Grid RPC data management integration into the SAGA RPC package

Yves Caniou, Eddy Caron, Frédéric Desprez, Gaël Le Mahec

September 2, 2009

This document is a draft for a proposition of the Grid RPC data management API integration into the SAGA API. Many points have to be discussed and modified to be conform to the SAGA Look-&-Feel without breaking the backward compatibility with the current SAGA Grid RPC specifications. The parts in blue are extracted from the SAGA core API v1.0 document, the parts in red are new.

1 Simple integration

Here, we just added the new `data_type`, `completion_mode` and `data_mode` enumeration types and the three base classes for the RPC data management: `data_parameter`, `memory_mapper` and `data_container`. The RPC calls are performed using the same `call` method with or without data management extension. Indeed, the `data_parameter` class is a direct subclass of the `parameter` class. However, because these classes don't share any field or method, **they probably should be distinct**.

```
package saga.rpc
{
  enum io_mode
  {
    In = 1, // input parameter
    Out = 2, // output parameter
    InOut = 3 // input and output parameter
  }

  enum data_type
  {
    Byte = 1,    // Byte
    Int = 2,     // integer type
    Float = 3,
    Double = 4,  // double type
    Complex = 5, // complex type
    String = 6,  // character string
    File = 7     // file
  }

  enum completion_mode
  {
    Wait_all = 1, // wait for all the transfers
    Wait_any = 2 // wait for one transfer among a set
  }
```

```
enum data_mode
{
  Volatile = 1, // volatile data, may be erased after use
  Unique_Volatile = 2, // volatile data, only one replica
  Persistent = 3, // persistent data
  Sticky = 4, // sticky data (pers. data that shouldn't be erased)
  Unique_Sticky = 5 // sticky data, only one replica
}

class parameter : extends saga::buffer
// from buffer saga::object
// from object saga::error_handler
{
  CONSTRUCTOR (in array<byte> data = "",
               in int size = 0,
               in io_mode mode = In,
               out buffer obj);
  set_io_mode (in io_mode mode);
  get_io_mode (out io_mode mode);
}
```

```
class data_parameter : extends saga::rpc::parameter
// from parameter saga::rpc for backward compatibility
// Should be discussed...
{
  CONSTRUCTOR(in array<int> dim_sizes,
              in array<uri> input_uris,
              in array<uri> output_uris,
              in data_type type,
              in array<data_mode> data_modes
              out parameter obj);

  DESTRUCTOR(in data_parameter obj);

  // to perform data transfers
  // (see the GridRPC proposal documentation
  transfer(in array<uri> additionnal_input_uris,
           in array<uri> additionnal_output_uris,
           in array<data_mode> additionnal_input_modes,
           in array<data_mode> additionnal_output_modes);

  // to wait the end of a transfer
  wait(in completion_mode mode);

  // to unbind the data from this parameter object
  unbind();

  // to free the data on the passed locations
  free(in array<uri> locations);

  // to de-/serialize this data
  serialize(in uri output);
  deserialize(out uri input);

  // fiels accessors
  set_dim_sizes (in array<int> dim_sizes);
  set_input_uris (in array<uri> input_uris);
  set_output_uris (in array<uri> output_uris);
  set_type (in data_type type);
  set_data_modes (in array<data_mode> data_modes);

  get_dim_sizes (out array<int> dim_sizes);
  get_input_uris (out array<uri> input_uris);
  get_output_uris (out array<uri> output_uris);
  get_type (out data_type type);
  get_data_modes (out array<data_mode> data_modes);
}

class memory_mapper : implements saga::object
implements saga::permissions
{
  CONSTRUCTOR();

  // the binary data is associated to the string key
  data_memory_map(in string key,
                  in array<byte> data);

  data_memory_unmap(in string key,
                    out array<byte> data);
}
```

```
class data_container : extends data_parameter
// from data_parameter saga::rpc
{
  CONSTRUCTOR(in array<int> dim_sizes,
              in array<uri> input_uris,
              in array<uri> output_uris,
              in array<data_mode> data_modes);
  CONSTRUCTOR(in data_parameter data);

  // add a data into this data container
  set_data(in data_parameter data, in int rank);
  // get a data from this data container
  get_data(out data_parameter data, in int rank);

  // add a data at the end of this data container
  push_back(in data_parameter data);
  // add a data at the beginning of this data container
  push_front(in data_parameter data);
  // get a data from the end of this container and remove it
  pop_back(out data_parameter data);
  // get a data from the beginning of this container and remove it
  pop_front(out data_parameter data);
}

class rpc : implements saga::object
implements saga::async
implements saga::permissions
// from object saga::error_handler
{
  CONSTRUCTOR (in session s,
               in saga::url url = "",
               out rpc obj );
  DESTRUCTOR (in rpc obj );
  // rpc method invocation
  call (inout array<parameter> parameters );
  // handle management
  close (in float timeout = 0.0);
}
}
```

All the notes of the saga::buffer are not applying. Must be corrected to conform the SAGA Look-&-Feel

1.1 Class data_parameter

For backward compatibility, the `data_parameter` class inherits the saga::rpc::parameter class.

The implementation of the `call` method then acts differently depending of the instance type (`saga::rpc::parameter` or `saga::rpc::data_parameter`). This class introduces different attributes and their respectives accessors (see Grid RPC data management proposal for more information about the attributes). It also introduces the `transfer`, `wait`, `unbind`, `free`, `serialize` and `deserialize` methods.

```
- CONSTRUCTOR
  Purpose: initializes a remote function handle
  Format: CONSTRUCTOR (in array<int> dim_sizes,
                       in array<uri> input_uris,
                       in array<uri> output_uris,
                       in data_type type,
                       in array<data_mode> data_modes
```

```
                        out parameter obj);
  Inputs: dim_sizes: the size of each dimension of the data
          input_uris: the URIs where the data can be found
          output_uris: the URIs where the data have to be copied after a call
          type: the data type
          data_modes: the data persistence modes

  InOuts: -
  Outputs: obj the newly created object
  PreCond: -
  PostCond: -
  Perms: -
  Throws: NotImplemented
          IncorrectURL
          BadParameter
          NoSuccess
  Notes: - notes of the saga::rpc::parameter

- DESTRUCTOR
  Purpose: destroy a parameter_data instance
  Format: DESTRUCTOR (in data_parameter obj)
  Inputs: obj: the object to destroy
  InOuts: -
  Outputs: -
  PreCond: -
  PostCond: -
  Perms: -
  Throws: -
  Notes: - notes of the parameter destructor
- transfer
  Purpose: proceed to the data transfer
  Format: transfer (in array<uri> additionnal_input_uris,
                    in array<uri> additionnal_output_uris,
                    in array<data_mode> additionnal_input_modes,
                    in array<data_mode> additionnal_output_modes);
  Inputs: additionnal_input_uris: A list of additionnal data source URIs
          additionnal_output_uris: A list of additionnal data destination URIs
          additionnal_input_modes: the corresponding persistence modes
          additionnal_output_modes: the corresponding persistence modes
  InOuts: -
  Outputs: -
  PreCond: -
  PostCond: -
  Perms: -
  Throws: NotImplemented
          IncorrectURL
          BadParameter
          DoesNotExist
          PermissionDenied
```

```
          AuthorizationFailed
          AuthenticationFailed
          NoSuccess
  Notes: -

- wait
  Purpose: wait for the end of the transfer(s)
  Format: wait (in completion_mode mode);
  Inputs: completion_mode: to wait for all the transfers end or for one transfer end
  InOuts: -
  Outputs: -
  PreCond: a transfer was initiated on this data_parameter object
  PostCond: at least one transfer ended
  Perms: -
  Throws: NotImplemented
          IncorrectURL
          BadParameter
          DoesNotExist
          PermissionDenied
          AuthorizationFailed
          AuthenticationFailed
          NoSuccess
  Notes: -

- unbind
  Purpose: unbind the Grid RPC data without freeing it on the remote servers
  Format: unbind ();
  Inputs: -
  InOuts: -
  Outputs: -
  PreCond: -
  PostCond: -
  Perms: -
  Throws: NotImplemented
          NoSuccess
  Notes: -

- free
  Purpose:  frees the data on a subset or on all the different locations where
            the data is stored, and unbind the handle and the data. This function
            may be used to explicitly erase the data on a storage resource.
  Format: free (in array<uri> locations);
  Inputs: locations: an list of location where to erase the data. If empty, the
          data is erased on all the locations where it is stored
  InOuts: -
  Outputs: -
  PreCond: -
  PostCond: the object does not reference the data
  Perms: -
```

```
Throws: NotImplemented
        IncorrectURL
        BadParameter
        DoesNotExist
        PermissionDenied
        AuthorizationFailed
        AuthenticationFailed
        NoSuccess
Notes: -
```

TODO: Method specification details

To be continued with the (de)serialize methods and attributes accessors.

1.2 Class data_container

This class extends the `data_parameter` class introducing variable length list of data. (see the Grid RPC data management API proposal for more details)

1.3 Class memory_mapper

This class is used to manage data stored on local memory It provides an uniform manner to access the data with URI. See more details in the Grid RPC data management API proposal.

2 Two different RPC classes

We define a `rpc_data` class which is a subclass of the `rpc` class and which adds a new `call` method using an array of `data_parameter`. The `data_parameter` class is no more a subclass of the `parameter` class.

```
package saga.rpc
{
  ...
  class rpc : implements saga::object
  implements saga::async
  implements saga::permissions
  // from object saga::error_handler
  {
    CONSTRUCTOR (in session s,
                 in saga::url url = "",
                 out rpc obj );
    DESTRUCTOR (in rpc obj );
    // rpc method invocation
    call (inout array<parameter> parameters );
    // handle management
    close (in float timeout = 0.0);
  }
```

```
  class rpc_data : extends saga::rpc::rpc
  implements saga::async
  implements saga::permissions
  // from object saga::error_handler
  {
    CONSTRUCTOR (in session s,
                 in saga::url url = "",
                 out rpc obj );
    DESTRUCTOR (in rpc obj );
    // the new call method
    call (inout array<data_parameter> parameters );
  }
}
```

2.1 Class rpc_data

This class extends the `saga::rpc::rpc` class introducing the data management extensions proposed in the Grid RPC data management API proposal.

```
- CONSTRUCTOR
  Purpose:  initialize a remote function handle
  Format: CONSTRUCTOR (in session s,
                       in saga::url url = "",
                       out rpc obj );
  Inputs: s: saga session to use
          url: the methods URL
  InOuts: -
  Outputs: obj: the newly created object
  PreCond: -
  PostCond: the instance is open
  Perms: Query
  Throws: NotImplemented
          IncorrectURL
          BadParameter
          DoesNotExist
          PermissionDenied
          AuthorizationFailed
          AuthenticationFailed
          NoSuccess
  Notes: all the notes of the saga::rpc::rpc CONSTRUCTOR

- DESTRUCTOR
  Purpose:  destroy the object
  Format: DESTRUCTOR (in rpc obj);
  Inputs: obj: the object to destroy
  InOuts: -
  Outputs: -
  PreCond: -
  PostCond: the instance is closed
  Perms: -
  Throws: -
```

```
  Notes: all the notes applying to saga::rpc::DESTRUCTOR

- call
  Purpose: call the remote procedure
  Format:  call (inout array<data_parameter> parameters );
  Inputs: -
  InOuts: parameters: argument/results for call
                      the type used for the data is data_parameter
  Outputs: -
  PreCond: the instance is open
  PostCond: the instance is available for another call
  Perms: Exec
  Throws: NotImplemented
          IncorrectURL
          BadParameter
          DoesNotExist
          IncorrectState
          PermissionDenied
          AuthorizationFailed
          AuthenticationFailed
          Timeout
          NoSuccess
```