

特征计算在linux平台下运行兼容性改进

之前开发平台使用windows, 使用vc作为编译器, 没有考虑到linux平台和gcc系列编译器的兼容性. 上周对兼容性进行了一定的改进, 新的版本已经加入到版本库中.

主要修改部分为下面两个部分:

1. 编译器级别的差异
2. os对构建系统影响的差异

编译器级别差异

1. g++和vc对于C标准库调用方式的不一致

在g++中如果调用 `abs` 方法, 默认只会调用C语言中对int类型进行处理的结构, 而不是C++ 中根据不同参数类型调用重载形式的 `abs` 方法.

所以下面的代码是不兼容, 甚至是bug的.

```
double d = 2.0;  
abs(d);
```

解决方法, 强制调用 `std::abs` 形式.

2. 对于右值变量放入到参数中的调用差异.

在vc中下面的代码是可以通过编译的:

```
void foo(A& a);  
A make_a();  
  
foo(make_a);    // vc中可以将一个临时变量放入到非const的参数中
```

但是在g++中, 上面的代码就会存在问题.

如果一个变量是临时变量, 语义上就不应该被改变, 所以g++的做法更加合理, 解决方法: 将不会改变数据的地方加入const

os对构建系统的影响

linux平台下使用boost的 `test framework` 需要额外的设置:

```
# disable all auto link  
add_definitions(-DBOOST_ALL_NO_LIB)
```

```
# in *nix system, you need to define BOOST_TEST_DYN_LINK
# (http://abauchu.net/blog/post/2009/12/02/Unit-tests-with-BoostTest)
if (UNIX)
    add_definitions(-DBOOST_TEST_DYN_LINK)
endif()
```

在cmake中加入了上面的限制, 在win下和linux下都可以正常找到boost.