jl

薛东植个人简历.doc

gumbo 项目

sql模块：
1.不重复申请内存，用静态变量
2.先判断，再申请内存
3.多个判断，先判断单个字符的（IPS状态机选最长的）
4.不用time
5.读取配置申请内存的逻辑（见pvsd的fdb）
6.相同逻辑公用函数
7.宏或者内联

增量库bug： 先创建shm，然后加载。第一次读取的时候，配置文件不在，导致读取默认值（不为零，因为还有其他的），第二次存在配置。第一次读取配置比较小，第二次比较大，因此第一次创建内存后要加载发现shm比binary小就不加载，但是第二次就加载了，而且没有重现创建shm，因为之前已经创建了。

ad_appd 全局结构体内容混乱（sql关键字）：
发现是可重新环境，因此修改配置文件，二分法定位到位置，然后猜测与之相关的长度限制

如何优化性能：
对比测试：假如注释某个函数，发现达标。那就加大连接数，看最大值为多少，然后去掉注释，减少连接数，看最小值，得到区间较大，就改。然后取怀疑的区间最大的几个改了（总和达标了），然后修改后，再测。

如何审核代码：
a) 确认审核是哪种类型： 功能正常？ 功能有问题？
b) 如果是功能正常，就是挑常见容易发错的地方看：
A）一个函数的细节
1）字符串指针，是否在特殊情况越界，是否以0结尾（strcpy_n，拼接字符串）（memchr的长度，查找双引号里面的内容）
2）匹配的操作是否没做？（free，unlock）
3）常见的语法错误：STL的erase
4）特殊情况的判断：为空？负数？读配置有个限制值
5）关键路径的效率的问题：strcmp，拷贝（内存池或静态），判断顺序是否可调整，函数消耗（状态机内联）
6）经验的东西：已有实现复用（bhchr,memchr，strmstr）？checklist?是否阻塞（下发规则）？不能用time？文件IO？……
6）多个函数共同作用：
1）资源匹配操作：申请内存后，外层退出的匹配操作
2）顺序：共享内存的重启和初始化，读写者异常退出重启，可重入（rsu），一个进程定时器初始化共享内存（第一次创建大小，第二次太大踩内存），多线程读写退出顺序保证（全局变量）
3）