Item 239
git_comments:

1. dump period must be greater than or equals 10 milliseconds
2. check data
3. create task
4. basic check
5. * * create new profile task * @param serviceId monitor service id * @param endpointName monitor endpoint name * @param monitorStartTime create fix start time task when it's bigger 0 * @param monitorDuration monitor task duration(minute) * @param minDurationThreshold min duration threshold * @param dumpPeriod dump period * @return task create result
6. calculate task execute range
7. check limit The duration of the monitoring task cannot be greater than 15 minutes
8. if any task time bucket in this range, means already have task, because time bucket is base on task end time
9. Each service can monitor up to 1 endpoints during the execution of tasks
10. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
11. * * @author MrPro
12. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
13. * * profile task database bean, use none stream * * @author MrPro
14. if null or empty means the task create success, otherwise get create error reason
15. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
16. get data id when create success
17. * * create profile task result * * @author MrPro
18. add service name
19. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
20. * * search profile task list * @param serviceId monitor service * @param endpointName endpoint name to monitored * @return
21. * * handle profile task queries * * @author MrPro
22. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
23. * * @author MrPro
24. * * search task list in appoint time bucket * @param serviceId monitor service id, maybe null * @param endpointName endpoint name, maybe empty * @param startTimeBucket time bucket bigger than or equals, nullable * @param endTimeBucket time bucket small than or equals, nullable * @param limit limit count, if null means query all * @return
25. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
26. * * process all profile task query * * @author MrPro
27. * * profile mutation GraphQL resolver * * @author MrPro
28. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
29. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
30. * * profile query GraphQL resolver * * @author MrPro
31. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
32. * * Profile task create need data * * @author MrPro
33. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR

CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *

34. * * @author MrPro
35. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
36. * * parse profile task data * @param data * @return
37. * * @author MrPro
38. set the system properties that can be used in test codes
39. ~ Licensed to the Apache Software Foundation (ASF) under one or more ~ contributor license agreements. See the NOTICE file distributed with ~ this work for additional information regarding copyright ownership. ~ The ASF licenses this file to You under the Apache License, Version 2.0 ~ (the "License"); you may not use this file except in compliance with ~ the License. You may obtain a copy of the License at ~ ~ http://www.apache.org/licenses/LICENSE-2.0 ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License. ~
40. substitute application.yml to be capable of cluster mode
41. !/usr/bin/env bash Licensed to the SkyAPM under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
42. !/usr/bin/env bash Licensed to the SkyAPM under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
43. verify get task list
44. create profile task
45. * * verify create profile task * @param minutesAgo * @throws Exception
46. final String swWebappPort = System.getProperty("sw.webapp.port", "32783");
47. verify create task
48. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
49. * * @author Mrpro
50. verify basic info
51. set the system properties that can be used in test codes
52. ~ Licensed to the Apache Software Foundation (ASF) under one or more ~ contributor license agreements. See the NOTICE file distributed with ~ this work for additional information regarding copyright ownership. ~ The ASF licenses this file to You under the Apache License, Version 2.0 ~ (the "License"); you may not use this file except in compliance with ~ the License. You may obtain a copy of the License at ~ ~ http://www.apache.org/licenses/LICENSE-2.0 ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License. ~
53. !/usr/bin/env bash Licensed to the SkyAPM under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
54. verify get task list
55. create profile task
56. * * verify create profile task * @param minutesAgo * @throws Exception
57. final String swWebappPort = System.getProperty("sw.webapp.port", "32783");
58. verify create task
59. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
60. * * @author Mrpro
61. verify basic info
62. set the system properties that can be used in test codes
63. ~ Licensed to the Apache Software Foundation (ASF) under one or more ~ contributor license agreements. See the NOTICE file distributed with ~ this work for additional information regarding copyright ownership. ~ The ASF licenses this file to You under the Apache License, Version 2.0 ~ (the "License"); you may not use this file except in compliance with ~ the License. You may obtain a copy of the License at ~ ~ http://www.apache.org/licenses/LICENSE-2.0 ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License. ~
64. Download MySQL connector.
65. Modify application.yml to set MySQL as storage provider.
66. !/usr/bin/env bash Licensed to the SkyAPM under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
67. !/usr/bin/env bash Licensed to the SkyAPM under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
68. verify get task list
69. create profile task

70. * * verify create profile task * @param minutesAgo * @throws Exception
71. final String swWebappPort = System.getProperty("sw.webapp.port", "32783");
72. verify create task
73. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
74. * * @author Mrpro
75. verify basic info
76. ~ Licensed to the Apache Software Foundation (ASF) under one or more ~ contributor license agreements. See the NOTICE file distributed with ~ this work for additional information regarding copyright ownership. ~ The ASF licenses this file to You under the Apache License, Version 2.0 ~ (the "License"); you may not use this file except in compliance with ~ the License. You may obtain a copy of the License at ~ ~ http://www.apache.org/licenses/LICENSE-2.0 ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License. ~
77. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
78. * * @author MrPro
79. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
80. * * @author MrPro
81. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
82. * * @author MrPro
83. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
84. * * @author MrPro
85. ~ Licensed to the Apache Software Foundation (ASF) under one or more ~ contributor license agreements. See the NOTICE file distributed with ~ this work for additional information regarding copyright ownership. ~ The ASF licenses this file to You under the Apache License, Version 2.0 ~ (the "License"); you may not use this file except in compliance with ~ the License. You may obtain a copy of the License at ~ ~ http://www.apache.org/licenses/LICENSE-2.0 ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License. ~
86. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
87. * * Profile client, use profile.graphqls, base on {@link SimpleQueryClient} * * @author MrPro
88. * * e2e profile, create profile task entity * * @author MrPro
89. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
90. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
91. * * e2e profile, create profile task result entity * * @author MrPro
92. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
93. * * @author MrPro
94. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
95. * * wrap profile task create result, for profileTaskCreation.gql * * @author MrPro
96. * * Profile task bean for e2e GraphQL test result * * @author MrPro

97. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
98. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
99. * * @author MrPro
100. * * profile query data * * @author MrPro
101. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
102. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
103. * * @author MrPro
104. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. *
105. * * @author MrPro
106. ~ Licensed to the Apache Software Foundation (ASF) under one or more ~ contributor license agreements. See the NOTICE file distributed with ~ this work for additional information regarding copyright ownership. ~ The ASF licenses this file to You under the Apache License, Version 2.0 ~ (the "License"); you may not use this file except in compliance with ~ the License. You may obtain a copy of the License at ~ ~ http://www.apache.org/licenses/LICENSE-2.0 ~ ~ Unless required by applicable law or agreed to in writing, software ~ distributed under the License is distributed on an "AS IS" BASIS, ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. ~ See the License for the specific language governing permissions and ~ limitations under the License. ~
107. add profile service implementations

git_commits:

1. **summary:** Provide thread monitor create task feature (#4145)
message: Provide thread monitor create task feature (#4145) * provide thread monitor task service to create * provide thread monitor task GraphQL implements * 1. change thread monitor field names 2. fix `getThreadMonitorTaskList` query time bucket error * provide config stream processor * update submodule `query-protocol` * resolve e2e-mysql error * remove useless storage interface method. * rename `ThreadMonitorTask` to `ProfileTask` * change e2e-profile to the top level * fix profile test analyze error * remove StringUtil#isBlank. * 1. remove create profile task duration unit 2. remove GraphQL getTask list duration field 3. add `profileTaskQueryMaxSize` in `storage` -> `elasticsearch(7)` configuration, default get 200 profile task * provide e2e different storage tests * 1. fix rat check 2. remove DurationUtils.java#toSecond 3. remove ProfileTaskQueryEs7DAO * fix e2e code format error * provide es6 and es7 storage e2e tests * change e2e profile es module artifactId Co-authored-by: kezhenxu94 <kezhenxu94@163.com>
label: code-design

github_issues:

1. **title:** [FEATURE] Performance profile at method level in production environment
body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
2. **title:** [FEATURE] Performance profile at method level in production environment
body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
3. **title:** [FEATURE] Performance profile at method level in production environment
body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature:

https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)

4. **title:** [FEATURE] Performance profile at method level in production environment
   body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
5. **title:** [FEATURE] Performance profile at method level in production environment
   body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
6. **title:** [FEATURE] Performance profile at method level in production environment
   body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
7. **title:** [FEATURE] Performance profile at method level in production environment
   body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
8. **title:** [FEATURE] Performance profile at method level in production environment
   body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
9. **title:** [FEATURE] Performance profile at method level in production environment

body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)

10. **title:** [FEATURE] Performance profile at method level in production environment
    body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
11. **title:** [FEATURE] Performance profile at method level in production environment
    body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
12. **title:** [FEATURE] Performance profile at method level in production environment
    body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)
13. **title:** [FEATURE] Performance profile at method level in production environment
    body: Please answer these questions before submitting your issue. - Why do you submit this issue? - [ ] Question or discussion - [ ] Bug - [ ] Requirement - [x] Feature or performance improvement ___ I wrote about the thread monitoring proposal in the mail list. The doc is now complete to introduce this feature: https://docs.google.com/document/d/1rxMf1WN3PaFaZp7r8JmtwfdkmjLTcFW_ETAZv5FIU-s/edit Because the amount of development is relatively large, I will divide the following steps into the development work: Step 1: Save thread monitoring tasks to storage and corresponding graphQL 1. Define thread monitoring DAO, graphql, mainly including adding task interface 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage Step 2: Downstream task to sniffer 1. Define the cache interface (CacheDAO), which can be used to query the cached data. Implementing JDBC storage 2. Complete queries in Elasticsearch and Elasticsearch 7 3. Create a CacheService and start a scheduled task to write data to the cache 4. Define the protocol file (query command) for OAP to communicate with the sniffer, and implement the gRPC server handler Step 3: sniffer executes commands to monitor traces 1. Start timing tasks for command reading, and define the gRPC protocol (for snapshot upload) 2. Monitor trace generation and report data 3. Write Testcase (JUnit) to ensure that snapshots can be created and reported after receiving the command Step 4: OAP saves thread snapshot 1. Define the receive module, define the interface for adding snapshots in the DAO, and receive the data save request passed by the sniffer 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Write Testcase, save operation based on received data Step 5: Query data and display graphql 1. Define query interface and graphql for querying task list, related trace list, and snapshot 2. Implementing JDBC storage 3. Implementing Elasticsearch storage 4. Implementing Elastic-search 7 storage 5. Complete test code (e2e-cluster-test-runner)

github_issues_comments:

1. Branch `profile` will be created after 6.6.0 release. PRs of all steps should send to there, after all features tested and e2e test setup, I will submit a `un-squash` merge PR to the master. This is a very important feature, @apache/skywalking-committers, welcome and look forward all of you could take part in and help on review. Design doc should be ready.
2. To all in this thread @mrproliu has made big progress in this feature. We miss the last step, then we could deliver the test env to UI team, @Fine0830. I have submitted the CFP about this for SRECon and JavaZone, I hope I could at least get one passed.
3. > To all in this thread > > @mrproliu has made big progress in this feature. We miss the last step, then we could deliver the test env to UI team, @Fine0830. > I have submitted the CFP about this for SRECon and JavaZone, I hope I could at least get one passed. hmm...I don't understand what the test env is and what we need to do in UI.
4. > hmm...I don't understand what the test env is and what we need to do in UI. Test env is the one(by using docker), we will give you to set up locally, then you could easily query . UI design is included in the Google doc, here are the links 1. Main page, https://bwd5l7.axshare.com/#id=8sgf47&p=thread-monitor-page&g=1 1. Add task page, https://bwd5l7.axshare.com/#id=61ddhs&p=add-task-modal&g=1 @Fine0830 The UI design is just a proposal and matched the proposal, for details, after you read, we could talk more.

5. All backend related PRs have been merged, and e2e tests passed. @mrproliu Next step, let's provide a local test image with data to the UI team(@Fine0830). Notice, the TTL should be set as long as possible to avoid the profile data being deleted.
6. The data is ready for the UI team(@Fine0830) to access. - By running this command, you can start the OAP server locally, which already contains the data content: `docker run -d -p 12800:12800 -p 9090:9090 mliu111477/profile-graphql:0.0.5` - Data query related GraphQL has generated query related statements through postman(exclude create). [You can refer to this file](https://www.getpostman.com/collections/8f84a74da2bb9bb6f118). After importing, you can use.
7. @Fine0830 Above inform should be ready for your local run and test. UI draft is there if you have any issue, please ping me, we could discuss, especially about the UI design. ï¿¼
8. I will review the UI design draft again tomorrow, and try to finalize with @mrproliu
9. The UI design has been finished. - Add task model: ![add-task-modal](https://user-images.githubusercontent.com/3417650/74311813-40cacb80-4dab-11ea-9d20-9568546cdc89.png) - Profile main page: ![thread-monitor-page](https://user-images.githubusercontent.com/3417650/74311842-53450500-4dab-11ea-9c1e-ff21c26736f5.png)
10. @Fine0830 This is a backend performance benchmark for analysis button action. https://cwiki.apache.org/confluence/display/SKYWALKING/Performance+profile+analyze+benchmark Give the tips `The analysis usually takes 2-10s, please wait.` when user confirms to do analysis action at the UI.
11. Hi, @Fine0830. I missed the profiled segment query in GraphQL, now has added. I create the new docker images, please running this command to run it. `docker run -d -p 12800:12800 -p 9090:9090 mliu111477/profile-graphql:0.0.8` The postman has also added this query: https://www.getpostman.com/collections/8f84a74da2bb9bb6f118

github_pulls:

1. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
2. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
3. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
4. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
5. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
6. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: test
7. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
8. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
9. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
10. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
11. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
12. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
13. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
14. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
15. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
16. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
17. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
18. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: code-design
19. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
20. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: test
21. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
22. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql

label: code-design
23. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
24. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
25. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: code-design
26. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
27. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: code-design
28. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
29. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
30. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
31. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
32. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: code-design
33. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
34. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: documentation
35. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
36. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
37. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
38. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
39. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
40. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
41. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
42. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
43. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
44. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
45. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
46. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
47. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
48. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
49. **title:** Provide thread monitor create task feature
body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
label: code-design
50. **title:** Provide thread monitor create task feature

body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql

51. **title:** Provide thread monitor create task feature
    body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql
52. **title:** Provide thread monitor create task feature
    body: Please answer these questions before submitting pull request - Why submit this pull request? - [ ] Bug fix - [x] New feature provided - [ ] Improve performance - Related issues #4104 ___ ### New feature or improvement - Follow step one, create and query thread monitor task according to graphql

github_pulls_comments:

1. # [Codecov](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=h1) Report > Merging [#4145](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=desc) into [master](https://codecov.io/gh/apache/skywalking/commit/f64f3c14332b2c3808d86167d4b03ff096359aac?src=pr&el=desc) will **decrease** coverage by `0.17%`. > The diff coverage is `0%`. [![Impacted file tree graph](https://codecov.io/gh/apache/skywalking/pull/4145/graphs/tree.svg?width=650&token=qrILxY5yA8&height=150&src=pr)](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=tree) ```diff @@ Coverage Diff @@ ## master #4145 +/- ## == - Coverage 27.03% 26.85% -0.18% == Files 1147 1156 +9 Lines 25141 25309 +168 Branches 3641 3674 +33 == Hits 6797 6797 - Misses 17739 17907 +168 Partials 605 605 ``` | [Impacted Files](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=tree) | Coverage Î” | | |---|---|---| | [...kywalking/oap/server/core/query/DurationUtils.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItY29yZS9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5d2Fsa2luZy9vYXAvc2VydmVyL2NvcmUvcXVlcnkvRHVyYXRpb25Vd
   | `0% <Ã¸> (Ã¸)` | :arrow_up: | | [...ing/oap/server/core/source/DefaultScopeDefine.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItY29yZS9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5d2Fsa2luZy9vYXAvc2VydmVyL2NvcmUvc291cmNlL0RlZmF1bHRTY
   | `0% <Ã¸> (Ã¸)` | :arrow_up: | | [...walking/oap/server/core/storage/StorageModule.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItY29yZS9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5d2Fsa2luZy9vYXAvc2VydmVyL2NvcmUvc3RvcmFnZS9TdG9yYWdlT
   | `0% <Ã¸> (Ã¸)` | :arrow_up: | | [...alking/oap/query/graphql/GraphQLQueryProvider.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItcXVlcnktcGx1Z2luL3F1ZXJ5LWdyYXBocWwtcGx1Z2luL3NyYy9tYWluL2phdmEvb3JnL2FwYWNoZS9za3l3YWxraW5nL29hc
   | `0% <0%> (Ã¸)` | :arrow_up: | | [...lking/oap/query/graphql/resolver/ProfileQuery.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItcXVlcnktcGx1Z2luL3F1ZXJ5LWdyYXBocWwtcGx1Z2luL3NyYy9tYWluL2phdmEvb3JnL2FwYWNoZS9za3l3YWxraW5nL29hc
   | `0% <0%> (Ã¸)` | | | [...ng/oap/query/graphql/resolver/ProfileMutation.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItcXVlcnktcGx1Z2luL3F1ZXJ5LWdyYXBocWwtcGx1Z2luL3NyYy9tYWluL2phdmEvb3JnL2FwYWNoZS9za3l3YWxraW5nL29hc
   | `0% <0%> (Ã¸)` | | | [...oap/server/core/query/ProfileTaskQueryService.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItY29yZS9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5d2Fsa2luZy9vYXAvc2VydmVyL2NvcmUvcXVlcnkvUHJvZmlsZVRhc2
   | `0% <0%> (Ã¸)` | | | [...skywalking/oap/server/core/CoreModuleProvider.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItY29yZS9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5d2Fsa2luZy9vYXAvc2VydmVyL2NvcmUvQ29yZU1vZHVsZVByb3ZpZ
   | `0% <0%> (Ã¸)` | :arrow_up: | | [...rage/plugin/jdbc/h2/dao/H2ProfileTaskQueryDAO.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItc3RvcmFnZS1wbHVnaW4vc3RvcmFnZS1qZGJjLWhpa2FyaWNwLXBsdWdpbi9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5
   | `0% <0%> (Ã¸)` | | | [...king/oap/server/core/query/entity/ProfileTask.java](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree#diff-b2FwLXNlcnZlci9zZXJ2ZXItY29yZS9zcmMvbWFpbi9qYXZhL29yZy9hcGFjaGUvc2t5d2Fsa2luZy9vYXAvc2VydmVyL2NvcmUvcXVlcnkvZW50aXR5L1Byb2
   | `0% <0%> (Ã¸)` | | | ... and [19 more](https://codecov.io/gh/apache/skywalking/pull/4145/diff?src=pr&el=tree-more) | | ------ [Continue to review full report at Codecov](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=continue). > **Legend** - [Click here to learn more](https://docs.codecov.io/docs/codecov-delta) > `Î” = absolute <relative> (impact)`, `Ã¸ = not affected`, `? = missing data` > Powered by [Codecov](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=footer). Last update [f64f3c1...0c7617a](https://codecov.io/gh/apache/skywalking/pull/4145?src=pr&el=lastupdated). Read the [comment docs](https://docs.codecov.io/docs/pull-request-comments).
2. This is for step 1?
3. The e2e tests are keeping failure, we need to figure out why first.
4. > This is for step 1? Yes, this is for task creation and query.
5. I have a suggestion for `e2e-profile `. I think we should add `ThreadMonitorTask` to`e2e-cluster e2e-mysql e2e-single-service` instead of adding a new e2e.
6. **body:** > I have a suggestion for `e2e-profile `. I think we should add `ThreadMonitorTask` to`e2e-cluster e2e-mysql e2e-single-service` instead of adding a new e2e. At present, it's only graphql in OAP to add query test, but the interaction in sniffer will be integrated later. I think it's better to use a new E2E
   label: test
7. @kezhenxu94 I can't find anything related to the e2e failure in the log, even no error logs from OAP. Could you check why? Or do we miss some logs?
8. > @kezhenxu94 I can't find anything related to the e2e failure in the log, even no error logs from OAP. Could you check why? Or do we miss some logs? The failures of the E2E tests are due to the missing update of the submodule (protocol), thus missing the newly-added file `query-protocol/profile.graphqls`, @mrproliu please update the submodule in this PR
9. The failure should be discoverable and reproducible in your local machine, if you make sure that your codes are the same with the remote repo @mrproliu
10. @kezhenxu94 He sent me this directly, that is why I asked. ![image](https://user-images.githubusercontent.com/5441976/71555072-71a4aa80-2a62-11ea-9421-91aa47fb1b89.png)
11. > The failure should be discoverable and reproducible in your local machine, if you make sure that your codes are the same with the remote repo @mrproliu OK, I have submitted it.
12. Oh, you didn't update the submodule, strange, why log can't show this?
13. > Oh, you didn't update the submodule, strange, why log can't show this? The failure is due to timeout and the e2e-container is killed by the plugin, so we have no chance to `cat` the logs
14. > The failure is due to timeout and the e2e-container is killed by the plugin, so we have no chance to cat the logs So, if we face the OAP bootstrap issue, then this error shows up without the logs. Could I make this as a conclusion of a possible case? Could we make this a part of e2e test faq?
15. > > I have a suggestion for `e2e-profile `. I think we should add `ThreadMonitorTask` to`e2e-cluster e2e-mysql e2e-single-service` instead of adding a new e2e. > > At present, it's only graphql in OAP to add query test, but the interaction in sniffer will be integrated later. I think it's better to use a new E2E But `es` and `mysql` is not covered, need to be added? @wu-sheng @kezhenxu94
16. > So, if we face the OAP bootstrap issue, then this error shows up without the logs. Could I make this as a conclusion of a possible case? Yes I agree. Because bootstrap failure should be always stable to reproduce, and should be discovered when establishing a new E2E case, my assumption is that the contributor should verify locally, and guaranteed the codes are all pushed to remote repo, unlike this case :) > Could we make this a part of e2e test faq? Sure, or maybe as an important note to contribute new E2E cases
17. > Sure, or maybe as an important note to contribute new E2E cases Could you submit a separated PR about this?
18. **body:** Before the further change, please make sure we have consistent name. Let's remove thread monitor in all names, only keep profile in name, including Graphql protocol. They are used randomly.
    label: code-design

github_pulls_reviews:

1. **body:** This should be top level e2e test, like `Compatibilities` test.
   label: test
2. Why do we need both methods?
3. **body:** They have very similar parameter, but different logic? It is hard to understand.
   label: code-design

4. they are using different logic. `getTaskListSearchOnStartTime`: It will use on creating a new task to check limit, Each service can monitor up to 1 endpoints during the execution of tasks. search time range on task start time. `getTaskList`: It will use on the UI level GraphQL searching. search time range on time bucket. At present, I don't have any good way to optimize these two methods.
5. Ok. I will change it after the e2e task failed issue resolved.
6. **body:** These two names show no difference. The storage implementor must be confused.
   label: code-design
7. The point is storage should consider data query only. You could query the list, then check. I still donâ€™t see the difference. You are talking using in different places. But this is wrong as two separated methods. You should provide a method a query existing task list based on given conditions, that is all. This method is not matching the UI query or create task. It is a storage interface.
8. **body:** There is already `StringUtil#isEmpty`. This is duplicated.
   label: code-design
9. Reading this makes me feel, we should set the duration in minute level only, and remove the step in the protocol. @kezhenxu94 @arugal What do you think?
10. I think we need a `sourceBuilder.size(limit);` here, right? @kezhenxu94 Otherwise, ES query will only get top 10?
11. Yes, we get 10 docuemnts only by default.
12. Profile should be tested in every storage. H2, MySQL, es6, es7.
13. **body:** > Reading this makes me feel, we should set the duration in minute level only, and remove the step in the protocol. @kezhenxu94 @arugal What do you think? Make sense to me, but `minute` level seems too large in time span, what about making it `second` level?
    label: code-design
14. > There is already `StringUtil#isEmpty`. This is duplicated. They're different (trivial though) IMO, `StringUtil.isEmpty(" \t") == false` while `StringUtil.isBlank(" \t") == true`
15. **body:** That's true. I think to get 10 doc it's enough. Should I add a size limit?
    label: documentation
16. I highly doubt this is the purpose of creating this API..
17. Why 10 is enough? I don't remember we limit the task amount.
18. Maybe I should do String#trim on GrapthQL resolver?
19. Why do you need a trim? Who will send an empty string?
20. According to https://github.com/apache/skywalking-query-protocol/blob/master/profile.graphqls#L25-L26, this duration is for the task, I think minute makes more sense.
21. This threshold for segment(nothing related to span) is minDurationThreshold, refer to https://github.com/apache/skywalking-query-protocol/blob/master/profile.graphqls#L30
22. Oh, I just think is using on the task creation limit check, forget it will use on GraphQL. Should I add a limit and from param in the GraphQL?
23. You could have a limit in DAO interface level, but for UI, if we want to set the limit, we should set paging in the GraphQL parameter
24. Because if we set a limit in UI without paging, people would not see the full list forever. I still think the full list is enough for UI as the profiling task should not a long list.
25. One more thing, as we have TTL for profile task, we could remove the duration here, https://github.com/apache/skywalking-query-protocol/blob/master/profile.graphqls#L68. We should always show the full list. Any suggestion? @arugal @kezhenxu94
26. Do we need to support order by `time`?
27. Yes, should order by start time `DESC`
28. > Why do you need a trim? Who will send an empty string? @wu-sheng As long as the string is typed by the users, it's totally possible that it will include blank characters, or even unprintable/invisible characters, we're suffering from this because the users copy-and-pastes from else where (especially Windows users, `\r` is painful), another solution is to trim and validate the input in the frontend, but we may have other user interface, CLI I mean
29. If that is the concern, `trim` the data in the GraphQL service level, this kind of blank issue clearly is related to humans, not to codes.
30. **body:** I think this method is not necessary as the duration unit has been removed from the protocol
    label: code-design
31. Does this DAO have any different with ES6's?
32. They are the same. I deleted, use `ProfileTaskQueryEsDAO` on the es7 module provides.

jira_issues:

jira_issues_comments: