Subject: Re: I request the implementation of callback5
Posted by aftershock on Tue, 15 May 2018 18:26:14 GMT
View Forum Message <> Reply to Message

it does not seem to be work..

Instead , I had to do this

```
        Atomic wait_until;
 wait_until.store ( 1, std::memory_order_relaxed );
 threads[free_index].Run ( [&,a,bot1,result_mode, stat_group_id]
 {
  VectorMap<String, double> params2 = pick ( params );
  wait_until.store ( 0, std::memory_order_relaxed );

  execute_bot_in_background ( bot1, a, pick ( params2 ), result_mode, stat_group_id );


 } );

 while ( 1 )
 {
  // memory barrier for visibility semantics


  // spin wait
  if ( !wait_until.load ( std::memory_order_acquire ) )
  {
   break;
  }
 }
```

I wonder if that was safe
threads[free_index].Run (THISBACK5 (execute_bot_in_background, bot1, a, pick(params), result_mode, stat_group_id));

How was params copied? Could it happen by the time pick would transfer it.. params could go out of scope faster?

By the way, using lambda's value copying is that thread safe?