基于Dubbo的分布式系统架构规划（以第三方支付系统为例）



## 二、结合业务场景，对系统的应用类型进行划分：

(1)   服务子系统 ------ 账户、交易、对账、结算、打款、风控……

(2)   内部管理应用 ---- 运营、风控、会计……

(3)   对外业务应用 ---- 门户、代理商系统……

(4)   对外接入应用 ---- 网关、前置、交易接口……

(5)   定时任务应用 ---- 结算、日终、统计分析……

(6)   其它应用 -------- 对账、消息队列处理……

**服务子系统：Dubbo服务提供者**

**其它类型的应用：Dubbo服务消费者**

## 三、结合应用架构，实现分布式系统架构所需的第三方应用和中间件：

(1)  消息队列----------- ActiveMQ

(2)  分布式缓存--------- Redis

(3)  分布式文件系统 ---- FastDFS

(4)  反向代理服务器 ---- Nginx、Apache

(5)  集群与负载均衡 ---- Keepalived、HAproxy、LVS

(6)  应用服务器-------- JBoss、Tomcat

(7)  数据库------------- MySQL、Oracle、DB2

(8)  数据库分布式处理系统（集群、分库、分表） ---- Cobar

(9)  容器引擎----------- Docker

(10)系统日志管理 ------ Logstash

(11)分布式系统监控 ---- Zabbix

(12)其它 --------------- CA证书、密码键盘、防篡改系统……

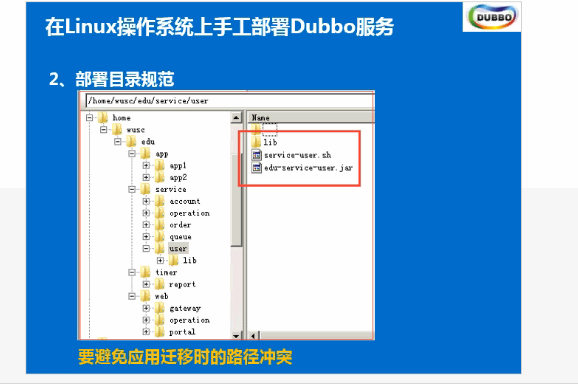
高可用、高性能、可扩展、便于运维管理、符合系统检测要求……

# [基于Dubbo的分布式系统架构-在Linux操作系统上手工部署Dubbo服务](http://blog.csdn.net/qq_26504875/article/details/51813822)

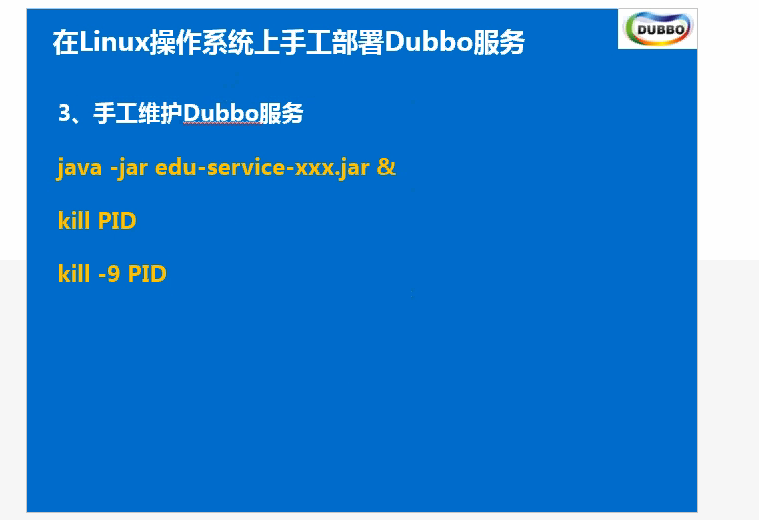
一、环境变量配置



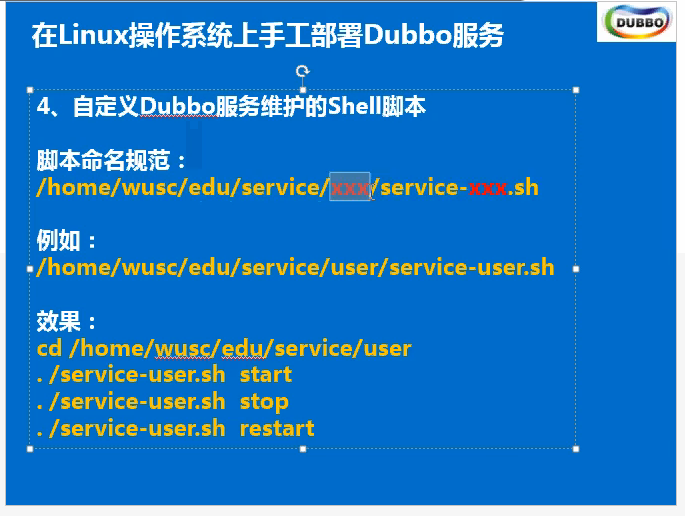
二、部署目录规范



三、手工维护Dubbo服务



四、自定义Dubbo服务维护的Shell脚本



       自定义shell脚本：

**[plain]** [view plain](http://blog.csdn.net/qq_26504875/article/details/51813822) [copy](http://blog.csdn.net/qq_26504875/article/details/51813822)

1. #!/bin/sh
3. ## java env
4. export JAVA\_HOME=/usr/local/java/jdk1.7.0\_72
5. export JRE\_HOME=$JAVA\_HOME/jre
7. ## service name
8. APP\_NAME=user
10. SERVICE\_DIR=/home/wusc/edu/service/$APP\_NAME
11. SERVICE\_NAME=edu-service-$APP\_NAME
12. JAR\_NAME=$SERVICE\_NAME\.jar
13. PID=$SERVICE\_NAME\.pid
15. cd $SERVICE\_DIR
17. case "$1" in
19. start)
20. nohup $JRE\_HOME/bin/java -Xms256m -Xmx512m -jar $JAR\_NAME >/dev/null 2>&1 &
21. echo $! > $SERVICE\_DIR/$PID
22. echo "=== start $SERVICE\_NAME"
23. ;;
25. stop)
26. kill `cat $SERVICE\_DIR/$PID`
27. rm -rf $SERVICE\_DIR/$PID
28. echo "=== stop $SERVICE\_NAME"
30. sleep 5
31. ##
32. ## edu-service-aa.jar
33. ## edu-service-aa-bb.jar
34. P\_ID=`ps -ef | grep -w "$SERVICE\_NAME" | grep -v "grep" | awk '{print $2}'`
35. if [ "$P\_ID" == "" ]; then
36. echo "=== $SERVICE\_NAME process not exists or stop success"
37. else
38. echo "=== $SERVICE\_NAME process pid is:$P\_ID"
39. echo "=== begin kill $SERVICE\_NAME process, pid is:$P\_ID"
40. kill -9 $P\_ID
41. fi
42. ;;
44. restart)
45. $0 stop
46. sleep 2
47. $0 start
48. echo "=== restart $SERVICE\_NAME"
49. ;;
51. \*)
52. ## restart
53. $0 stop
54. sleep 2
55. $0 start
56. ;;
58. esac
59. exit 0

五、服务维护对象