In this project, I implement an asynchronous server-client connection, a thread pool and v8 in server.

The asynchronous server-client connection is almost same to project2.

The thread pool is for server to handle several processes of searching in the data structure. I use "boost::thread_group::group.create_thread" in on_read, and use "boost::bind" to bind operate_db(), which is a function for operate database structure, including get(), put(), etc.

By reading the codes from lecture, I think there are two ways for invoke c++ functions by javascript. One is for global functions, another is for member functions. I use the member function way in my project.

For javascript codes, my thoughts is let javascript invoke c++ functions, instead of transmit intact javascript codes. And I change the protocol of REDUCE a little bit: I add line 6 for the parameter of filter.

Another interesting thing for v8. I find I have to define class in this way:

```
    class talk_to_client;
    talk_to_client *UnwrapClass(v8::Handle<v8::Object> jso);
    struct talk_to_client{
    …
    }
```

It means I need to define UnwrapClass() out of struct.
If I define in this way:

```
    class talk_to_client{
    talk_to_client *UnwrapClass(v8::Handle<v8::Object> jso);
    …
    }
```

Then I got compile errors. I'm still figuring out the reason.

I provide one mutex for each list. When initialize the lists, I push a new mutex to boost::ptr_vector, so I get 2^16 locks. Every time I visit a list, I lock the mutex of that certain list. So different threads cannot visit the same list at the same time.

At last, I understand that global variables are bad for coding. However, I know little about transmit parameters between javascript and c++, I have to temporarily define some global variables. I'll continue figure it out.