Orip:基于内存数据库的海量数据实时处理解决方案

李朝铭

(北京开拓天际信息技术有限公司首席架构师)

2010.4.3

目录

什么是海量数据处理？

当数据量不断突破TB、PB、EB级时，当企业关键应用开始数据大集中、应用管理大集中时，当企业数据库中心向多城市扩张时，当前的数据库体系开始面临各种发展瓶颈：

如何将单一数据库通过Cluster集群方式扩展？

如何利用Cache机制提高数据库访问速度、如何性能调优？

如何保证数据库的备份、容灾安全、同步性、一致性？

--摘自本次大会网站主页

当在线用户数达到数万甚至是数百万时，当每分钟有数十万甚至数千万SQL需要处理时，当用户请求要求在秒级甚至在毫秒级得到响应时，而满足这些需求我们的money却非常有限时……

应选用何种数据库？

数据库如何设计？

其安全可靠性如何保障？

--这是本演讲的主题

什么是内存数据库

- 什么是内存数据库？

  数据主要存储于内存中，具备数据库的基本功能

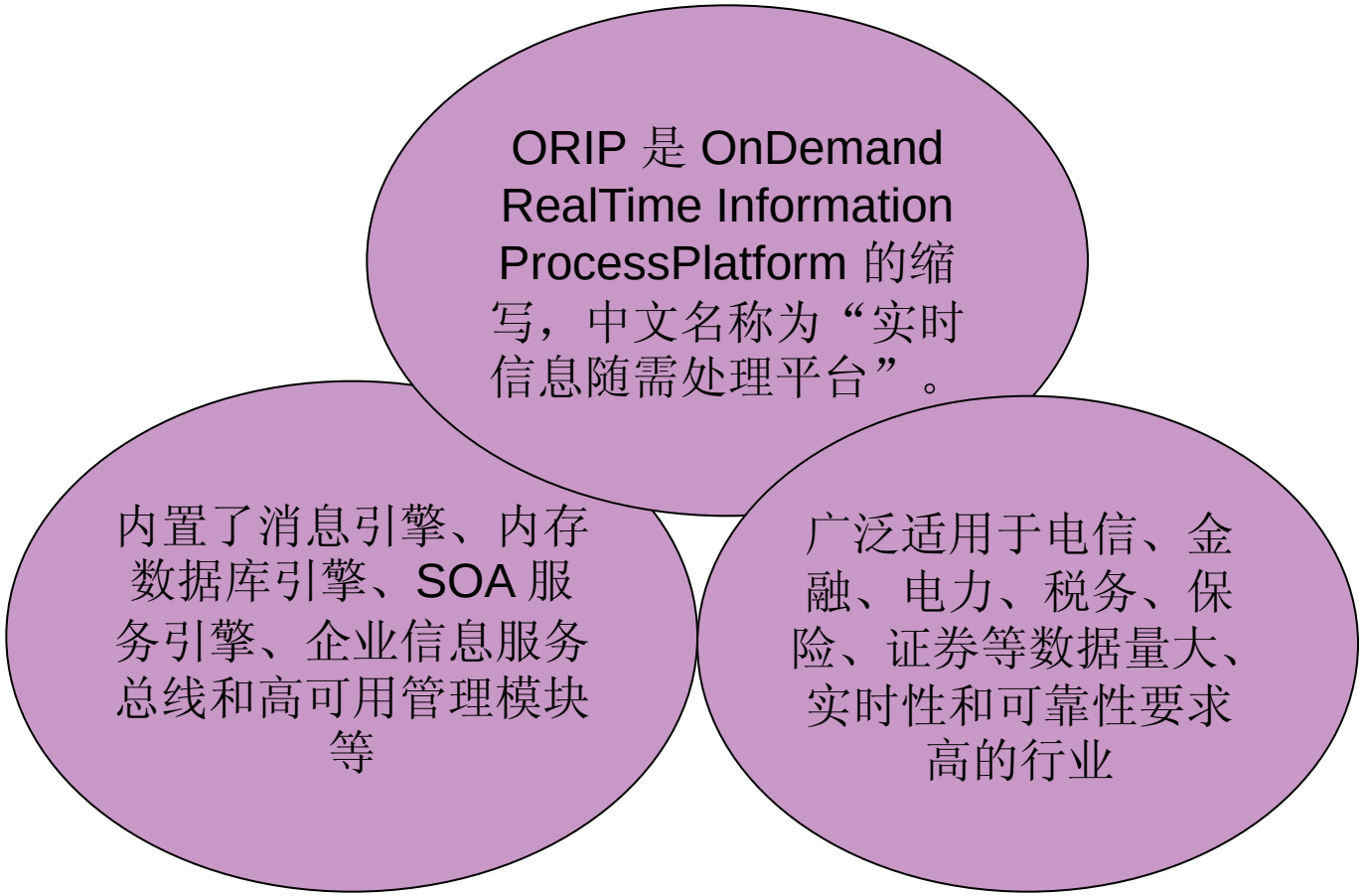
- 为什么要使用内存数据库?

  一些应用要求极高的数据处理性能，而内存数据库性能远远高于基于磁盘的关系数据库；内存价格已下降

- 常见的内存数据库有哪些？

  （Altibase、Timesten、SolidDB、BerkeleyDB、eXtremeDB …..)

什么是ORIP？



内存数据库

消息中间件

SOA服务平台

应用运行平台

动态Web服务器

HA管理软件

开拓天际
KAITONE

更高的综合性能

- 相对传统关系数据库方式，总体性能有10倍以上的提高
- DELL2950(2*4 2.5GHz CPU，16GB 内存)上并发处理平均处理能力接近每秒20万个 SQL
- 在千兆网络下，ORIP单事例每秒可同时处理5万个以上的并发HTTP请求（含内存数据库访问）

高可靠性

- Orip提供了redo日志，保证了主机或ORIP进程在意外down掉的情况下，数据能够可靠地恢复
- checkPoint机制兼顾了性能和故障恢复时间
- 支持事务和非事务模式
- 支持一个服务内多个事务穿插进行

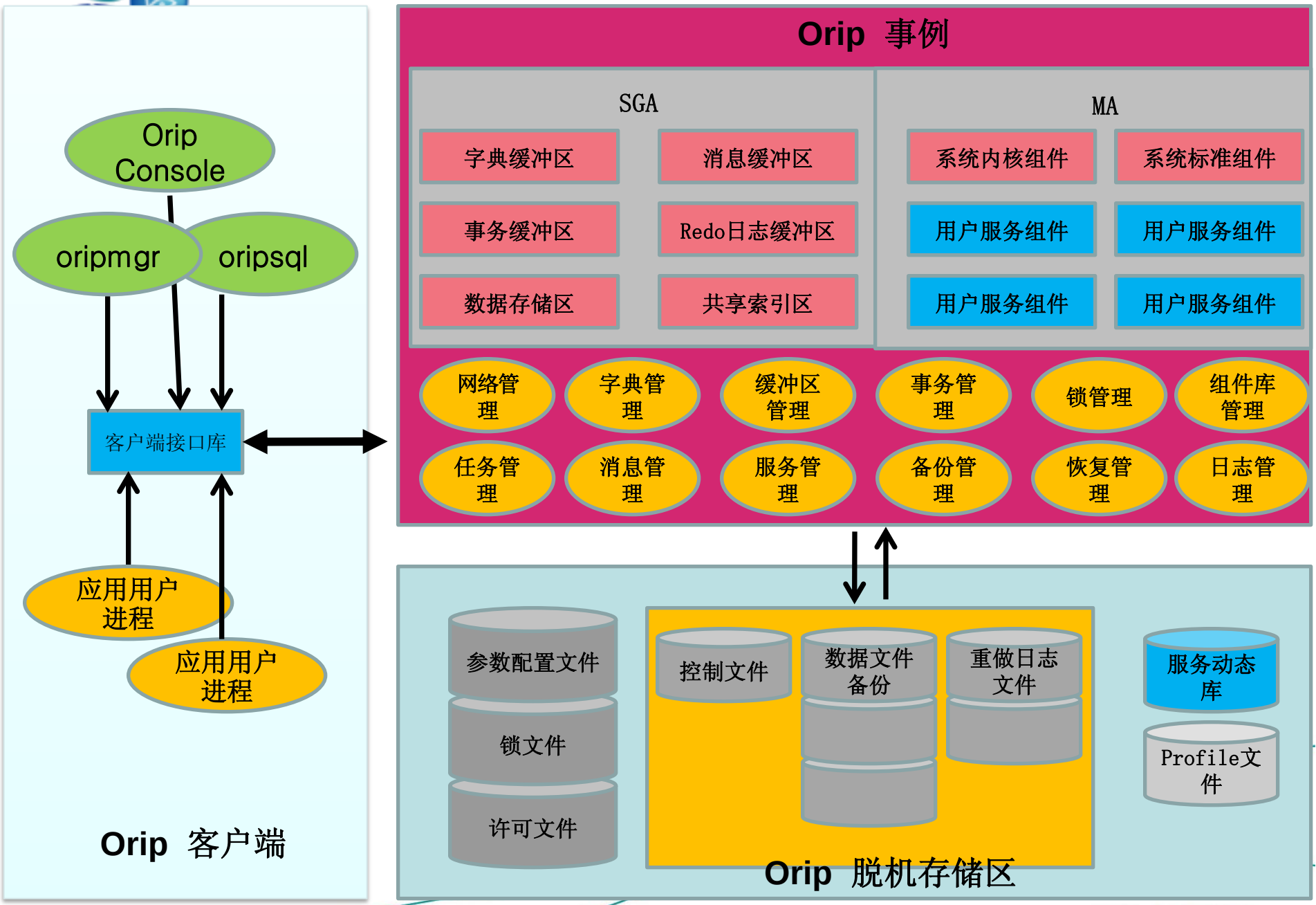
更高的可用性

- ORIP内置高可用(HA)软件特性，可支持双机热备、双机热并行、N+1备份、多机并行等
- 可在无磁盘阵列柜的情况下，实现双机数据（内存数据库）的准实时同步
- 在不启用ORIP内置HA特性的情况下，可安装oripMon守护进程，将自动重启意外down掉的orip事例

- 内存数据库简介
- orip内存数据库架构
- orip内存数据库应用开发框架
- orip内存数据库应用案例
- orip内存数据库开发

开拓天际
KAITONE

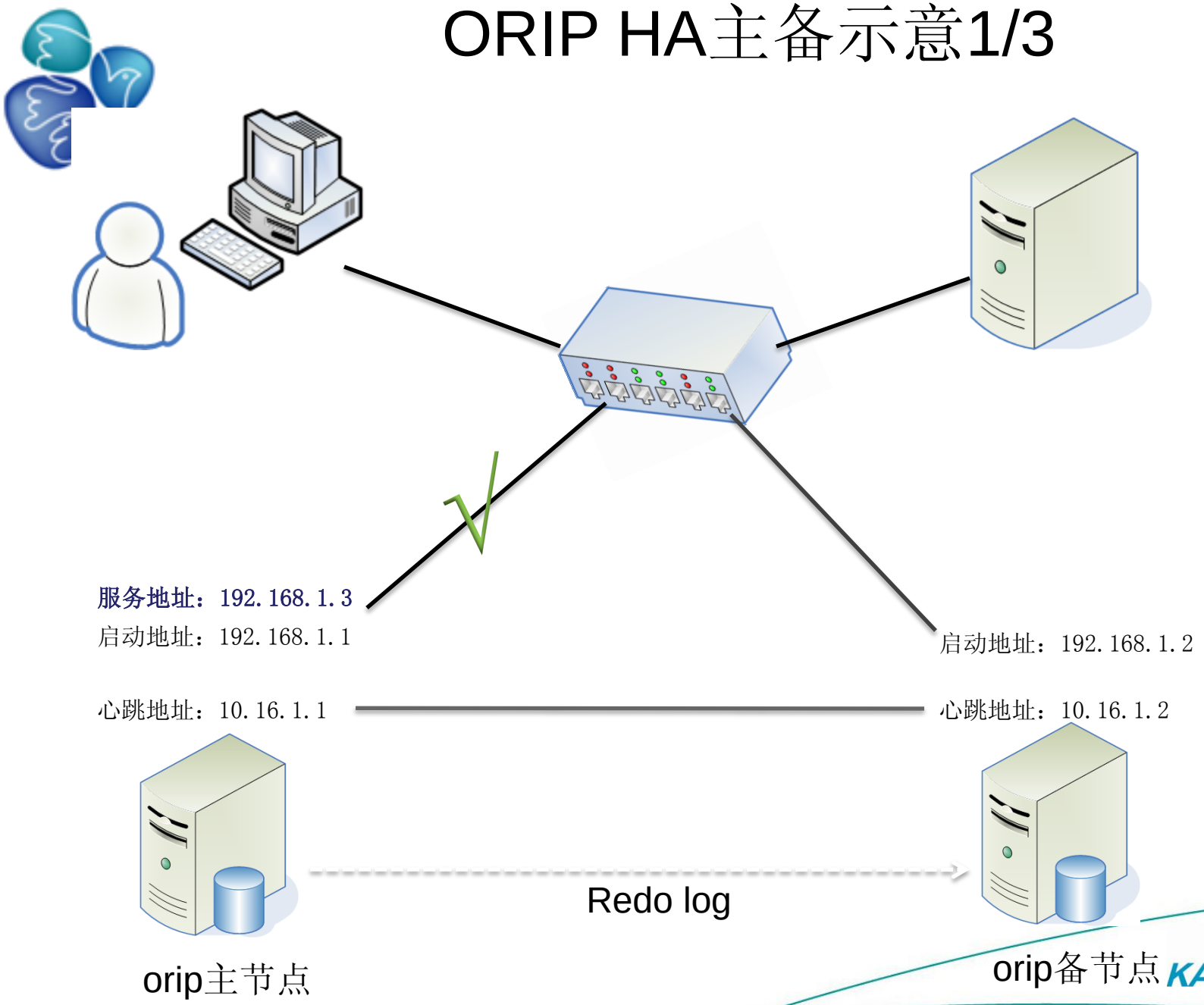
ORIP体系结构


Orip Console
oripmgr
oripsql
客户端接口库
应用用户进程
应用用户进程
Orip 客户端
Orip 事例
SGA
字典缓冲区
消息缓冲区
事务缓冲区
Redo日志缓冲区
数据存储区
共享索引区
MA
系统内核组件
系统标准组件
用户服务组件
用户服务组件
用户服务组件
用户服务组件
网络管理
字典管理
缓冲区管理
事务管理
锁管理
组件库管理
任务管理
消息管理
服务管理
备份管理
恢复管理
日志管理
参数配置文件
锁文件
许可文件
控制文件
数据文件备份
重做日志文件
服务动态库
Profile文件
Orip 脱机存储区

Orip内存数据库管理工具

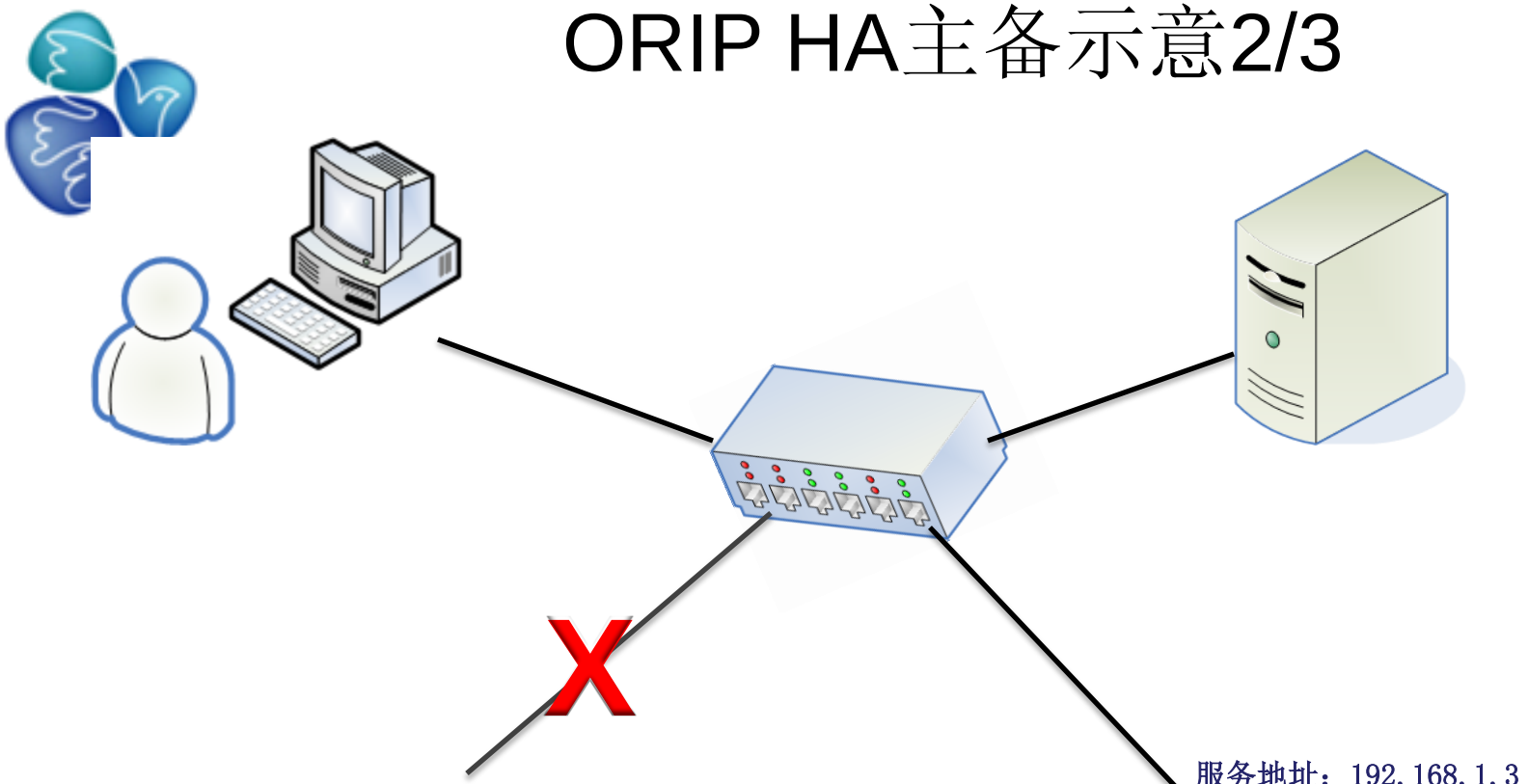
- oripmgr
- oripsql
- oripexp/oripimp
- orip console

ORIP HA主备示意1/3

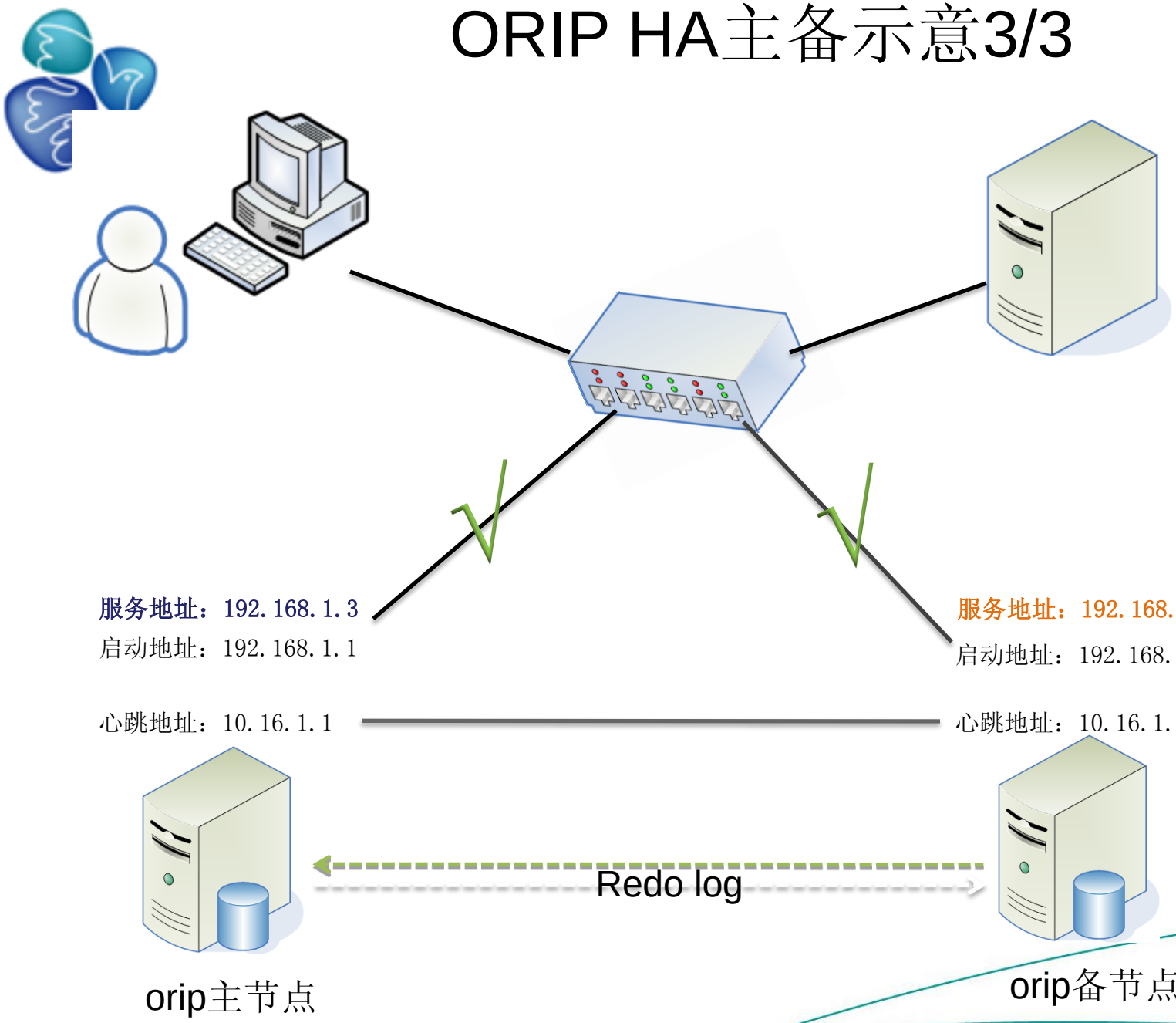


开拓天际 KAITONE

ORIP HA主备示意2/3



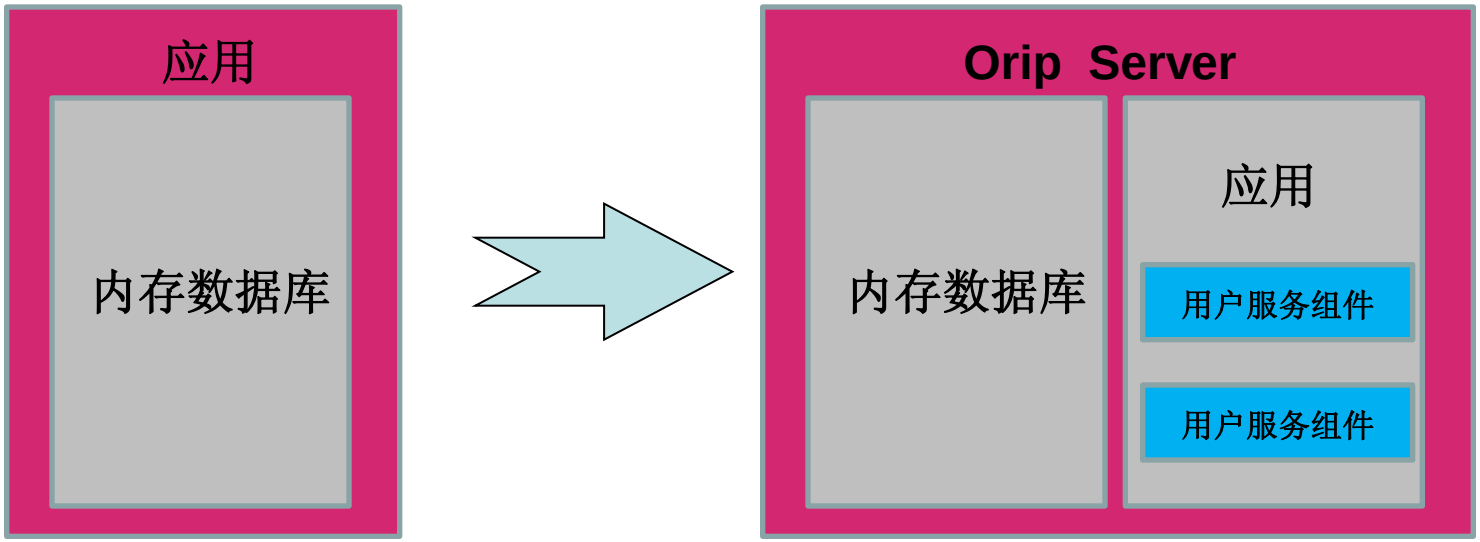
ORIP HA主备示意3/3



- 内存数据库简介
- orip内存数据库架构
- orip内存数据库应用开发框架
- orip内存数据库应用案例
- orip内存数据库开发

开拓天际
KAITONE

内存数据库应用开发架构



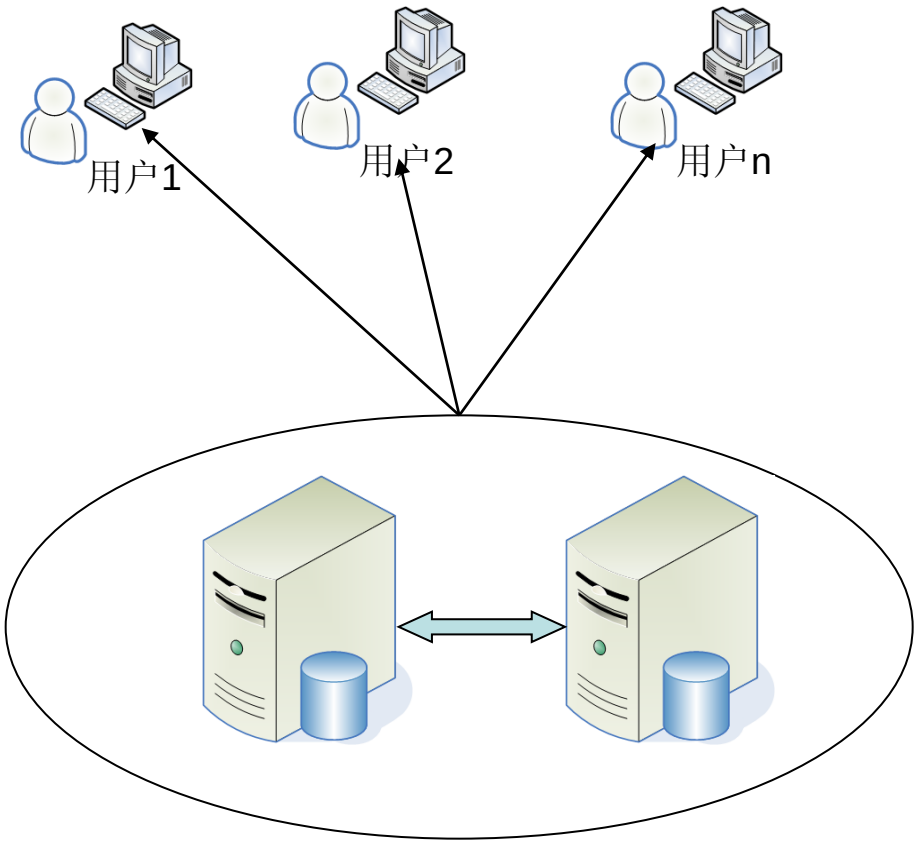
常规嵌入式内存数据库
应用开发模式

ORIP嵌入式内存数据库
应用开发模式

- 内存数据库简介
- orip内存数据库架构
- orip内存数据库应用开发框架
- orip内存数据库应用案例
- orip内存数据库开发

开拓天际
KAITONE

支持一亿用户的电信用户帐单/余额查询系统



内存表：

1.用户身份认证表

2、用户余额表

3、用户帐单表（每月一个）

服务器配置：

IBM3850 2台（每台64GB内存，4CPU），双机并行互为备份

操作系统：

Linux 64位

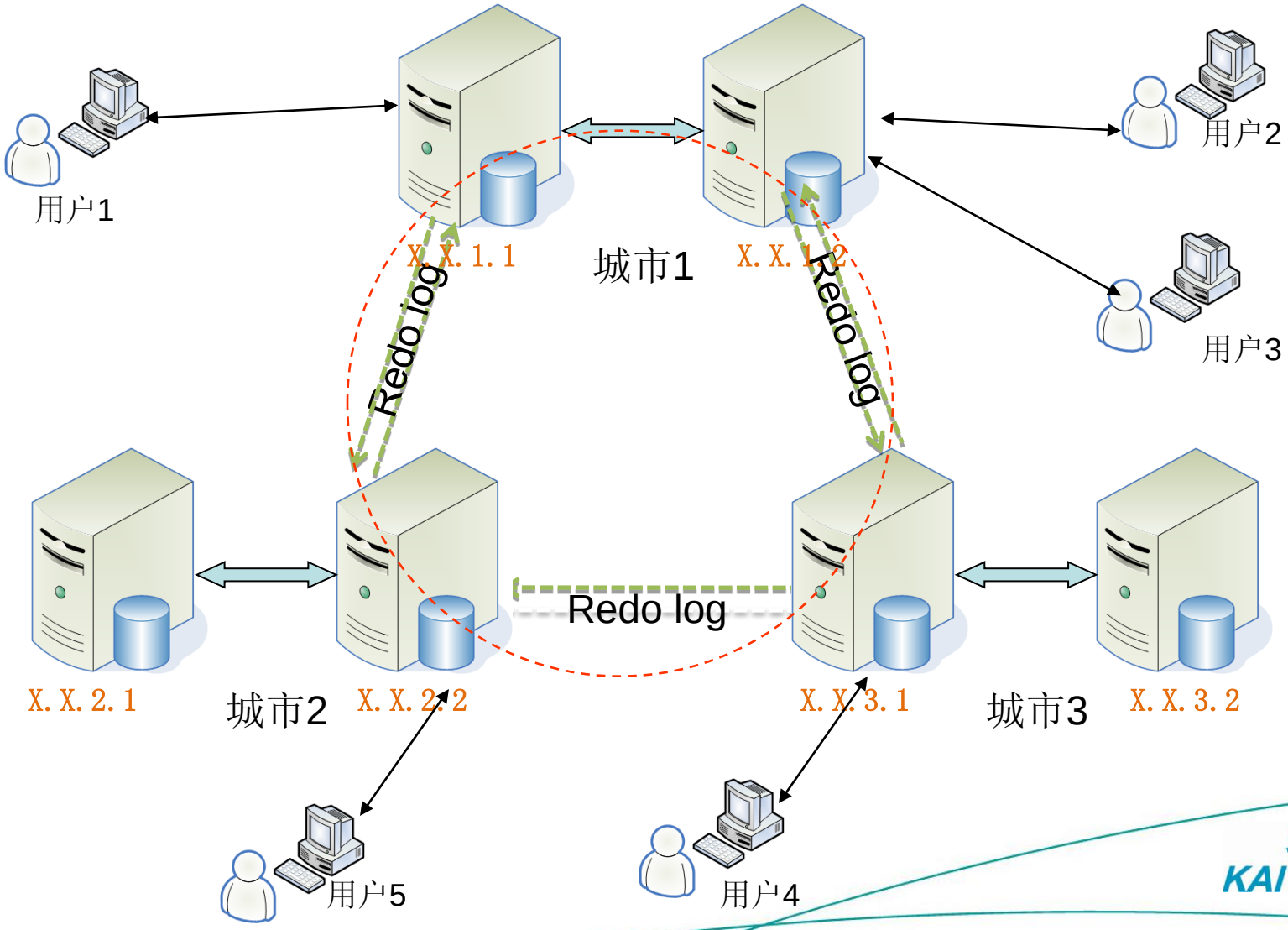
支持百万用户同时在线的IM设计1/2

```
--IM用户信息表
create table imUser
(userName varchar(16) primary key,
uid unsigned int,
petName varchar(16),
password varchar(16),
registerTime unsigned int);

--IM 用户朋友表
create table imUserFriend
(uid unsigned int primary key,
friendId unsigned int primary key,
groupId unsigned int);
```

```
--IM Uid与用户对应表
create table imUserUid
(uid unsigned int primary key,
userName varchar(16),
socketId int,
ip unsigned int,
port unsigned short int,
offlineMsg char,
status  char,
lastLoginTime unsigned int);
```

支持百万用户同时在线的IM设计2/2



- 内存数据库简介
- orip内存数据库架构
- orip内存数据库应用开发框架
- orip内存数据库应用案例
- orip内存数据库开发

开拓天际
KAITONE

Orip内存数据库开发示例1/2

```
#include<orip_server.h>
/*insert demo */
Int  serviceInsertDemo (char *inPara, char **outPara, int *retLen)
{
  char dbMsisdn[12];
  char dbName[32];
  int dbAge = 0;
  LDATETIME dbDt;
  CLI_PTR *cliPtr = NULL;
 ……
  cliPtr = (CLI_PTR *) mallocCli (100);
  cliPtr = addInsertHead (cliPtr, dbMsisdn, "mdb_demo", &trans1, &flag);
  cliPtr =    addField (cliPtr, "msisdn", dbMsisdn, strlen (dbMsisdn), DT_CHAR,
     &flag);
  cliPtr = addField (cliPtr, "name", dbName, strlen (dbName), DT_CHAR, &flag);
  cliPtr = addField (cliPtr, "age", (char *) &dbAge, 4, DT_INT, &flag);
  cliPtr = addField (cliPtr, "dt", (char *) &dbDt, 8, DT_LDATETIME, &flag);
  callInsert (cliPtr, &ret);
  ……
}
```

Orip内存数据库开发示例2/2

```
#include<orip_server.h>
/*select demo */
int
serviceSelectDemo (char *inPara, char **outPara, int *retLen)
{
  char dbMsisdn[12];
  char dbName[32];
  int dbAge = 0;
  LDATETIME dbDt;
  SELECT_PARA para[4];
  CLI_PTR *cliPtr = NULL;
  char *ptr = NULL;
  char *freePtr = NULL;
  ......
  strncpy (dbMsisdn, inPara, 11);
  dbMsisdn[11] = 0;
  para[0].val = (char *) &dbMsisdn;
  para[1].val = (char *) &dbName;
  para[2].val = (char *) &dbAge;
  para[3].val = (char *) &dbDt;
  cliPtr = (CLI_PTR *) mallocCli (100);
  cliPtr = addSelectHead (cliPtr, dbMsisdn, "mdb_demo", &flag);
  cliPtr = addFieldEmpty (cliPtr, "msisdn", DT_CHAR, &flag);
  cliPtr = addFieldEmpty (cliPtr, "name", DT_CHAR, &flag);
  cliPtr = addFieldEmpty (cliPtr, "age", DT_INT, &flag);
  cliPtr = addFieldEmpty (cliPtr, "dt", DT_LDATETIME, &flag);
```

```
cliPtr = addWhereSign (cliPtr);
ptr = callSelect (cliPtr, &ret);
freePtr = ptr;
  if (ret >= 0)
    {
     /*获取首条记录*/
      ret =
        getSelectFirstResult (&ptr, &para[0], &para[1],
&para[2], &para[3],NULL);
      char *dt = NULL;
      dt = l_dttoc1 (dbDt);
      printf
("\ndbMsisdn=%s,dbName=%s,dbAge=%d,dt=%s\n",
dbMsisdn, dbName, dbAge, dt);
      free (dt);
    }
  else if (ret < 0)
    {
      printf ("\nselect error:%d\n", ret);
      return ret;
    }
  if (freePtr)
    free (freePtr);
  return 0;
}
```

开拓天际
KAITONE

演示

谢谢！

orip.kaitone@gmail.com

开拓天际
KAITONE