

SACC 第八届中国系统架构师大会  
2016 SYSTEM ARCHITECT CONFERENCE CHINA 2016

架构创新之路

# 58到家分布式服务框架 DSF(Daojia Service Framework)

任桃术@58到家

# 目录

DSF产生背景

DSF介绍

服务治理实践

# 背景

## ➤ 统一服务框架

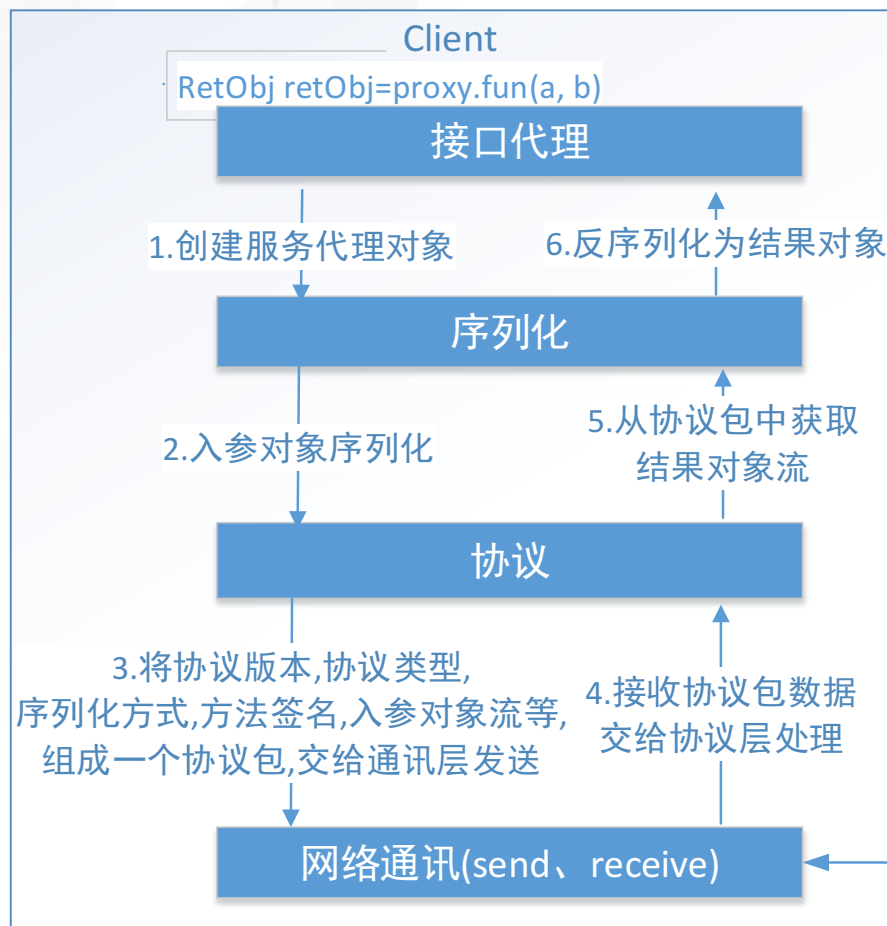
多服务框架：58同城RPC框架、Dubbo框架、.....  
维护成本高

## ➤ 统一服务治理

注册中心、监控、流控、调用跟踪

# DSF介绍

## ➤ RPC框架核心流程



# DSF介绍

## ➤ 整体架构

DSF服务治理管理平台

DSF Client

Java动态代理

负载均衡

上下文管理

网络通讯

Dtracker  
plugin

DSF注册中心

DSF序列化

DSF协议

DSF Server(容器)

异步处理

监控

权限

过滤器

网络通讯

# DSF介绍

## ➤ 协议

版本号

协议总长度

请求上下文编号

消息类型(Request,Response,Exception,Reboot...)

序列化方式(同城,DSF,Hession...)

压缩算法

加解密算法

消息体

定长包头(公共属性) + 变长包体(业务扩展)

# DSF介绍

## ➤ 序列化

四元组(参考BER(basic encoding rules)):

类型、对象总字节长度、对象属性序号、对象属性值

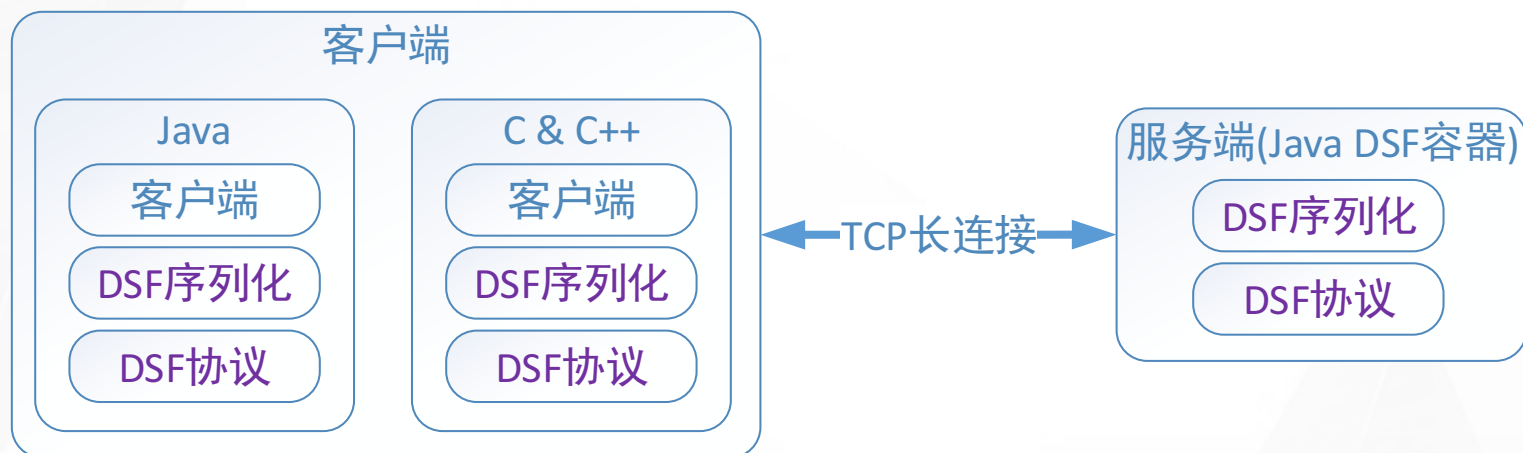
Element(自定义Class类)							
Typeid	Length	序号 sortid	属性值 Int num;	序号 sortid	Int age;	序号 sortid	String name;
567	15	1	12	2	23	3	“张三”

List对象序列化样例:

List typeid	List size	element1					element2			
589	2	typeid	length	sortid=1 Int num;	sortid=2 Int age;	sortid=3 String name;	typeid	length	sortid=1 Int num;	sortid=2 Int age;

# DSF介绍

➤ 跨语言、跨平台



客户端、服务端，使用相同的序列化和协议



# DSF介绍

## ➤ 高可用

服务多节点部署

健康检查

过载丢弃(请求阈值)

服务平滑重启

降级处理

客户端重试机制&故障转移

客户端超时处理

# DSF介绍

## ➤ 负载均衡

静态权重配置；  
服务节点动态请求超时权重调整；

## ➤ 安全性

IP黑白名单  
方法调用授权  
服务分组

# DSF介绍

## ➤ 易用性

## ✓ 运维部署

bin(启动脚本, 全局配置文件)

|---dsf(启动脚本)

|---dsf\_config.xml

|---dsf\_log4j.xml

README(相关说明)

docs(文档)

log(日志文件)

lib(DSF容器自身依赖的jar)

service

|---lib(服务依赖的公共jar)

|---deploy(DSF服务部署目录)

|---paycenter(支付中心服务)

|---paycenter.1.0.0.jar

|---dsf\_config.xml

|---dsf\_log4j.xml

|---order(订单服务)

|---order.1.0.0.jar

|---dsf\_config.xml

|---dsf\_log4j.xml

# DSF介绍

## ➤ 易用性

### ✓ 实时监控

#### **count** [ |second num |method methodName]

- \* show method call times in num seconds
- \* second : in num seconds statistics once (num default 1)
- \* method : for statistics method
- \* example : count
- \* example : count |second 3 |method getInfo

#### **time** |grep abc [ |column -tkda]

- \* show method execute time
- \* grep : condition
- \* column : show column a->all t->time k->key d->description
- \* example: time |grep getInfo
- \* example: time |grep getInfo |column -tk

#### **help**

- \* show help

#### **quit**

- \* quit monitor

#### **jvm** [option] [time] [count]

##### \*[option]:

- \* -gcutil: detection heap memory usage
- \* -class : load class
- \* -gcheap: heap memory used and committed
- \* -memory:JVM memory used
- \* -heap :Virtual Machine heap memory used
- \* -noheap:Virtual Machine noheap memory used
- \* -thread: thread counts
- \* time : [time] milliseconds apart test again
- \* count : detection [count] times
- \* example:jvm -gcutil
- \* example:jvm -gcutil 1000
- \* example:jvm -gcutil 1000 5

#### **exec**|top

##### |netstat -na

- \* exec command (at present only allow:top or netstat)
- \* example:exec |top

# DSF介绍

## ➤ 易用性

### ✓ 开发简单

服务端(接口定义、接口实现、序列化实体类)

```
@DSFServiceContract
public interface IOrderService {
    @DSFOperationContract
    public Order getOrder(long orderId);
}

@DSFServiceBehavior(lookUP="OrderService")
public class OrderService implements IOrderService{
    @Override
    public Order getOrder(long orderId) {
        // ...
        return null;
    }
}
```

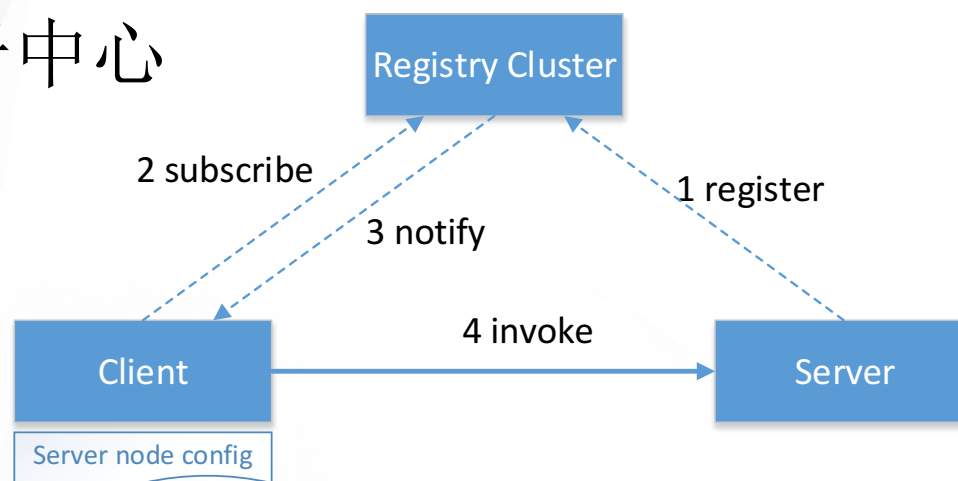
```
@DSFSerializable
public class Order {
    @DSFMember(sortId=1)
    private long orderId;
    @DSFMember(sortId=2)
    private Date orderDate;
    @DSFMember(sortId=3)
    private long userId;
    //get、set方法
}
```

客户端(配置加载、创建客户端代理、调用远程接口)

```
DSFInit.init("config/dsf.config");
IOrderService orderService = DSFProxyFactory.create(IOrderService.class, "tcp://order/OrderService");
orderService.getOrder(123456L);
```

# 服务治理

## ➤ 注册中心



- ✓ 发布 & 订阅 机制  
服务增、删节点，对服务消费方透明；  
方便做分组管理和安全策略；
- ✓ 服务健康状况监测  
TCP长连接、心跳监测；
- ✓ 高可用  
注册中心HA；  
客户端本地配置

# 服务治理

## ➤ 流控

阈值告警，超过流控阈值80%；

波动告警；

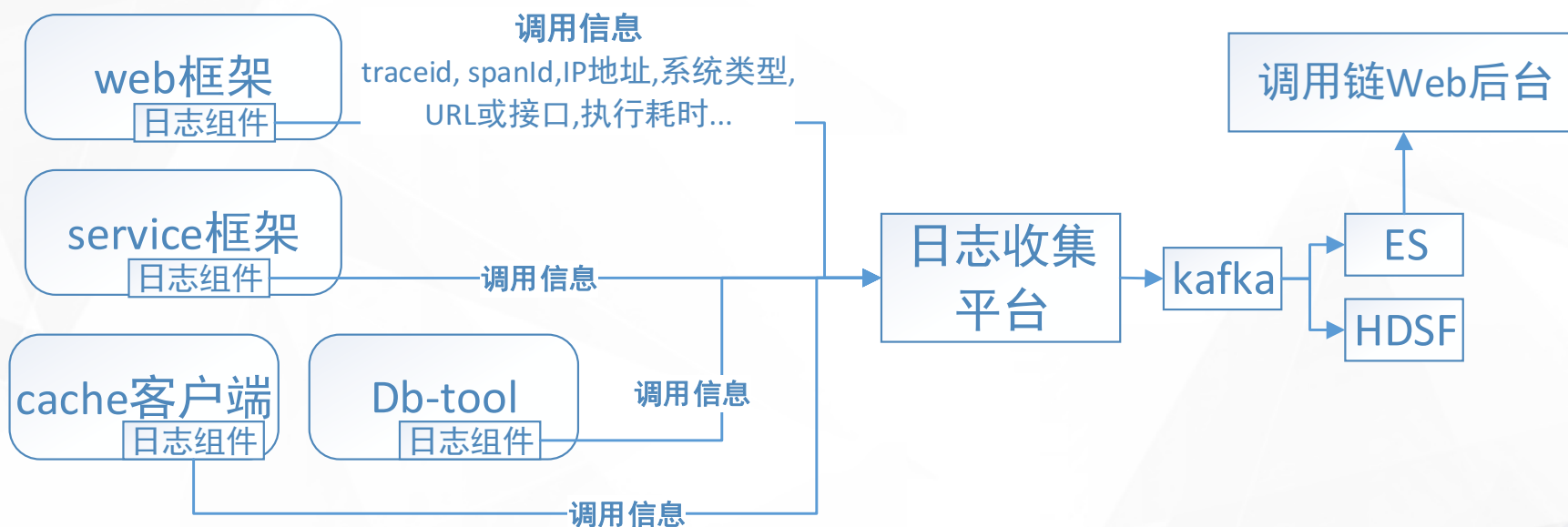
手动快速扩容(自动扩容)；

流控阈值在线调整，实时生效；

# 服务治理

## ➤ 调用跟踪系统(Dtracker)

基于日志的分布式调用跟踪系统

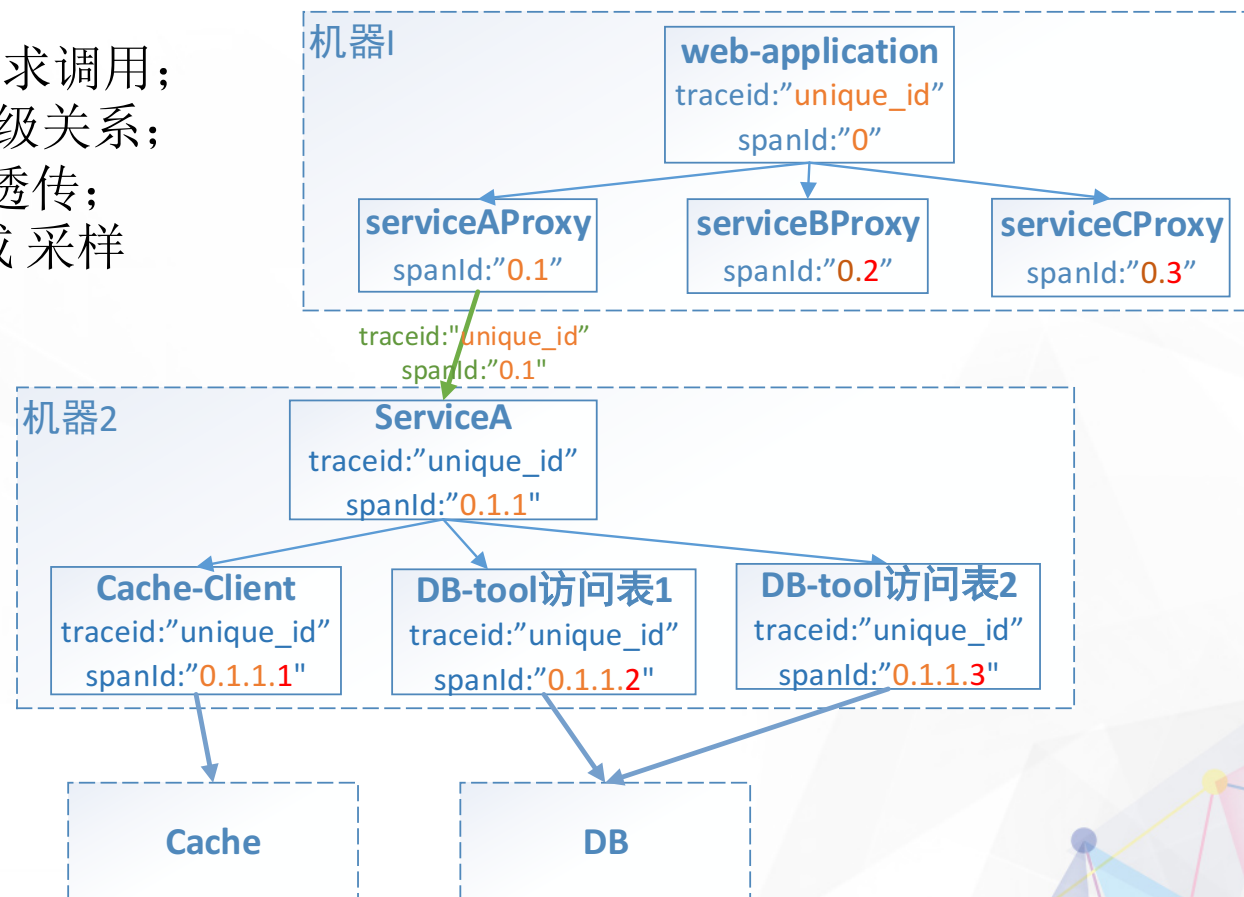




# 服务治理

## ➤ 调用跟踪系统(Dtracker)

- ✓ Traceid全局串联请求调用;
- ✓ spanId描述调用层级关系;
- ✓ traceid、spanId的透传;
- ✓ 数据采集: 全量或采样



# THANKS

SequeMedia  
盛拓传媒

IT168.com  
专注引导 10年

ChinaUnix

ITPUB  
www.itpub.net

“架构师之路” 公众号

