

Branch: master ▼

Find file

Copy path

[paxosstore](#) / [paxoskv](#) / [core](#) / **plog_wrapper.h** **dengoswei** - pass pins_wrapper_test;

1307944 on Aug 27, 2017

[1 contributor](#)

Raw

Blame

History



142 lines (105 sloc) 3.3 KB

```
1
2  /*
3   * Tencent is pleased to support the open source community by making PaxosStore available.
4   * Copyright (C) 2017 THL A29 Limited, a Tencent company. All rights reserved.
5   * Licensed under the BSD 3-Clause License (the "License"); you may not use this file except in compliance with the License.
6   * https://opensource.org/licenses/BSD-3-Clause
7   * Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" basis,
8   * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing
9   * permissions and limitations under the License.
10
11
12  #pragma once
13
14  #include <stdint.h>
15  #include <memory>
16  #include <tuple>
17  #include <string>
18  #include <chrono>
19  #include "pins_wrapper.h"
20
21
22  namespace paxos {
23
24  class Message;
25  class PaxosLog;
26  class PaxosInstance;
27  //class PInsAliveState;
28  //class PInsWrapper;
29
30  class PLogWrapper {
31  public:
32      // test function
33      std::tuple<int, std::unique_ptr<PInsWrapper>>
34          TestGetInstance(uint64_t msg_index) {
35          return getInstance(msg_index);
36      }
37
38      void TestSetPInsState(PInsAliveState* pins_state) {
39          pins_state_ = pins_state;
40      }
41
42      // end of test function
43
44  public:
45      PLogWrapper(
46          uint8_t selfid,
47          uint16_t member_id,
48          const std::string& key,
49          PInsAliveState* pins_state,
50          PaxosLog& plog_impl);
51
52      ~PLogWrapper();
```

```

53
54     std::tuple<int, std::unique_ptr<Message>> Step(const Message& msg);
55
56     std::tuple<
57         int,
58         std::shared_ptr<PInsAliveState>,
59         std::unique_ptr<Message>>
60     Set(uint64_t reqid, const std::string& raw_value, bool do_fast_accpt);
61
62     // <err, set_index>
63     std::tuple<
64         int,
65         std::shared_ptr<PInsAliveState>,
66         std::unique_ptr<Message>>
67     NormalSet(
68         uint64_t reqid,
69         const std::string& data, bool do_fast_accpt);
70
71     std::tuple<
72         int,
73         std::shared_ptr<PInsAliveState>,
74         std::unique_ptr<Message>>
75     PreemptSet(uint64_t reqid, const std::string& data);
76
77     std::tuple<int, std::unique_ptr<Message>> TryRedoProp();
78
79     // assistant function
80     PInsAliveState* SetPInsAliveState(PInsAliveState* pins_state);
81
82     PInsAliveState* GetPInsAliveState() const {
83         return pins_state_;
84     }
85
86     bool NeedDiskWrite() const {
87         return need_disk_write_;
88     }
89
90     bool NeedUpdateChosen() const {
91         return need_update_chosen_;
92     }
93
94     const std::string& GetKey() const {
95         return key_;
96     }
97
98     uint8_t SelfID() const {
99         return selfid_;
100     }
101
102     paxos::PaxosLog& GetPLog() {
103         return plog_impl_;
104     }
105
106     void ClearPInsAliveState() {
107         pins_state_ = nullptr;
108     }
109
110 private:
111
112     void setDiskWrite() {
113         need_disk_write_ = true;
114     }
115
116     void setUpdateChosen() {
117         need_update_chosen_ = true;
118     }

```

```
119
120     std::tuple<int, std::unique_ptr<PinsWrapper>>
121         getInstance(uint64_t msg_index);
122
123     std::tuple<int, std::unique_ptr<Message>>
124         stepInvalidIndex(const Message& msg);
125 private:
126     // key: => selfid_;
127     uint8_t selfid_ = 0;
128     // svr: => member_id_;
129     uint16_t member_id_ = 0;
130     const std::string key_;
131
132     bool need_disk_write_ = false;
133     bool need_update_chosen_ = false;
134
135     PinsAliveState* pins_state_ = nullptr;
136     PaxosLog& plog_impl_;
137 };
138
139 } // namespace paxos
140
141
```