家10+万经纪人多重号码需求，并管控通话。

Monika

数据方向负责人

基于实时分析计算框架，构建日志实时分析服务面向技术角色提供服务监控、预警能力，搭建统计平台服务面向产品、运营角色提供产品数据支撑，并用Go解决数据并发分析能力。

Rambone

研发工程师

擅长PHP，精通FE开发，具有数据分析公司工作背景，格式化数据精准展现。现参加话务、数据方向工作，并负责Services Panel的开发维护。

Skylee

研发工程师

擅长C++，具备Go开发能力，曾在中科院负责网络流量分析工作，擅长网络包处理与流量监控，目前投身于存储方向研发。

Kingho1d

实习工程师

C/C++基本功扎实，大学三年参加ACM竞赛，熟悉PHP开发，掌握一定的FE技能，目前参加话务、数据方向工作。

lianjia.com
lianjia.com
www.lianjia.com
SINCE 2001

内容摘要

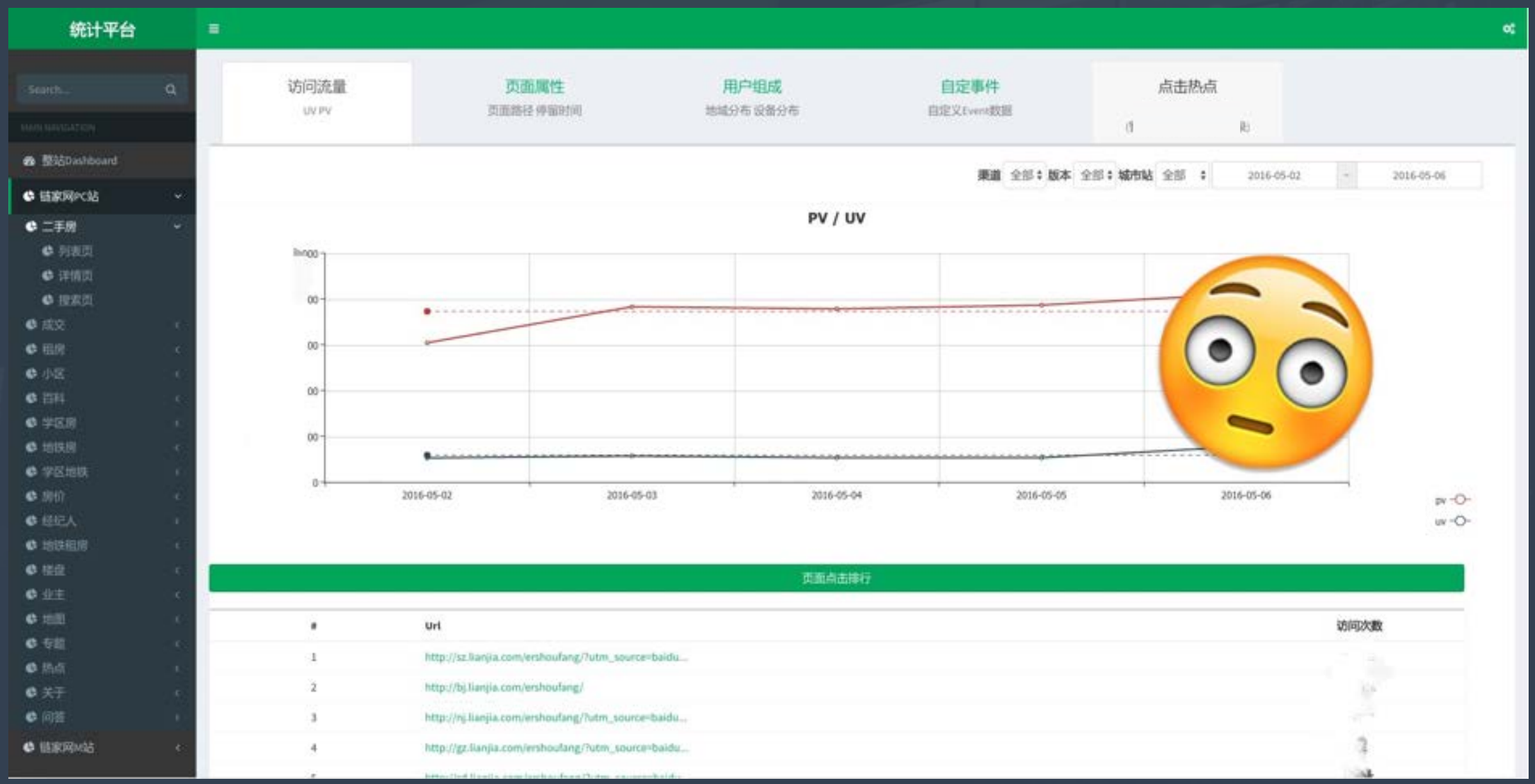
Summary

案例Case

Show Case

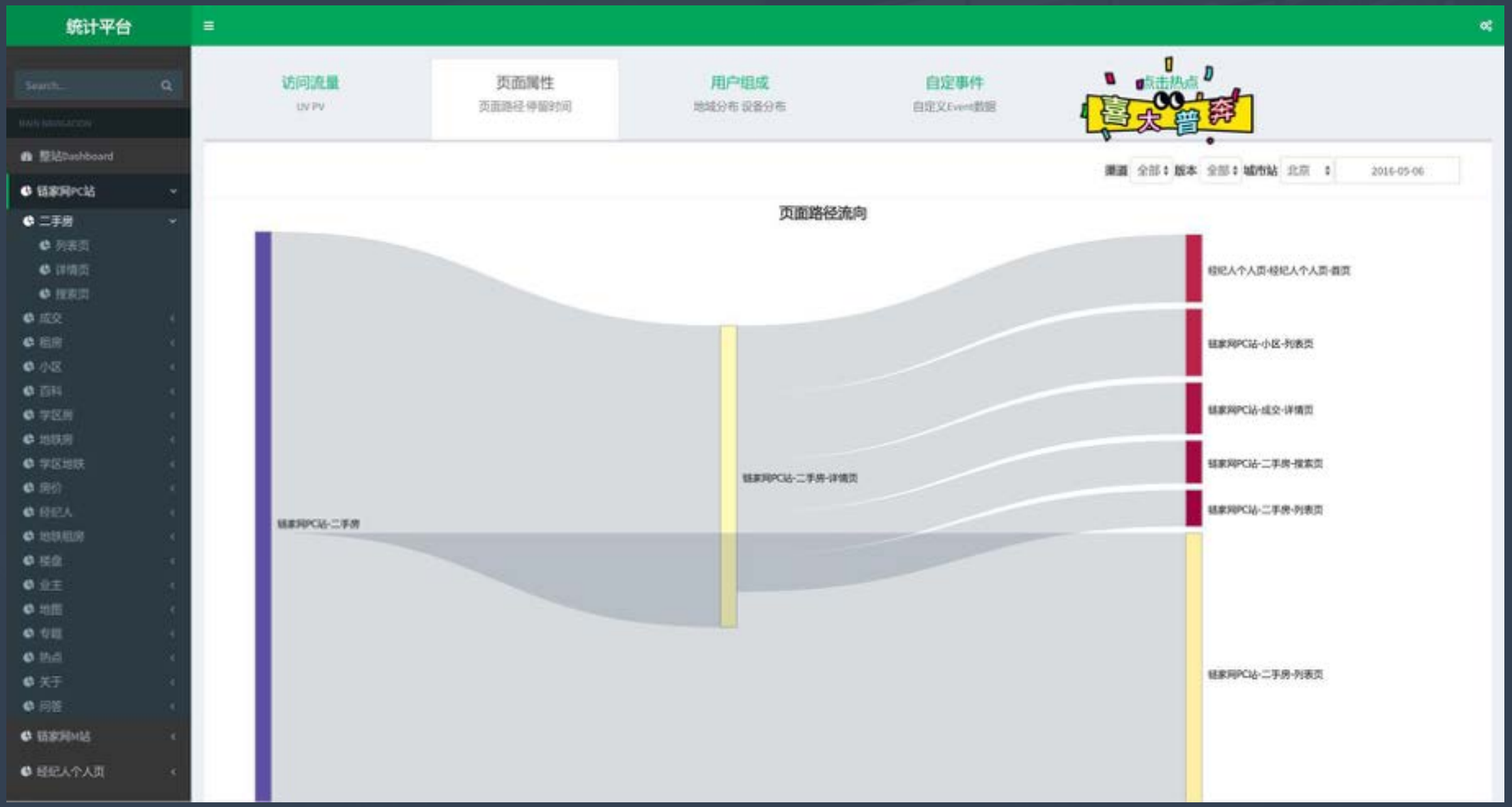
链家网 统计平台

Statistical service platform



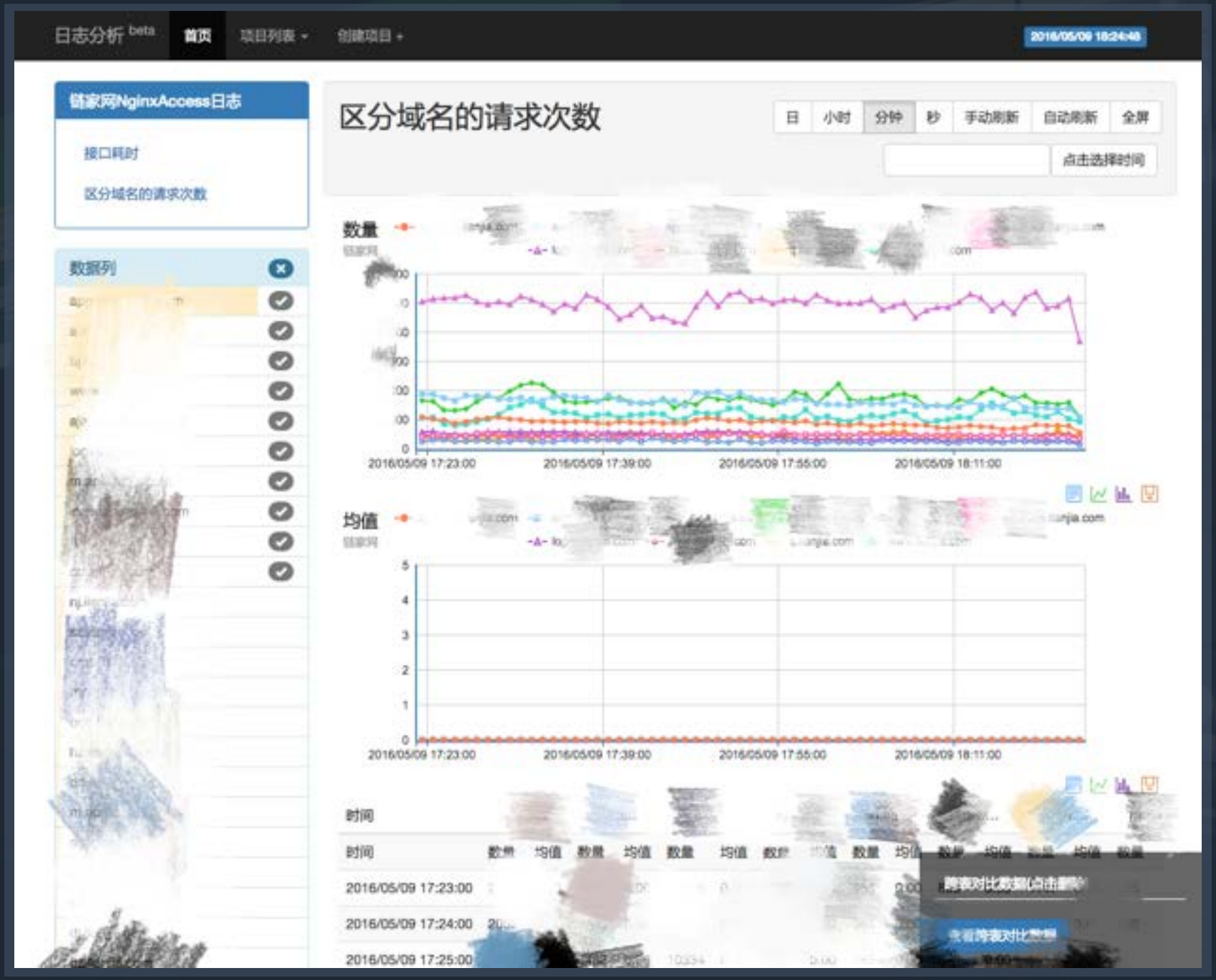
链家网 统计平台

Statistical service platform



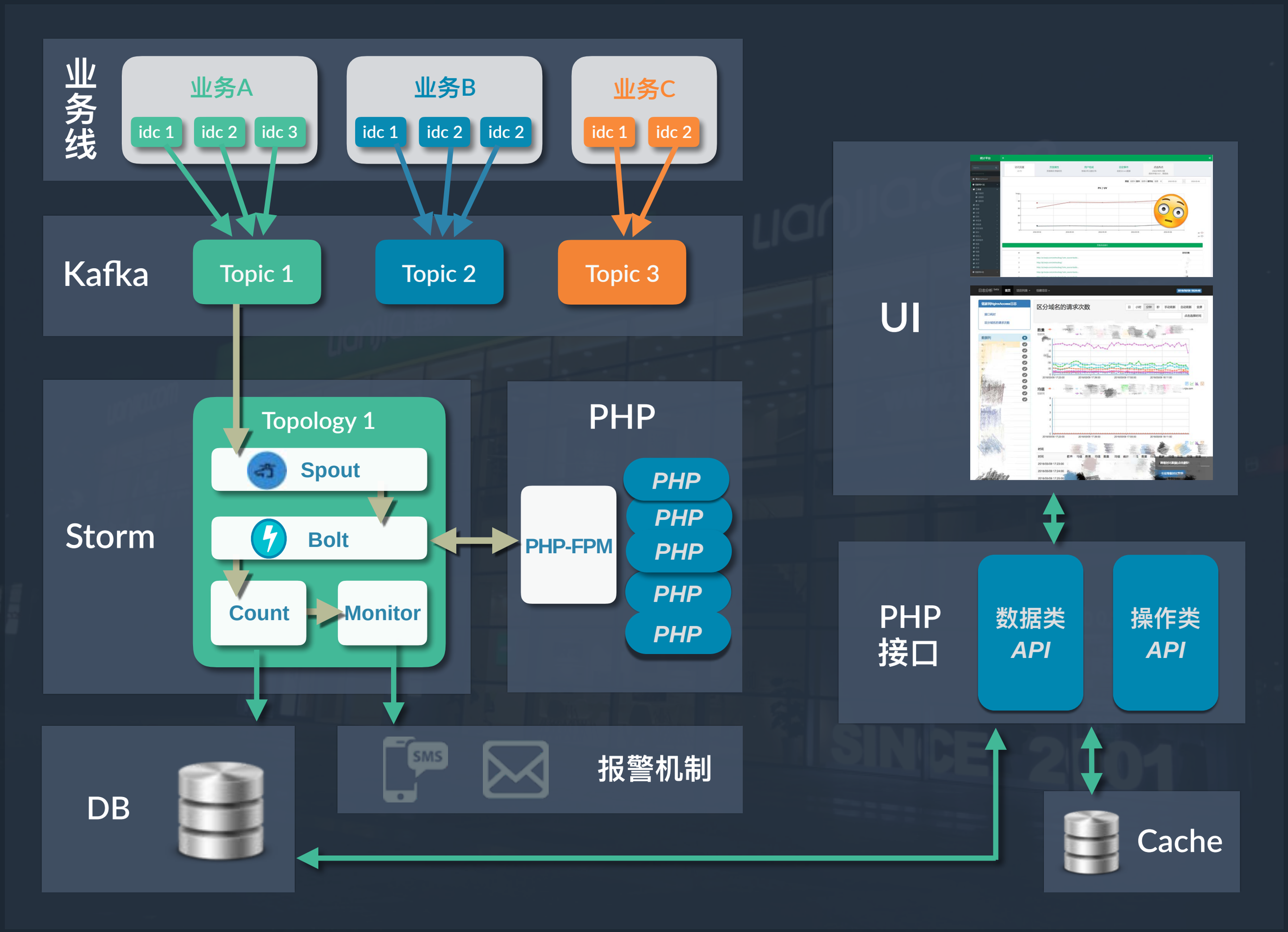
链家网 日志分析服务

Real-time log analyse service

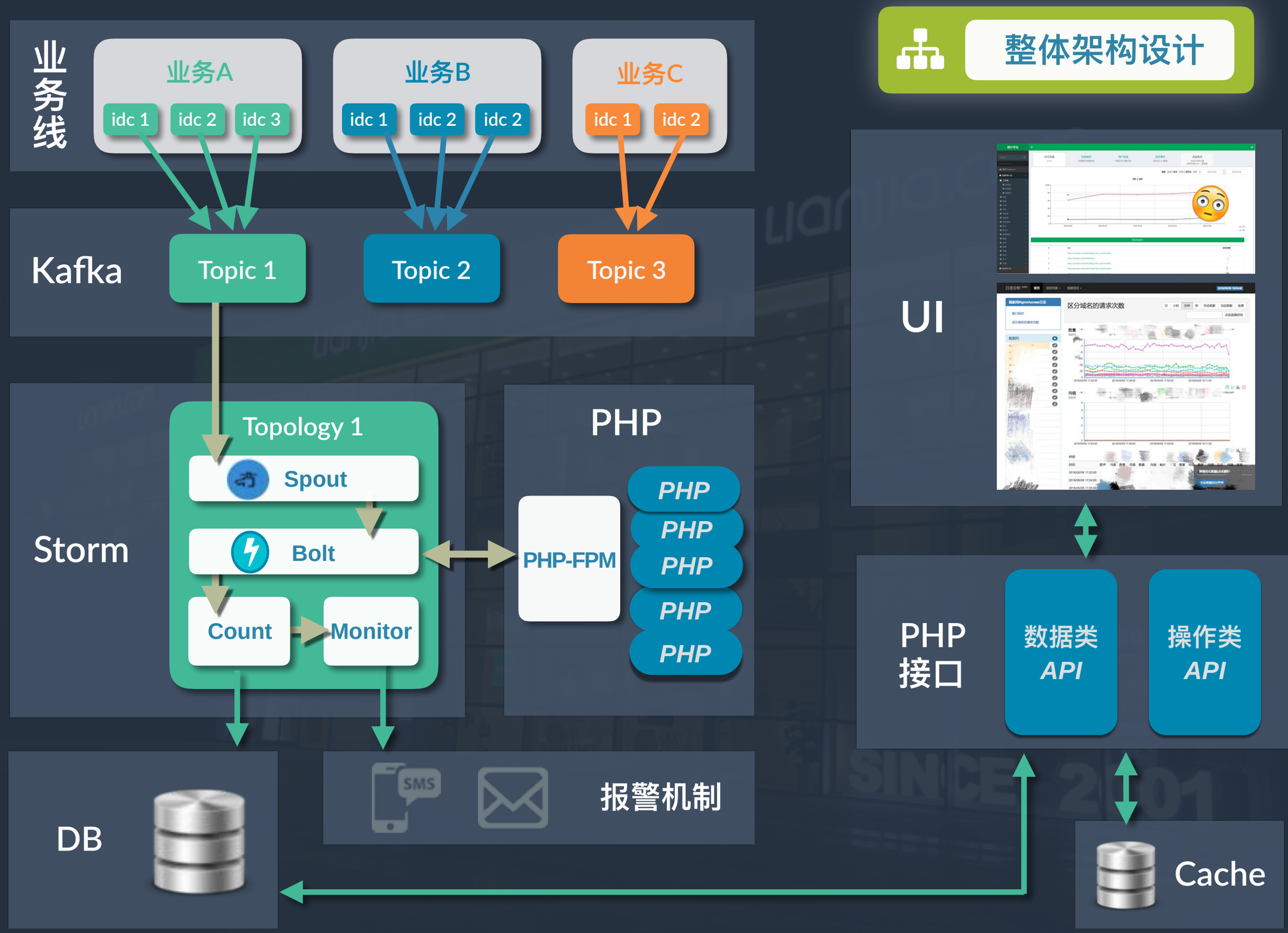


整体架构设计

Architecture design


业务线
业务A
idc 1
idc 2
idc 3
业务B
idc 1
idc 2
idc 2
业务C
idc 1
idc 2
Kafka
Topic 1
Topic 2
Topic 3
Storm
Topology 1
Spout
Bolt
Count
Monitor
PHP
PHP-FPM
PHP
PHP
PHP
PHP
PHP
PHP
UI
PHP
接口
数据类
API
操作类
API
DB
SMS
报警机制
Cache

整体架构设计

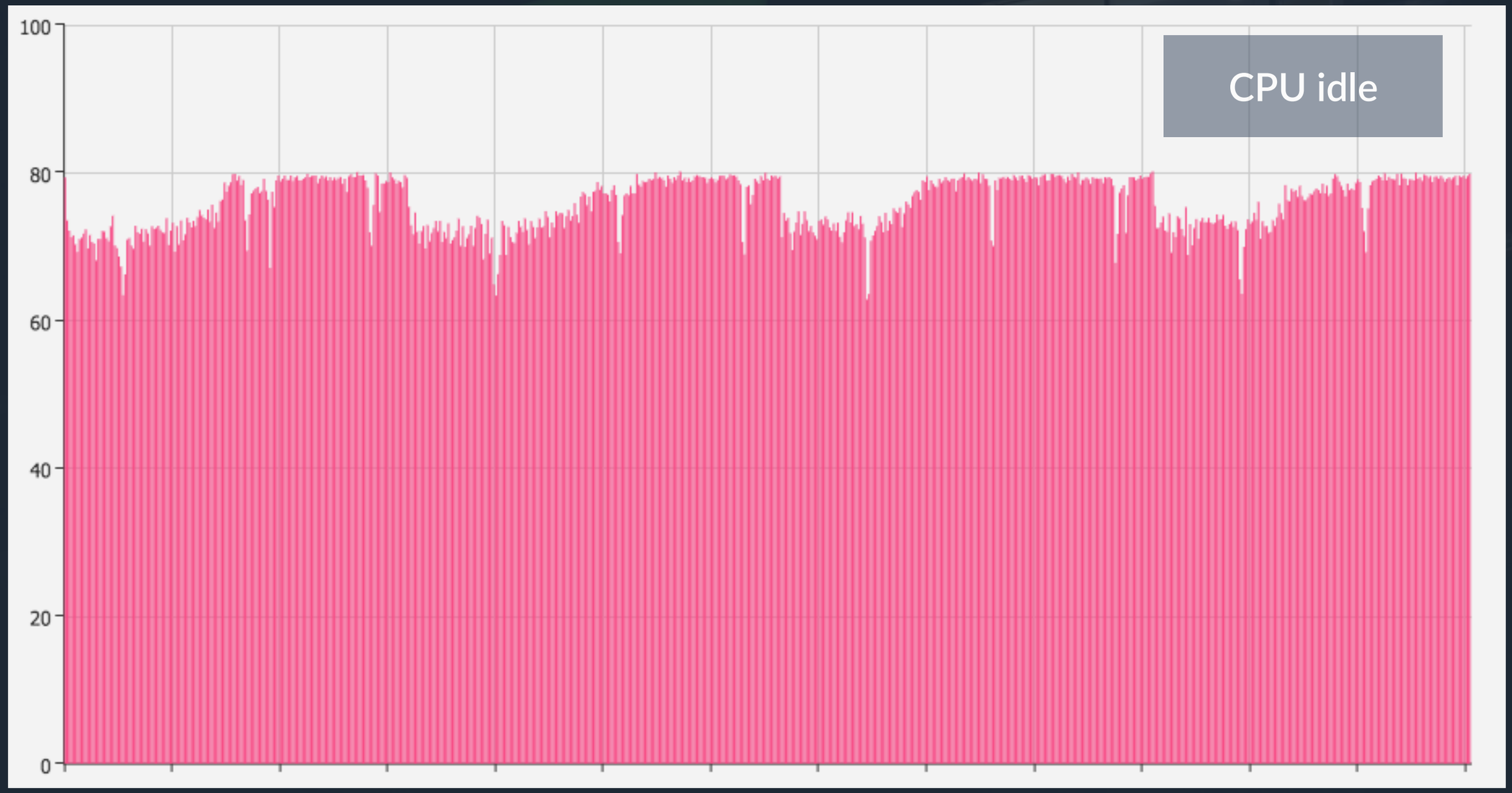

业务线
业务A
idc 1
idc 2
idc 3
业务B
idc 1
idc 2
idc 2
业务C
idc 1
idc 2
Kafka
Topic 1
Topic 2
Topic 3
Storm
Topology 1
Spout
Bolt
Count
Monitor
PHP
PHP-FPM
PHP
PHP
PHP
PHP
PHP
PHP
UI
PHP
接口
数据类
API
操作类
API
DB
SMS
报警机制
Cache

lianjia.com
lianjia.链家
lianjia.com
www.lianjia.com
SINCE 2001

性能相关数据

Performance data

系统: CentOS 7　　CPU: 2颗 Intel E5　　内存: 128G　　每秒30K行日志

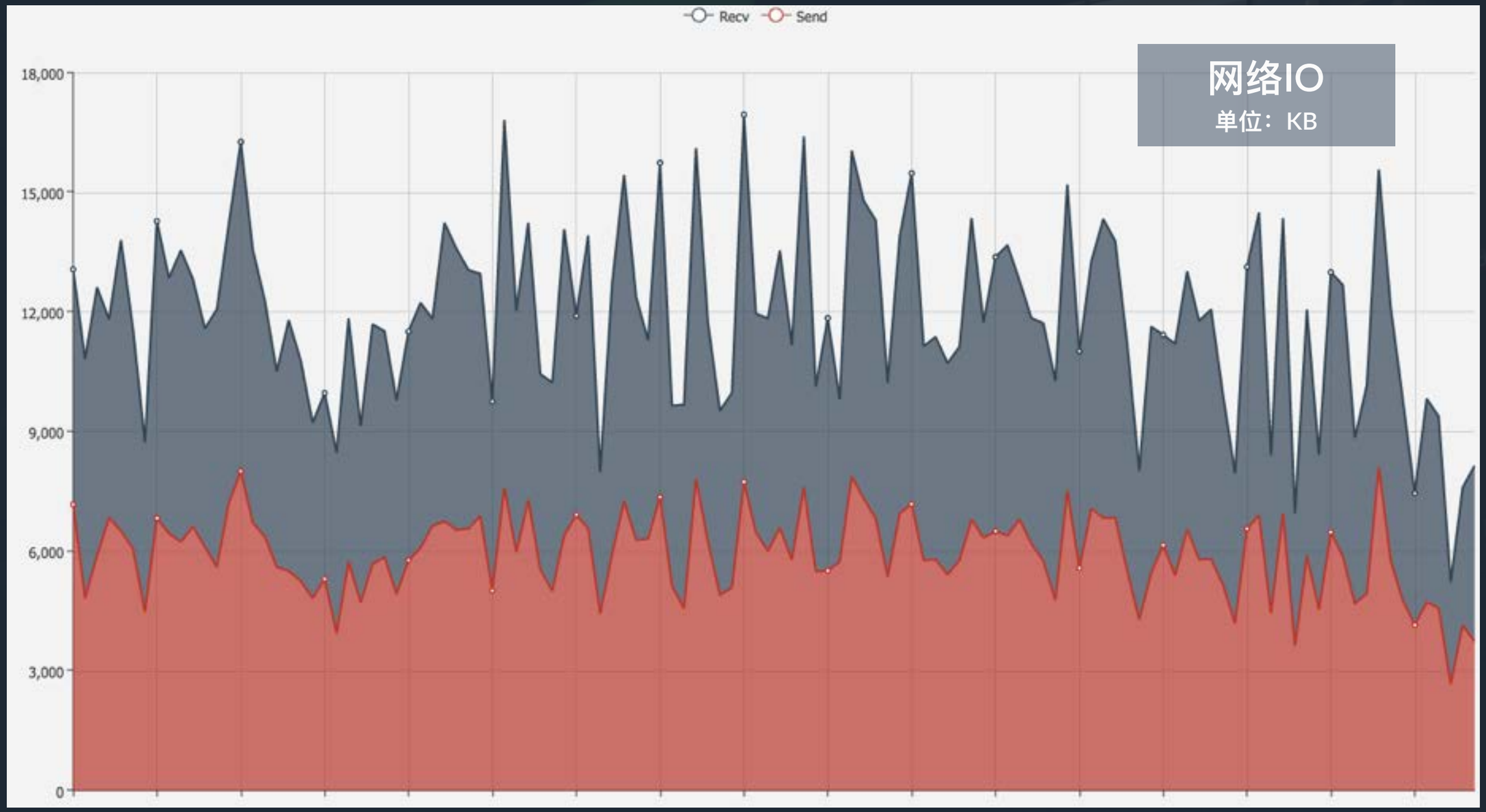


lianjia.com
lianjia.链家
lianjia.com
www.lianjia.com
SINCE 2001

性能相关数据

Performance data

系统: CentOS 7　　CPU: 2颗 Intel E5　　内存: 128G　　每秒30K行日志



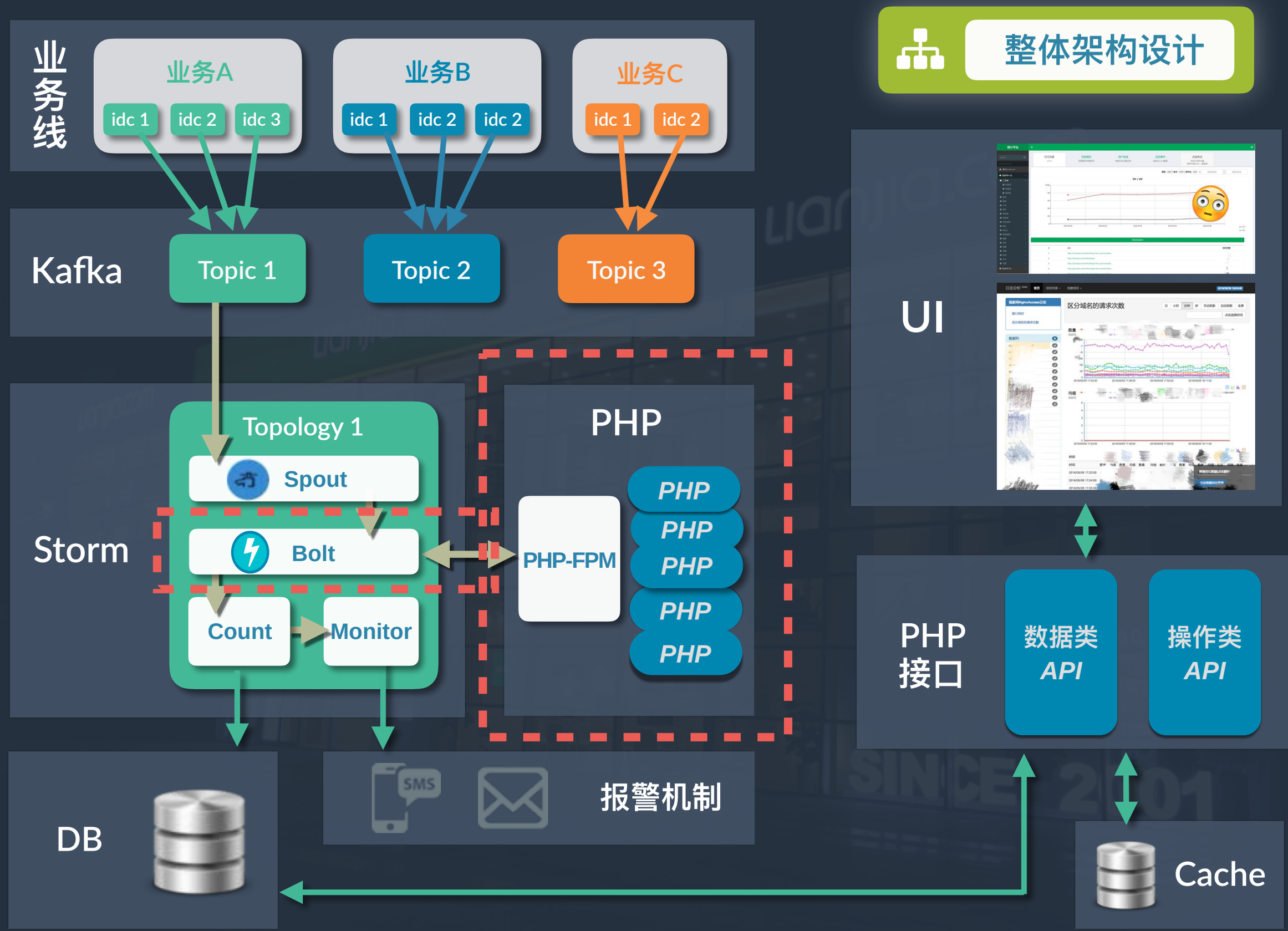
```
fcomplex Csqrt(fcomplex z)
{
    fcomplex c;
    float w;
    if ((z.r == 0.0) && (z.i == 0.0)) {
        c.r=0.0;
        c.i=0.0;
    } else {
        w = sqrt((sqrt(z.r*z.r + z.i
        if (z.r >= 0.0){
            c.r=w;
            c.i=z.i/(2.0*w);
```

PHP

PHP分析程序

php analysis program

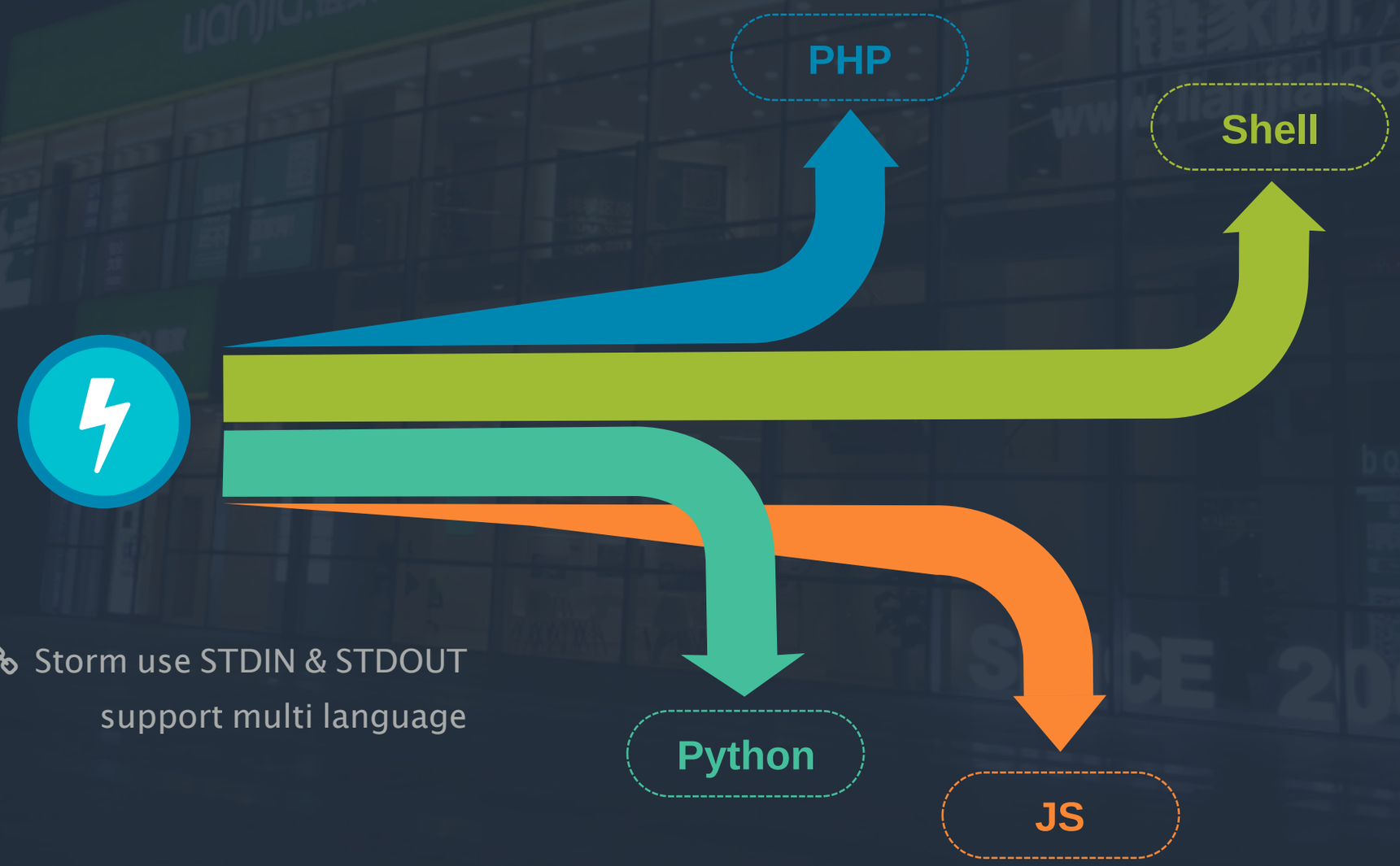
整体架构设计


业务线
业务A
idc 1
idc 2
idc 3
业务B
idc 1
idc 2
idc 2
业务C
idc 1
idc 2
Kafka
Topic 1
Topic 2
Topic 3
Storm
Topology 1
Spout
Bolt
Count
Monitor
PHP
PHP-FPM
PHP
PHP
PHP
PHP
PHP
PHP
UI
PHP
接口
数据类
API
操作类
API
DB
SMS
报警机制
Cache

lianjia.com
lianjia.链家
lianjia.com
www.lianjia.com
SINCE 2001

Storm 多语言支持

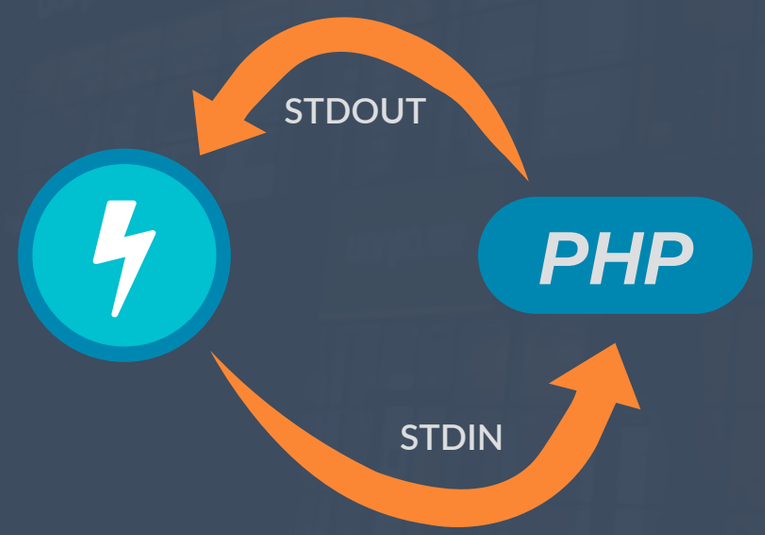
Storm multi language support



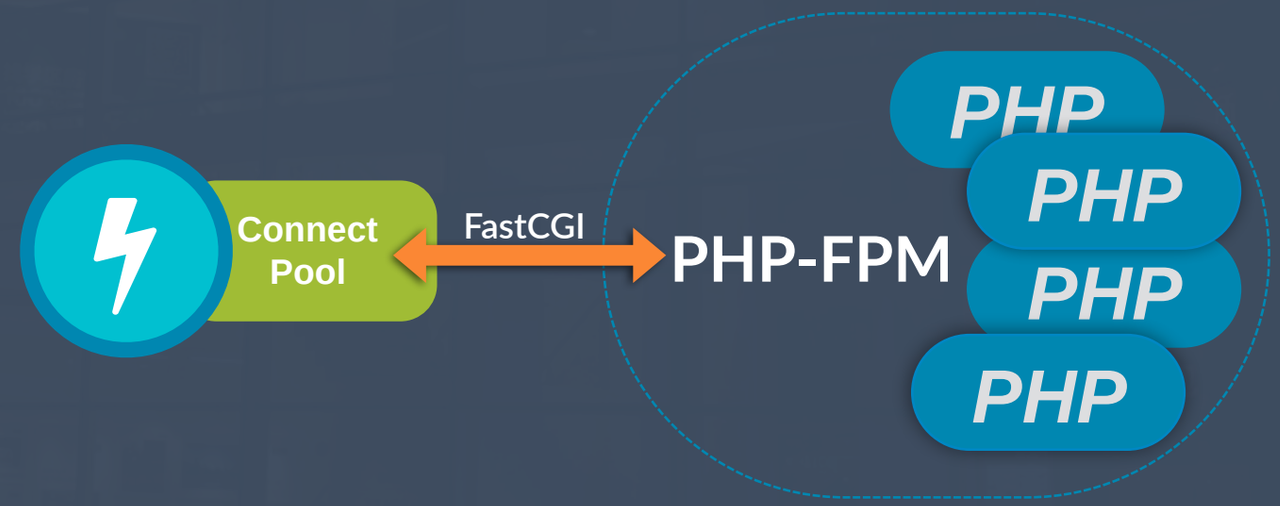
Storm use STDIN & STDOUT support multi language

连接 Storm 与 PHP

Connect Storm to PHP



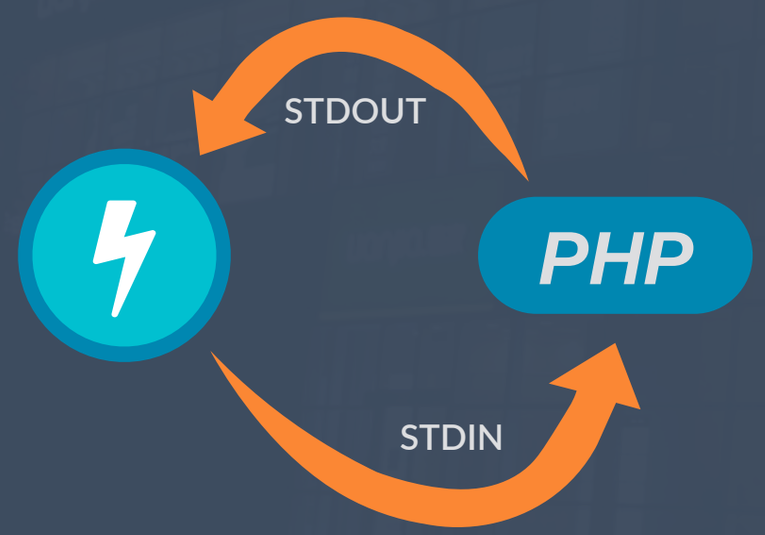
Traditional mode



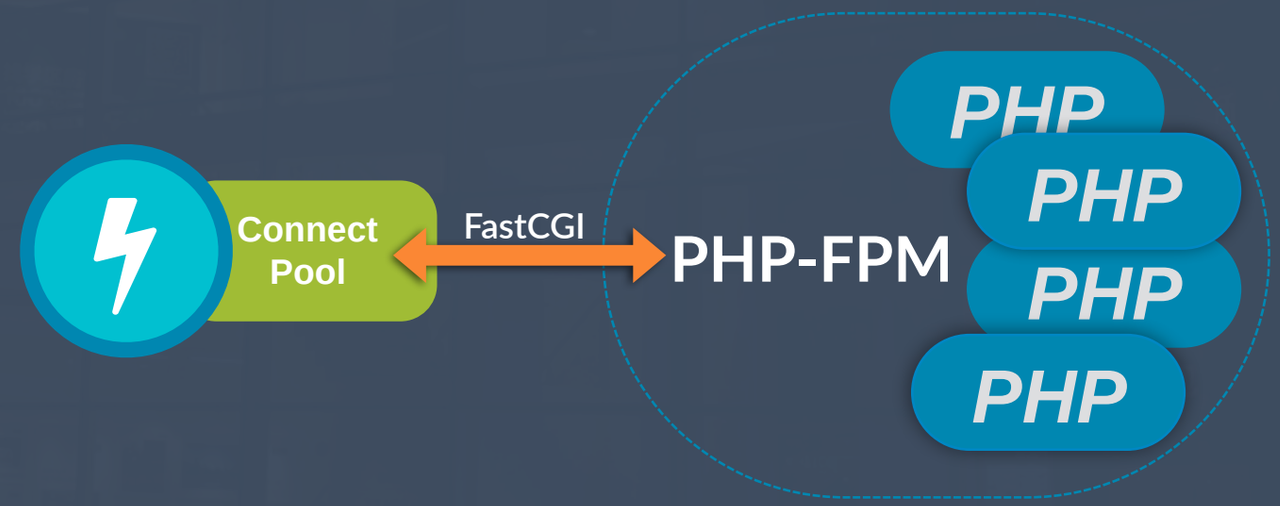
Transformed model

连接 Storm 与 PHP

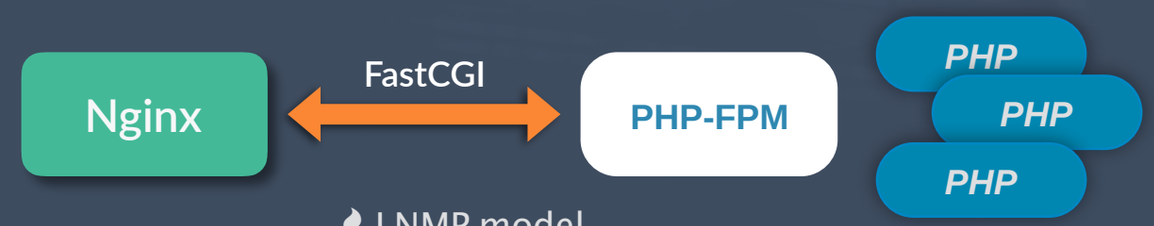
Connect Storm to PHP



Traditional mode



Transformed model



LNMP model

```java
package lianjia.rlam.utils;

import com.googlecode.fcgi4j.FCGIConnection;

public class FCGI{
    public static String post( String jobName, String log ) {
        try{
            FCGIConnection connection = FCGIConnection.open();
            connection.connect( new InetSocketAddress("127.0.0.1", 9000) );
            connection.beginRequest( "/home/work/code/bridge.php" );
            connection.setRequestMethod( "POST" );
            connection.addParams( "SCRIPT_NAME", "/home/work/code/bridge.php" );
            connection.addParams( "SCRIPT_FILENAME", "/home/work/code/bridge.php" );

            String logEncode = java.net.URLEncoder.encode( log, "utf-8" );
            String postStr = "name="+ jobName +"&log="+ logEncode;
            byte[] postData = postStr.getBytes();
            connection.setContentLength(postData.length);
            connection.write(ByteBuffer.wrap(postData));

            ByteBuffer buffer = ByteBuffer.allocate(10240);
            connection.read(buffer);
            buffer.flip();
            byte[] data = new byte[buffer.remaining()];
            buffer.get(data);

            connection.close();
            return new String(data);
        } catch ( IOException e ) {
            return "";
        }
    }
}
```

```php
<?php
    $_name = $_POST['name'] ?? null;
    $_log = $_POST['log'] ?? null;

    // 检查参数是否存在
    if( !$_name || !$_log ) {
        exit();
    }

    // 检查对应的解析代码是否存在
    $path = dirname(__FILE__).DIRECTORY_SEPARATOR;
    $path.= $_name;

    if( !file_exists( $path ) ) {
        exit();
    } else {
        include_once( $path );
    }

    // 调用用户自定义代码
    $className = "ParseClass".ucfirst( $_name );
    if( !class_exists( $className ) ) {
        exit();
    }
    $log = urldecode( $_log );
    $obj = new $className();
    $result = $obj->execute( $log );

    // 将结果按一定格式返回
    // ... ...
```

```java
package lianjia.rlam.utils;

import com.googlecode.fcgi4j.FCGIConnection;

public class FCGI{
    public static String post( String jobName, String log ) {
        try{
            FCGIConnection connection = FCGIConnection.open();
            connection.connect( new InetSocketAddress("127.0.0.1", 9000) );
            connection.beginRequest( "/home/work/code/bridge.php" );
            connection.setRequestMethod( "POST" );
            connection.addParams( "SCRIPT_NAME", "/home/work/code/bridge.php" );
            connection.addParams( "SCRIPT_FILENAME", "/home/work/code/bridge.php" );

            String logEncode = java.net.URLEncoder.encode( log, "utf-8" );
            String postStr = "name="+ jobName +"&log="+ logEncode;
            byte[] postData = postStr.getBytes();
            connection.setContentLength(postData.length);
            connection.write(ByteBuffer.wrap(postData));

            ByteBuffer buffer = ByteBuffer.allocate(10240);
            connection.read(buffer);
            buffer.flip();
            byte[] data = new byte[buffer.remaining()];
            buffer.get(data);

            connection.close();
            return new String(data);
        } catch ( IOException e ) {
            return "";
        }
    }
}
```

```php
<?php
    $_name = $_POST['name'] ?? null;
    $_log = $_POST['log'] ?? null;

    // 检查参数是否存在
    if( !$_name || !$_log ) {
        exit();
    }

    // 检查对应的解析代码是否存在
    $path = dirname(__FILE__).DIRECTORY_SEPARATOR;
    $path.= $_name;

    if( !file_exists( $path ) ) {
        exit();
    } else {
        include_once( $path );
    }

    // 调用用户自定义代码
    $className = "ParseClass".ucfirst( $_name );
    if( !class_exists( $className ) ) {
        exit();
    }
    $log = urldecode( $_log );
    $obj = new $className();
    $result = $obj->execute( $log );

    // 将结果按一定格式返回
    // ... ...
```

PHP 7

lianjia.com
lianjia.链家
lianjia.com
www.lianjia.com
SINCE 2001

PHP解析日志程序

PHP program for parse log

程序无状态

PHP解析日志的代码程序无历史状态，工组内容仅为解析日志文本用

Input数据的校验

传入信息的内容先做格式校验后再做解析逻辑，业务线与中间环节容易产生脏数据

代码足够精简

使用PHP内置函数处理文本数据，不建议引入框架，精简代码来保证单次处理效率。少启用PHP扩展

PHP-FPM配置

建议dynamic管理PHP进程，pm.max_children决定着整体处理速度，应充分利用机器能力

避免网络开销

不要在PHP解析程序中做网络请求类操作，如操作DB、Cache等，容易阻塞Storm任务

统一入口

类似于web项目，应当有统一的入口文件做公共处理部分，具体日志解析规则由日志归属业务线RD编写

Data
Collection
Shift

实时数据收集

Real time data collection

Kafka —— 分布式消息系统

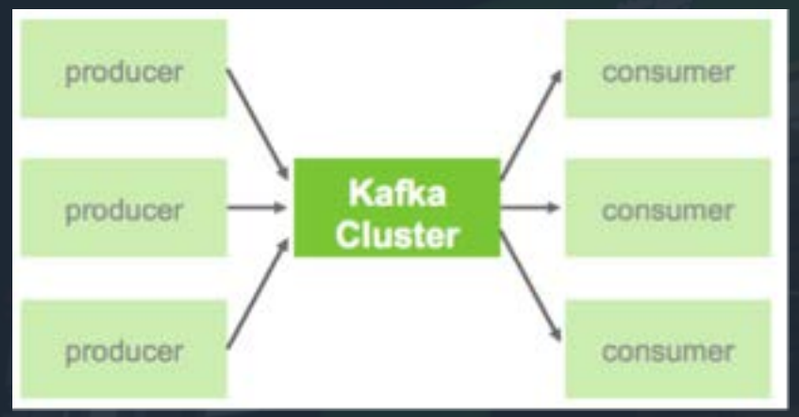
a high-throughput distributed messaging system source from LinkedIn

数据分析项目中 Producer 的首选！

Kafka源于LinkedIn，现属Apache开源项目。它在大量数据源与各类消费者程序之间构建了稳固可靠的沟通桥梁，快速、热扩展、持久化等特型让Kafka成为当下数据分析项目中 Producer 的首选消息系统。

Kafka —— 分布式消息系统

a high-throughput distributed messaging system source from LinkedIn



数据分析项目中 Producer 的首选！

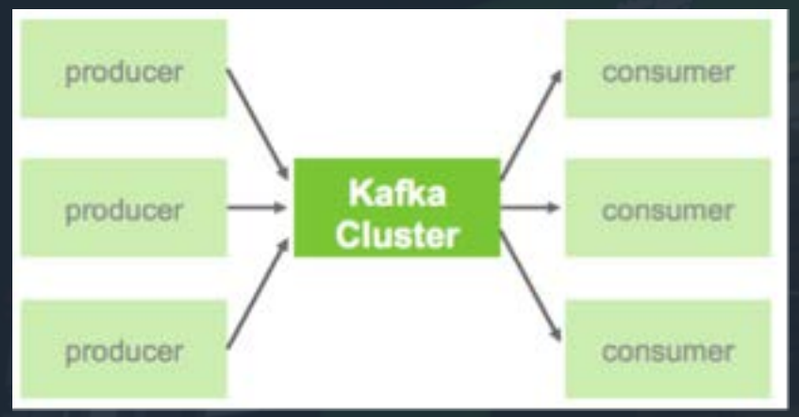
Kafka源于LinkedIn，现属Apache开源项目。它在大量数据源与各类消费者程序之间构建了稳固可靠的沟通桥梁，快速、热扩展、持久化等特型让Kafka成为当下数据分析项目中 Producer 的首选消息系统。

Model

Two models: queuing and publish-subscribe. the latter is more commonly used.

Kafka —— 分布式消息系统

a high-throughput distributed messaging system source from LinkedIn



数据分析项目中 Producer 的首选！

Kafka源于LinkedIn，现属Apache开源项目。它在大量数据源与各类消费者程序之间构建了稳固可靠的沟通桥梁，快速、热扩展、持久化等特型让Kafka成为当下数据分析项目中 Producer 的首选消息系统。

Model

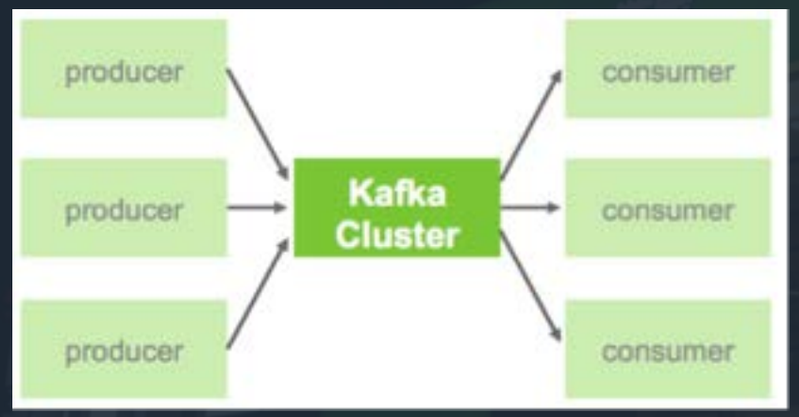
Two models: queuing and publish-subscribe. the latter is more commonly used.

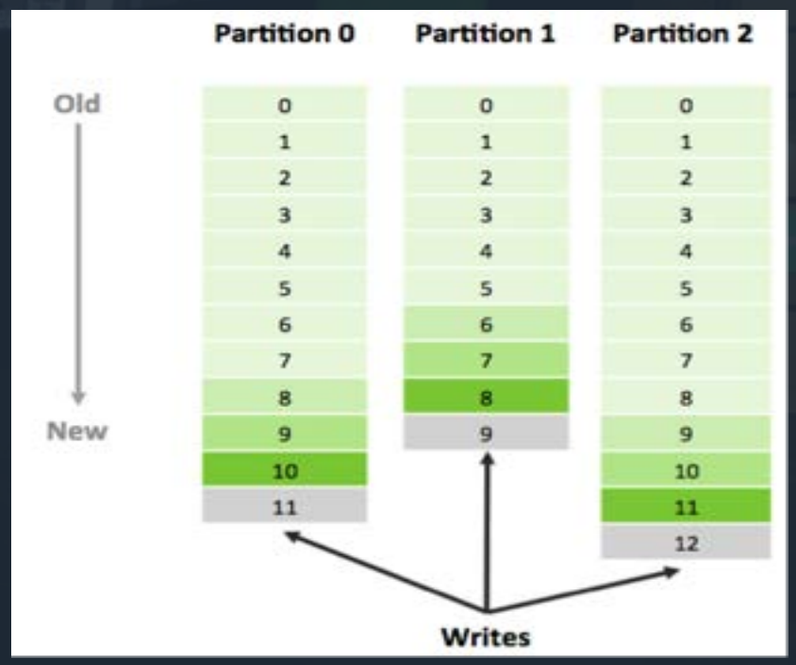
Feature

Scalability & Durability & Reliability & Performance & Fault-tolerant

Kafka —— 分布式消息系统

a high-throughput distributed messaging system source from LinkedIn





数据分析项目中 Producer 的首选！

Kafka源于LinkedIn，现属Apache开源项目。它在大量数据源与各类消费者程序之间构建了稳固可靠的沟通桥梁，快速、热扩展、持久化等特型让Kafka成为当下数据分析项目中 Producer 的首选消息系统。

Model

Two models: queuing and publish-subscribe. the latter is more commonly used.

Feature

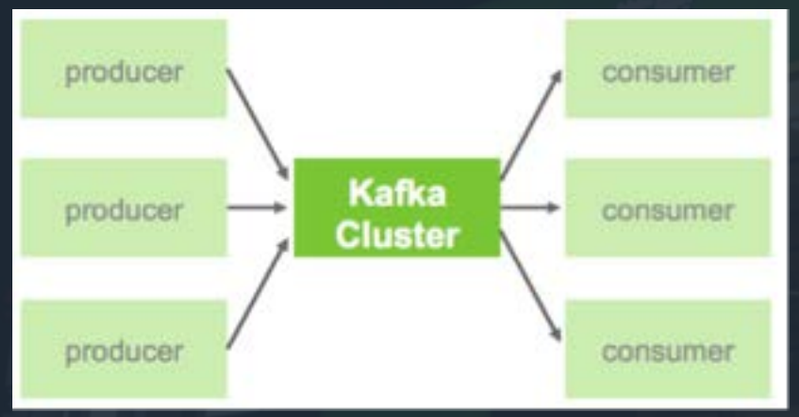
Scalability & Durability & Reliability & Performance & Fault-tolerant

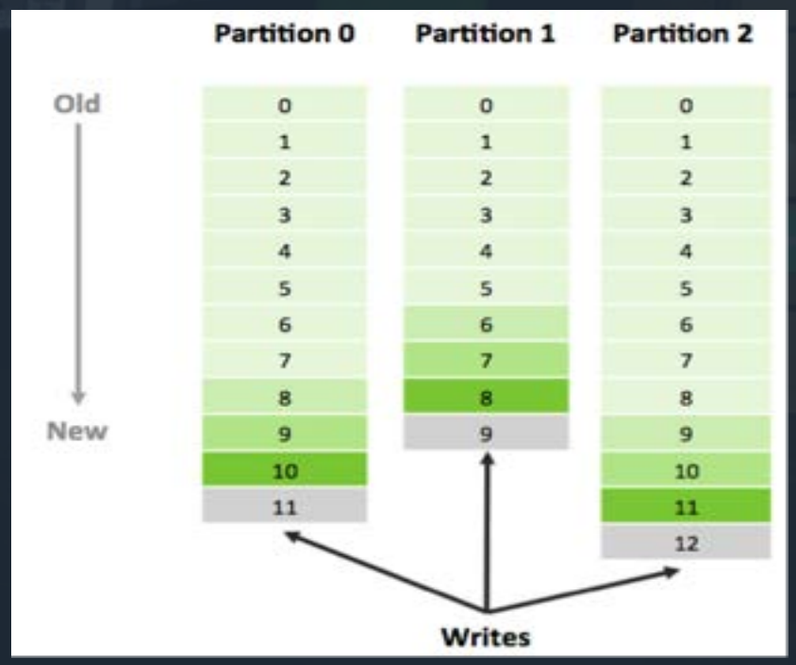
high-level abstraction

Each Topic corresponds to one or more log files. Each Topic has one or more partitions.

Kafka —— 分布式消息系统

a high-throughput distributed messaging system source from LinkedIn





数据分析项目中 Producer 的首选！

Kafka源于LinkedIn，现属Apache开源项目。它在大量数据源与各类消费者程序之间构建了稳固可靠的沟通桥梁，快速、热扩展、持久化等特型让Kafka成为当下数据分析项目中 Producer 的首选消息系统。

Model

Two models: queuing and publish-subscribe. the latter is more commonly used.

Feature

Scalability & Durability & Reliability & Performance & Fault-tolerant

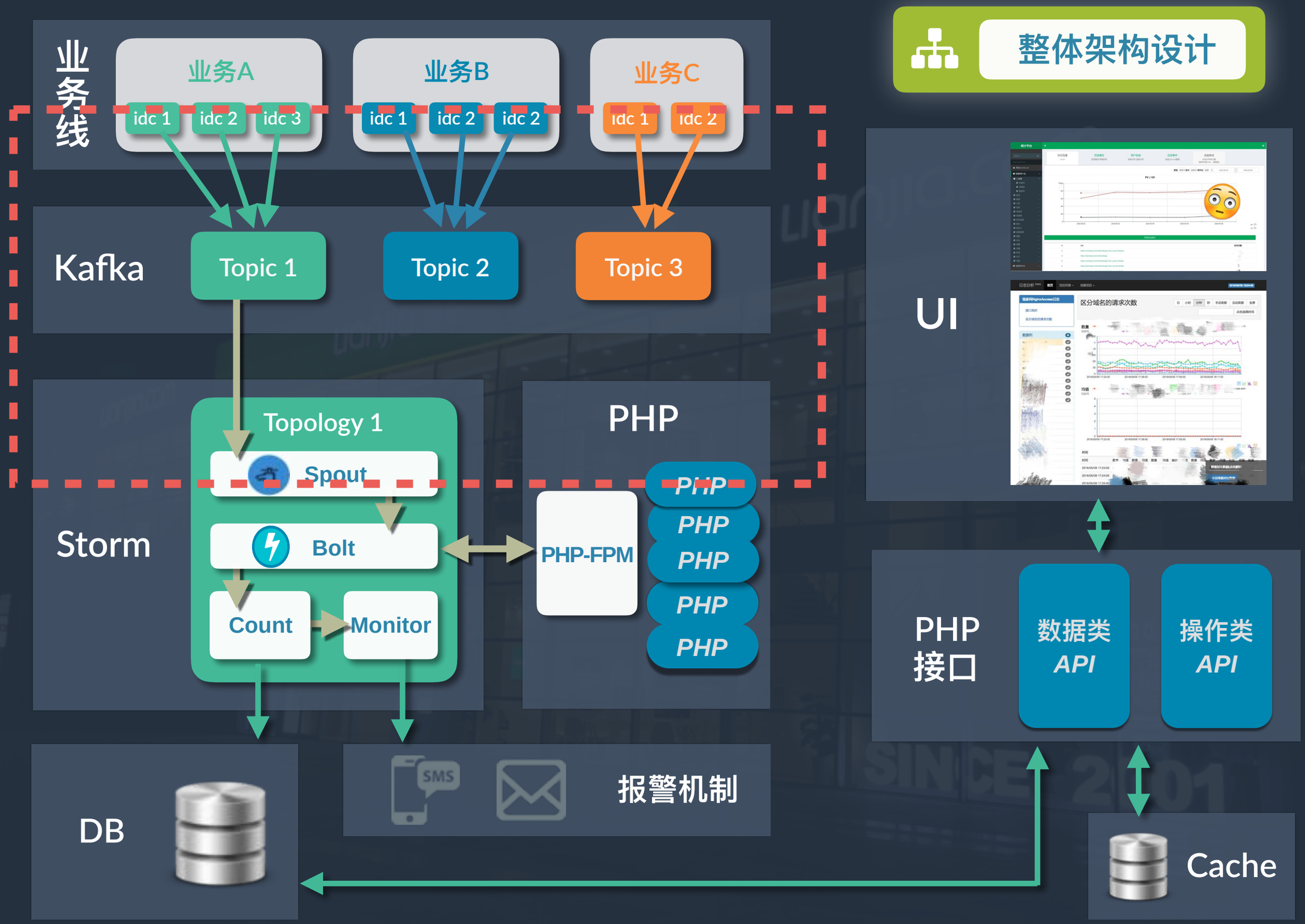
high-level abstraction

Each Topic corresponds to one or more log files. Each Topic has one or more partitions.

Ecosystem

Stream processing systems, Hadoop integration, monitoring, and deployment tools.

整体架构设计


业务线
业务A
idc 1
idc 2
idc 3
业务B
idc 1
idc 2
idc 2
业务C
idc 1
idc 2
Kafka
Topic 1
Topic 2
Topic 3
Storm
Topology 1
Spout
Bolt
Count
Monitor
PHP
PHP-FPM
PHP
PHP
PHP
PHP
PHP
PHP
UI
PHP
接口
数据类
API
操作类
API
DB
SMS
报警机制
Cache

lianjia.com
lianjia.链家
lianjia.com
www.lianjia.com
SINCE 2001

Kafka 实践中的注意项

should be noticed when using Kafka

消息状态

Kafka中的offset使用需在Consumer侧做好控制，避免消息被重复消费。

消息持久化

生产者向Kafka传递的消息会被Kafka做持久化（可配），灵活配置持久化时长，避免Kafka落地数据过大。

批量发送

可以通过配置设置一定的Buffer，批量处理消息，可以提升不错的效率，但注意截断问题。

消息时间序

每个Topic中的每个Partition内的内容是强时间顺序的，先入先出。多Partition的Topic内，Partition间可能乱序。

合理利用压缩

Kafka支持gzip、snappy、lz4三种压缩算法，压缩产生的CPU开销需要根据情况考虑是否启用压缩。

Zookeeper

Kafka依赖Zookeeper进行部署，在保障多ZK节点的同时，当心因ZK节点过多导致数据同步的时间开销。

实时分析

Real-time analysis

Storm —— 分布式实时计算系统

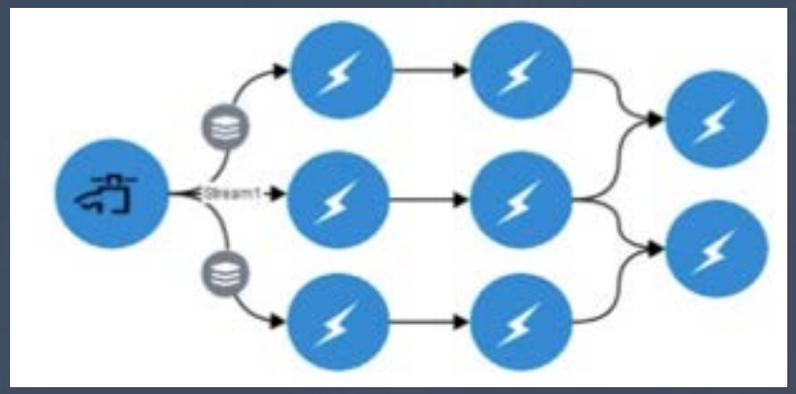
a distributed real-time computation system source from Twitter

极速处理大数据的实时计算系统！

供职于BackType的nathanmarz开发了Storm早期版本，后随着BackType被Twitter收购，Storm运用在Twitter很多实时系统中，并在Twitter期间开源成为Apache顶级项目！

Storm —— 分布式实时计算系统

a distributed real-time computation system source from Twitter



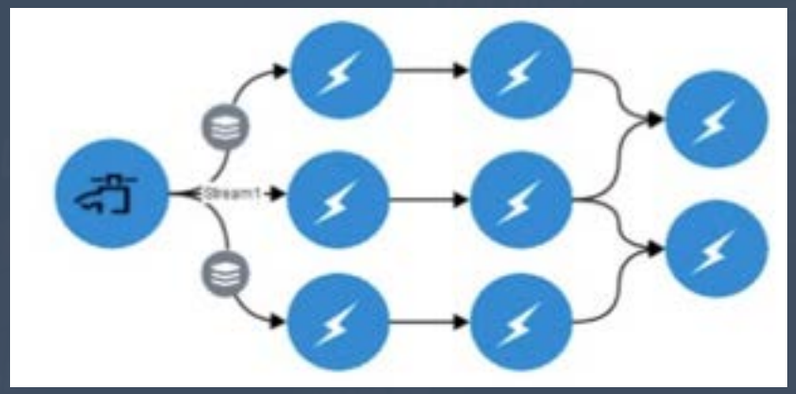
Topology of one Spout

极速处理大数据的实时计算系统！

供职于BackType的nathanmarz开发了Storm早期版本，后随着BackType被Twitter收购，Storm运用在Twitter很多实时系统中，并在Twitter期间开源成为Apache顶级项目！

Storm —— 分布式实时计算系统

a distributed real-time computation system source from Twitter



Topology of one Spout

极速处理大数据的实时计算系统！

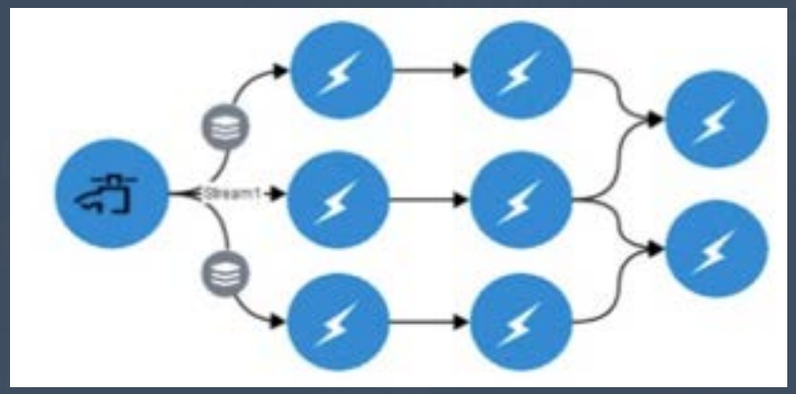
供职于BackType的nathanmarz开发了Storm早期版本，后随着BackType被Twitter收购，Storm运用在Twitter很多实时系统中，并在Twitter期间开源成为Apache顶级项目！

Feature

- Highly scalable
- Fault-tolerant
- Guarantees processing
- Language agnostic

Storm —— 分布式实时计算系统

a distributed real-time computation system source from Twitter



Topology of one Spout

极速处理大数据的实时计算系统！

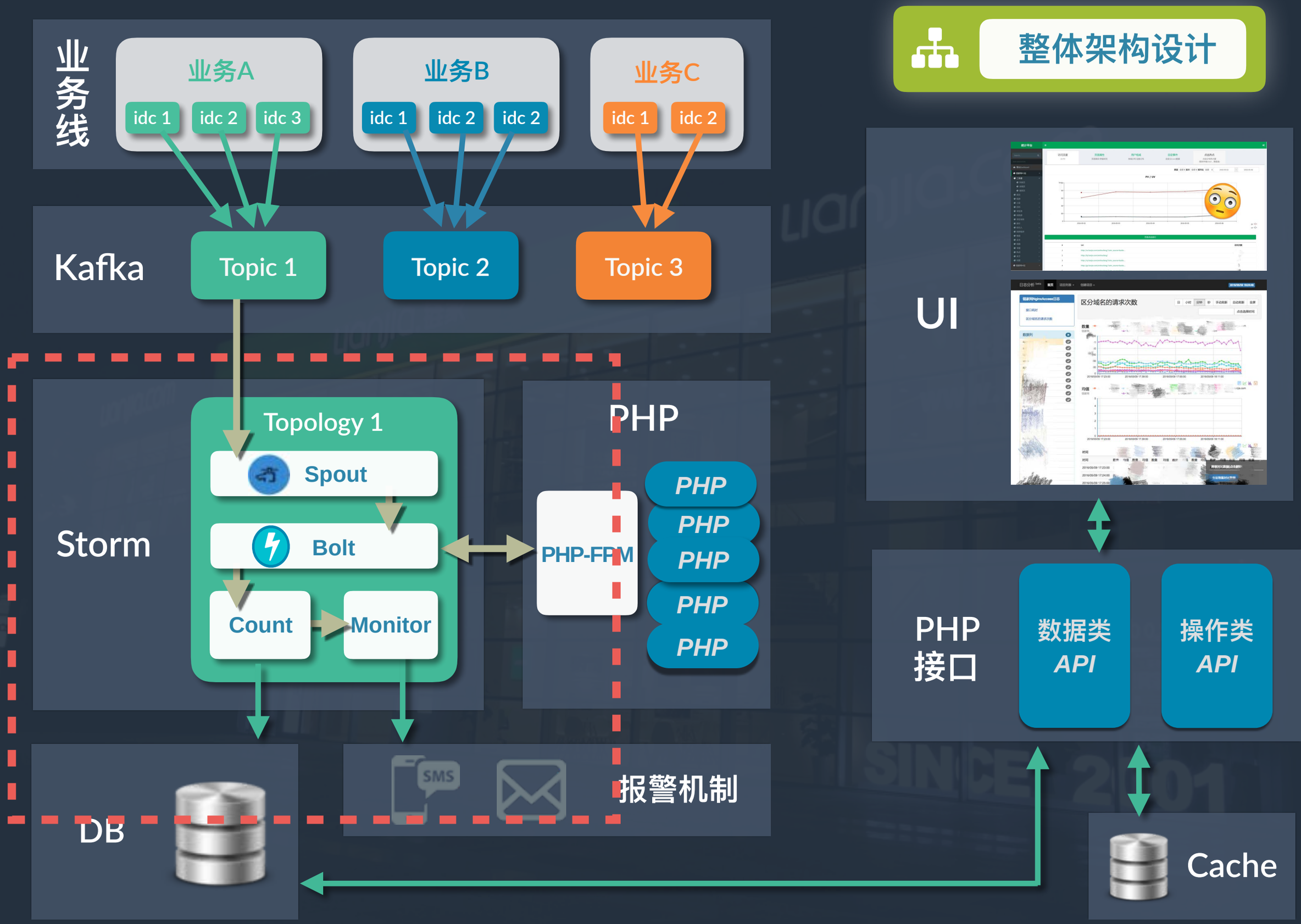
供职于BackType的nathanmarz开发了Storm早期版本，后随着BackType被Twitter收购，Storm运用在Twitter很多实时系统中，并在Twitter期间开源成为Apache顶级项目！

Feature

- Highly scalable
- Fault-tolerant
- Guarantees processing
- Language agnostic

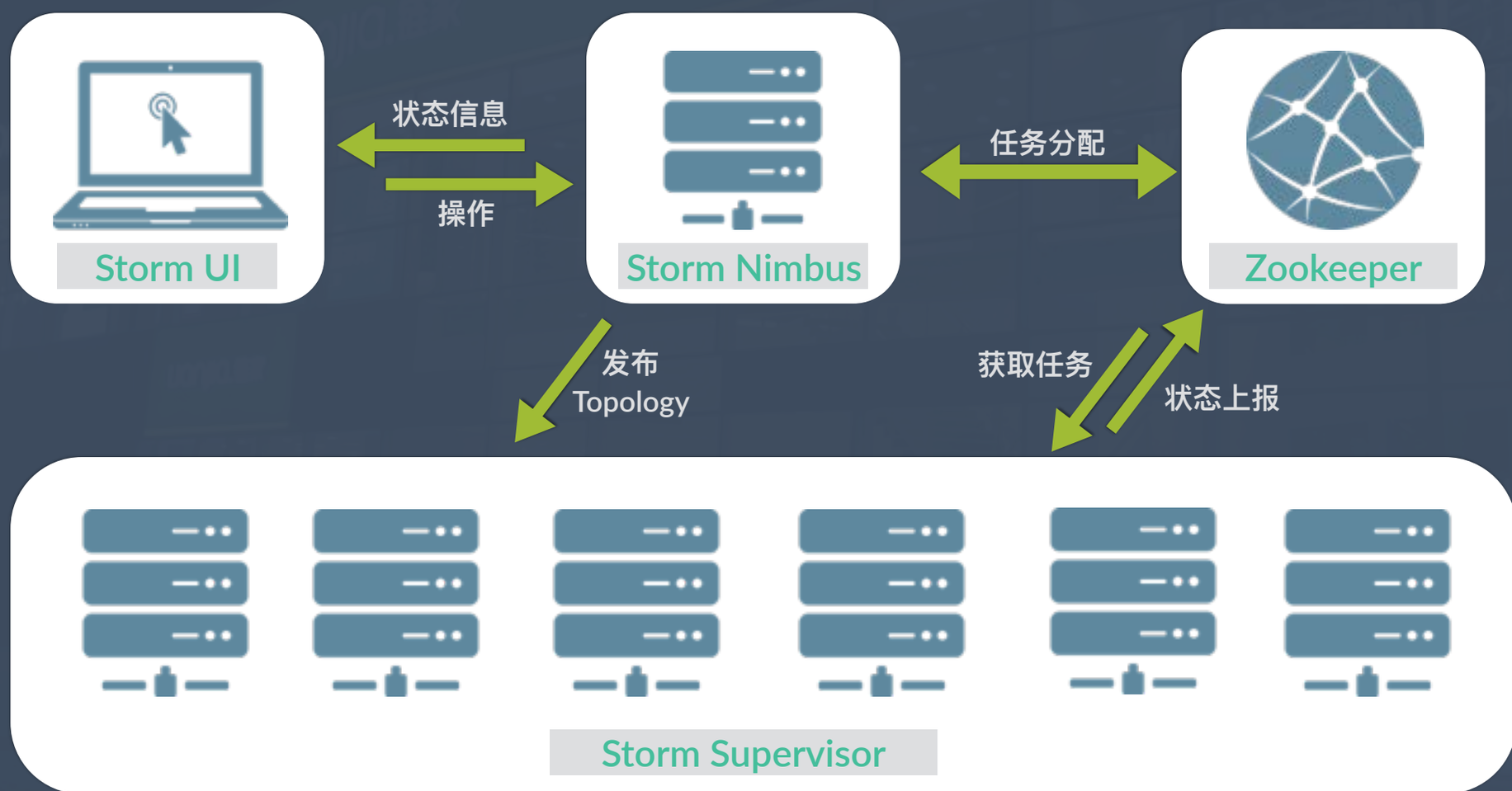
KLOUT · 爱奇艺 · twitter · yelp · Flipboard · Health Market Science · Alibaba.com · Baidu 百度 · Cerner · wego · Spotify · 淘宝网 Taobao.com · loggly

整体架构设计


业务线
业务A
idc 1
idc 2
idc 3
业务B
idc 1
idc 2
idc 2
业务C
idc 1
idc 2
Kafka
Topic 1
Topic 2
Topic 3
Storm
Topology 1
Spout
Bolt
Count
Monitor
PHP
PHP-FPM
PHP
PHP
PHP
PHP
PHP
PHP
UI
PHP
接口
数据类
API
操作类
API
SMS
报警机制
DB
Cache

Storm 集群组成

Storm cluster organization structure

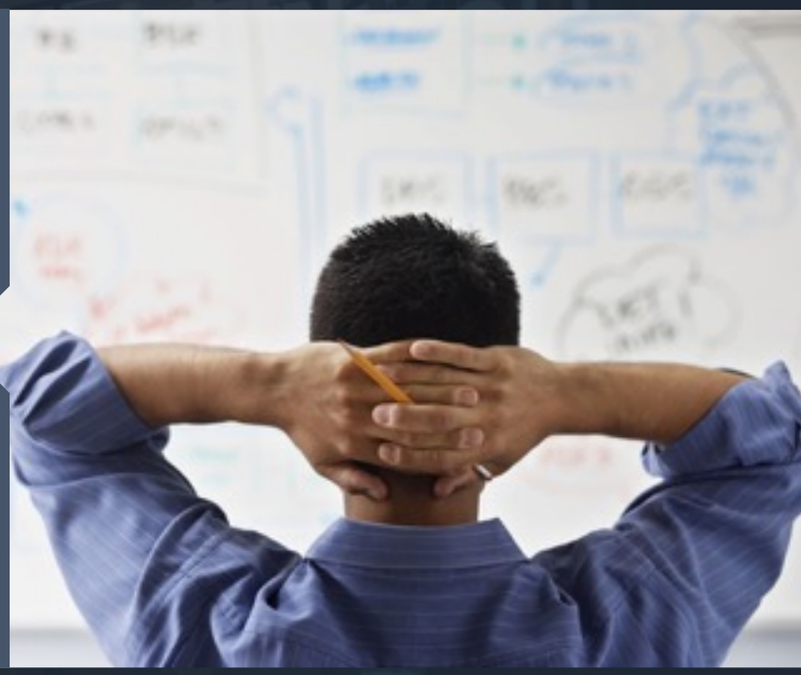


UI便于查看与操作 / Nimbus做为枢纽调度任务 / ZK负责同步信息 / Supervisor为实际运行设备

lianjia.com
lianjia.链家
lianjia.com
www.lianjia.com
SINCE 2001
Devlink
第二届PHP全球开发者大会
lianjia.链家.com

?

Q&A



Devlink 第二届PHP全球开发者大会

lianjia.com 链家.com