ogibayashi / fluentd-esper-demo

branch: master fluentd-esper-demo / src / main / java / com / mayoya / realtime / EsperSubscriber.java

Ogibayashi Hironori on Mar 16, 2013 - Some changes in benchmarking.

0 contributors

120 lines (101 sloc) | 3.244 kb Raw Blame History

```
1  package com.mayoya.realtime;
2
3  import org.zeromq.ZMQ;
4  import com.espertech.esper.client.EPServiceProvider;
5  import com.espertech.esper.client.EPRuntime;
6  import org.msgpack.template.Template;
7  import java.net.Socket;
8  import com.espertech.esper.client.EPServiceProviderManager;
9  import org.msgpack.MessagePack;
10 import org.msgpack.unpacker.Unpacker;
11 import java.util.List;
12 import org.msgpack.type.Value;
13 import org.msgpack.unpacker.Converter;
14 import static org.msgpack.template.Templates.TString;
15 import static org.msgpack.template.Templates.TValue;
16 import static org.msgpack.template.Templates.tMap;
17 import static org.msgpack.template.Templates.tList;
18 import com.espertech.esper.client.EPException;
19 import com.espertech.esper.client.EPAdministrator;
20 import java.util.Map;
21 import java.util.HashMap;
22 import java.util.Set;
23 import java.io.IOException;
24 import org.msgpack.template.Templates;
25 import java.util.Properties;
26 import org.msgpack.unpacker.BufferUnpacker;
27
28
29 public class EsperSubscriber {
30     EPServiceProvider cep;
31     EPRuntime cepRT;
32     EPAdministrator cepAdm;
33     MessagePack msgpack;
34     Template<Map<String, String>> mapTmpl;
35     Template<List<Value>> listTmpl;
36     ZMQ.Context context;
37     ZMQ.Socket subscriber;
38     EPLServer svr;
39
40     public EsperSubscriber(String[] filters, Properties conf){
41
42         cep = EPServiceProviderManager.getDefaultProvider();
43         cepRT = cep.getEPRuntime();
44         cepAdm  = cep.getEPAdministrator();
45
46
47         msgpack = new MessagePack();
48         mapTmpl = tMap(TString, TString);
49         listTmpl = tList(TValue);
50
51         context = ZMQ.context(1);
52         subscriber = context.socket(ZMQ.SUB);
53         subscriber.connect("tcp://localhost:5556");
54         for(String filter : filters){
55             subscriber.subscribe(filter.getBytes());
56         }
57
58         svr = new EPLServer(cepAdm,conf);
59
60     }
61
62     public void start() throws IOException{
63         byte[] msgByte;
64
65         svr.start();
66         BufferUnpacker unpacker = msgpack.createBufferUnpacker();
67         while(true){
68             if ((msgByte = subscriber.recv(ZMQ.NOBLOCK)) != null) {
69                 unpacker.feed(getMessage(msgByte));
70                 List<Value> valueList = unpacker.read(listTmpl);
71                 String tag = valueList.get(0).asRawValue().getString();
72                 String eventName = tag.replace(".","_");
73                 Value record = valueList.get(2);
74                 if(record.isMapValue()){
75                     Map<String, String> m = new Converter(record).read(mapTmpl);
76                     m.put("tag",tag);
77                     try{
78                         cepRT.sendEvent(m,eventName);
79                     } catch (EPException e){
80                         createSchemaFromMap(m, eventName, cepAdm);
81                     }
82                 }
83             }
84             else {
85                 try {
86                     Thread.sleep(100);
87                 }catch(Exception e){
88                     e.printStackTrace();
89                 }
90             }
91         }
92
93     }
94
95
96     public byte[] getMessage(byte[] received){
97         int i = 0;
98         while (received[i] != 0x20){
99             i++;
100        }
101        int msglen = received.length - i - 1;
102        byte[] ret = new byte[msglen];
103        System.arraycopy(received, i+1, ret, 0, msglen);
104        return ret;
105    }
106
107    public void createSchemaFromMap(Map m, String event, EPAdministrator cepAdm){
108        Map<String, Object> typeMap = new HashMap<String, Object>();
109
110        for(String key: (Set<String>)m.keySet()){
111            typeMap.put(key, String.class);
112        }
113        cepAdm.getConfiguration().addEventType(event, typeMap);
114        System.out.println("New event defined: " + event);
115    }
116
117    public static void main(String[] args) throws IOException{
118    }
119 }
```