CptS 464/564 Project #2
Distributed System "Top" Functionality

Given: Monday, October 3, 2010
Due: 5:00 PM Friday, October 21, 2011
Weight: 10% of Final Grade

Overview

In this project, you will create a distributed system "Top" functionality using RTI (Real-Time Innovations) DDS (Data Distribution Service).

The publisher will send CPU and memory usage data periodically, and the subscriber will use **ContentFilteredTopic** to get a subset of the messages. Also, in this project you will use some **QosPolicies** to control the behavior of your program.

Problem setting

`TopFunctionPublisher`

- It publishes data of type `TopFunction` periodically every 4 seconds. Please use "CS464/564_Project_2_ username" as topic name, substituting your EECS user name for "username".

- It uses **Ownership and OwnershipStrength QoS Policies** to allow for multiple publishers with different priorities. Only the message sent from the publisher with highest priority will be delivered to subscriber.

`TopFunctionSubscriber`

- It checks for and displays the message received from the publisher. It prints out a message of its own every 4 seconds. Again, please use "CS464/564_Project_2_ username" as topic name.

- It uses **ContentFilteredTopic** to subscribe the interested messages. For example, only the messages in which CPU usage is higher than 2% and memory usage is higher than 50% will be sent to the subscriber.

- It uses **Ownership QosPolicy** so that only the message sent from the publisher with highest priority will be delivered to subscriber.

The data is defined in top.idl:

```
// top.idl
struct TopFunction {
string username;
string hostname;
string currentTime;
```

```
float cpuUsage;
float memUsage;
long procNumber;
};
```

Expected output:

```
//Publisher:
Writing TopFunction, #0
Writing TopFunction, #1
Writing TopFunction, #2
……

//Subscriber:
TopFunction subscriber sleeping for 4 sec...
Received:
user: dalvarez
hostname: localhost.localdomain
currentTime: 2011/10/06 00:59:57
cpuUsage: 2.479339%
memUsage: 93.13231%
processes: 25

TopFunction subscriber sleeping for 4 sec...
Received:
username: dalvarez
hostname: localhost.localdomain
currentTime: 2011/10/06 01:00:03
cpuUsage: 7.6271186%
memUsage: 93.144295%
processes: 24
……
```

Additional 564 Work

Students enrolled in CptS 564 should **choose one more QosPolicy** to control your program's behavior. You will get 10% extra credits for a right implementation.

Implementation language

You can use either Java or C++/C.

Implementation steps

1. Create a file named top.idl which has been specified above.
2. Run `rtiddsgen` to generate the program files.
3. Write a class called `TopUtil` used to get current CPU and memory usage, and current number of processes.
4. Add `TopUtil` class source file to the makefile.
5. Modify the generated source code files according to the problem setting specified above.
6. Build and run your programs.

Note: You can run any number of publishers and subscribers programs, and can add and remove them dynamically from the domain.

Turn-ins

The project is due at 5:00 PM on Friday October 21. A zip file containing all your files must be submitted via Angel LMS to the corresponding lesson assignment in the course page. No hardcopy of the files is required. The files to submit are:

1. **All the files you created or modified by hand**. Note, this does not include files that are generated by the IDL compiler and not modified by you.

2. **Text files with sample output of your programs running**. (That is, the publisher sends messages and the subscriber receives and displays messages). You need to copy the output from the screen (or send it to a file) and print it out.

3. **Screenshot(s) of your programs running**.

Please do not modify the last accessed dates on your created or modified files after the submission date, in case we need to check the files.

Appendix-I

Everything you need to calculate CPU usage, memory usage and number of running processes is in the pseudo-filesystem **/proc**: CPU usage from /proc/stat, and /proc/meminfo for memory usage. Also, the existing processes ids appear in /proc as directories. Please figure out how to extract all the information first, before starting the actual programming.

Appendix-II

You can read the online documents at http://community.rti.com/ or the documents under $NDDSHOME/doc. The most important documents for this project would be RTI_DDS_GettingStartedGuide, RTI_DDS_UsersManual, and RTI DDS API documentation.

For ContentFilteredTopic, read:
- User's Manual: 5.4
- Getting Started Guide: 5.2.1

For QosPolicies, read:
- User's Manual: 4.1.7, 4.2, 6.5.13, 6.5.14
- Getting Started Guide: 4.2