OpenRTM-aist (Java) - 機能 #3225

同一プロセスのコンポーネント間のデータポート接続で、データ転送をダイレクトに行うようにする

2015/06/17 00:42 - n-ando

ステータス:	担当	**開始日:**	2015/06/17
優先度:	通常	**期日:**	
担当者:	win-ei	**進捗率:**	100%
カテゴリ:		**予定工数:**	0.00時間
対象バージョン:	RELEASE_1_2_0		

説明

同一プロセスに存在する複数のコンポーネント間のデータ転送においては、

- マーシャリング
- バッファリング
- アンマーシャリング

が行われているが、実際には単にOutPortからInPortの変数にデータを書き込むことができれば効率的である。
複数のRTCが並列に動作していたり、異なる周期で動作する場合などはバッファリングも必要になるケースも考えられるが、例えばECを共有しそれらのコンポーネントの実行がシーケンシャルであることが保証される場合、コンポーネントの実行・データ転送効率を上げる意味でも、OutPort→InPortで直接変数に書き込む機能があればよいのでこれを実装する。

関係しているリビジョン

リビジョン 757 - 2016/02/15 10:11 - win-ei

Direct data put functionality between data ports has been implemented. refs #3225

リビジョン 801 - 2016/03/21 19:44 - win-ei

Direct data put functionality between data ports has been implemented. refs #3225

リビジョン 803 - 2016/03/22 16:45 - win-ei

Added get_configuration function. refs #3225

履歴

#1 - 2015/06/17 22:12 - n-ando

仕様

- ConnectorProfile.properties["dataport.outport.direct_dataput.disable"] != YES の時、または当該プロパティーが設定されていない場合にDirect data put モードにする
- Direct data-putモードでは、同一プロセス内のInPortとOutPortのpush接続時に、corba_cdrのインターフェースを通すことなく、InPortの変数にOutPortから直接データを書き込む
- Direct data putモードでは、データが変数に書き込まれた場合にInPort::isNew()=true, InPort::isEmpty()=falseとなり、InPort::read()がよばれた後は、変数およびバッファの状態によりInPort::isNew(),isEmpty()の状態が変化する

実装方法

- InPort
  - 変数書き込みフラグとミューテックスを追加
  - void write(const DataType&) 関数（OutPortConnectorからコール）を用意し、変数書き込み、フラグtrue代入をロック下で行う
  - 変数書き込みフラグの状態に応じてisNew(),isEmpty()を修正
  - read()関数内で変数書き込みフラグがtrueの場合のロジックを追加
  - InPort変数への参照をミューテックスで保護

- OutPortConnector
  - InPortのサーバントを与える setInPort() 関数を追加
  - write()関数内にInPortのwrite()関数を呼ぶ static_cast<InPort<DataType>*>(m_directInPort)->write(data); コードを追加

- OutPortBase
  - コネクタープロファイルのプロパティー (dataport.outport.)direct_dataput.disable を取得。disable=YESなら通常の処理、でなければ以下の処理を行う
  - Push接続時のコネクタ生成ロジック内で、InPortのローカルサーバントを取得する
  - 取得できない場合は、リモートオブジェクトなのでそのまま継続、取得できる場合は、コネクタにInPortのサーバントを渡す

(setLocalInPort())
  - サーバント取得ロジック： InPortBase* inport = dynamic_cast<InPortBase*>(poa->reference_to_servant(obj));

#2 - 2016/02/15 10:19 - win-ei

- 進捗率 を 0 から 100 に変更

動作確認

- ConnectorComp.java に以下を追加。

```
CORBA_SeqUtil.push_back(nvholder,
NVUtil.newNVString("dataport.outport.direct_dataput.disable",  "YES"));
```

- rtc.conf で ConsoleOut をpreload、precreate、preactivateする。
  具体的には以下をrtc.confへ追加する。

```
manager.modules.load_path:./RTMExamples/SimpleIO
manager.modules.preload:  ConsoleOut
manager.components.precreate:  ConsoleOut

manager.components.preactivate:  ConsoleIn0,  ConsoleOut0
```

- ConsoleIn.sh を起動後、Connector.sh を起動して動作確認

#3 - 2016/02/15 10:26 - win-ei

- 進捗率 を 100 から 50 に変更

#4 - 2017/09/06 15:23 - katami

動作確認

- rtc.conf で ConsoleOut をpreload、precreate、preactivateする。
  具体的には以下をrtc.confへ追加する。

```
manager.modules.load_path:./RTMExamples/SimpleIO
manager.modules.preload:  ConsoleOut
manager.components.precreate:  ConsoleOut

manager.components.preactivate:  ConsoleIn0,  ConsoleOut0
```

- [direct][push][new]

```
$  rtcon  ./localhost/ubuntu.host_cxt/ConsoleIn0.rtc:out  ./localhost/ubuntu.host_cxt/ConsoleOut0.rtc:in  -p  dataport.interface_type=direct
```

- [direct][push][flush]

```
$  rtcon  ./localhost/ubuntu.host_cxt/ConsoleIn0.rtc:out  ./localhost/ubuntu.host_cxt/ConsoleOut0.rtc:in  -p  dataport.interface_type=direct  -p  d
ataport.subscription_type=flush
```

- [direct][push][periodic]

```
$  rtcon  ./localhost/ubuntu.host_cxt/ConsoleIn0.rtc:out  ./localhost/ubuntu.host_cxt/ConsoleOut0.rtc:in  -p  dataport.interface_type=direct  -p  d
ataport.subscription_type=flush
```

- [direct][pull]

```
$  rtcon  ./localhost/ubuntu.host_cxt/ConsoleIn0.rtc:out  ./localhost/ubuntu.host_cxt/ConsoleOut0.rtc:in  -p  dataport.interface_type=direct  -p  d
ataport.dataflow_type=pull
```

#5 - 2017/09/06 15:24 - katami

- ステータス を 新規 から 担当 に変更

- 担当者 を win-ei にセット

- 進捗率 を 50 から 100 に変更

#6 - 2017/09/06 15:24 - katami

- 対象バージョン を RELEASE_1_2_0 にセット