Subject: Value size ?
Posted by mingodad on Sun, 13 Apr 2014 21:43:17 GMT
View Forum Message <> Reply to Message

I saw in several places talking about efficiency but when I look at sizeof(Value) == 48 (64bits) I'm not sure about it.

I saw that Value has a String member called "data" and also an Atomic member called "refcount" to be used in special cases, but String also has the same "refcount" for special cases, isn't it a repetition ?

I was looking at sizeof(boost::any) == 8 (64bits) and sizeof(cdiggins::any) == 16 (64bits) (http://www.codeproject.com/Articles/11250/High-Performance-D
ynamic-Typing-in-C-using-a-Repla) and wondering why we need so much memory for Value ?

It's used on collections and this size adds to the end, also I saw on database examples using ValueMap inside ValueArray.
Why construct individual maps with lots of duplicated string keys ?
Wouldn't be more efficient to have only one set of keys and an Vector<ArrayValue> let's call it ValueTable ?

```
Renderer& Renderer::operator()(const char *id, const SqlSelect& sel)
{
 ValueArray list;
 ValueMap vm;
 SqlR sql;
 sql * sel;
 while(sql.Fetch(vm))
  list.Add(vm);
 return operator()(id, list);
}
```