Table of Contents

xrootd in SRM and GLUE: a proposal

Introduction

This document has the following purposes:

- to describe how the different SRM implementations manage the xrootd protocol;
- to describe how the different storage systems publish xrootd to the information system;
- to highlight some problems of the current situation;
- to propose how to solve these problems.

DISCLAIMER: this proposal has not yet been publicly discussed in a GDB meeting as therefore it has not been officially blessed in any way.

How to get an xrootd TURL in an *srmPrepareToGet* and an *srmPrepareToPut* function

One of the arguments of the functions is a *TTransferParameters* structure, which contains an *arrayOfTransferProtocols* string array. Each protocol is therefore identified by a string (*protocol identifier*). The SRM will return to the client a TURL for the first supported protocol in the array. The following tables show the type of TURL returned by different SRM implementations for different protocol identifiers; entries in red are proposed for deprecation, entries in green are proposed for implementation and entries in black should not change.

CASTOR

Protocol identifier	TURL syntax	Actual TURL protocol	Understood by ROOT
root	`castor://...`	rootd protocol	yes
xroot	`root://...`	xrootd protocol	yes

Support for the rootd protocol is a legacy feature and it is expected to be phased out some time in the future. As a consequence, `root` as protocol identifier will become unsupported.

Problem: the `root` identifier corresponds to a different access protocol in CASTOR and dCache.

Proposal: nothing should change in CASTOR.

dCache

Protocol identifier	TURL syntax	Actual TURL protocol	Understood by ROOT
root (1)	`root://...`	xrootd protocol	yes
xroot	`root://...`	xrootd protocol	yes

(1) This value will not be deprecated until it is guaranteed that no clients critically depend on it.

Problem: the `root` identifier corresponds to a different access protocol in CASTOR and dCache.

Proposal: to use `xroot` rather than `root` as protocol identifier, as CASTOR and DPM do.

DPM, BeStMan

Protocol identifier	TURL syntax	Actual TURL protocol	Understood by ROOT
xroot	`xroot://...`	xrootd protocol	not by default

Problem: the TURL syntax is not compatible by default with ROOT.

Proposal: either return TURLs of the form `root://` or make ROOT understand TURLs of the form `xroot://` (see next section).

TURL prefix

The TURL prefix can either be `root` or `xroot`. Arguments can be given in favour on one choice over the other, for example:

- `root`: most used for historical reasons; supported by default by ROOT; coherent with the current CASTOR and dCache behaviour;
- `xroot`: equal to the protocol identifier; coherent with the current DPM behaviour.

Even if ROOT does not support by default the `xroot://` syntax, it can be easily customized to do so.

Proposal: An SRM server MUST return TURLs that start by `root://` or `xroot://` for transfers that are to use the xrootd protocol. An SRM server MAY return TURLs starting by `root://` for xrootd protocol transfers if it provides a service for a community whose software require xrootd TURLs of this form (e.g. the LHC experiments). An SRM server that supports no such community SHOULD return TURLs starting `xroot://` for xrootd protocol transfers.

How to publish support for the xrootd protocol in the information system

GLUE 1.3 and the "WLCG Installed capacity" document

According to the GLUE 1.3 schema specification and the "WLCG Installed Capacity" document version 1.9, all Storage Elements must adhere to these prescriptions:

- **Access protocol**. If xrootd is a supported access protocol, it MUST be published and the *Type* should be listed here;
- **Control protocol**. If an SE provides an "xroot door" it MUST publish a control protocol object for xrootd and MUST NOT do so otherwise. The *Type* MUST be `xroot` and the full control endpoint is of the form `[x]root://...`.

Although it is not written anywere and is not compulsory by any means, it is desirable to use the same string for the access protocol type in GLUE and the protocol identifier in SRM. There are at least two arguments in favour:

- it avoids the need to know which protocol identifier corresponds to which access protocol *Type*;
- some SAM tests assume they are equal.

GLUE 2.0

In GLUE 2.0, the `StorageAccessProtocol` entity has a *Type* attribute, whose value for the xrootd protocol is `xrootd`, but it could be easily changed to `xroot`. it is not mandatory to publish a given access protocol to make it discoverable. The concept of control protocol is not present and it is basically replaced by the `StorageEndpoint`.

There are not yet any WLCG-specific recommendations on how to use the GLUE 2.0 schema.

CASTOR

Access protocol type	Actual access protocol
xroot	xrootd protocol

Problem: CASTOR does not publish anything at all. An official CASTOR information provider is currently in development.

Proposal: to publish `xroot` as *Type*.

dCache

Access protocol type	Actual access protocol
root (1)	xrootd protocol
xroot	xrootd protocol

(1) This value will not be deprecated until it is guaranteed that no clients critically depend on it.

Problem: the current value, `root`, is not acceptable for xrootd.

Proposal: to publish `xroot` as *Type*.

DPM

Access protocol type	Actual access protocol
xroot	xrootd protocol

Problem: none.

`xroot` or `xrootd`?

A point that was discussed was whether to use `xroot` or `xrootd` as protocol identifier. Even if in the GLUE 2.0 working group, `xrootd` was agreed as the official access protocol name, this can still be easily changed. Hence the

Proposal: to use everywhere `xroot` and amend the GLUE 2.0 specifications.

Conclusions and proposal

To summarize, the following proposal is made:

- `xroot` MUST be a valid protocol identifier in SRM for the xrootd protocol. This requires a change only in dCache;
- `root` SHOULD NOT be used as protocol identifier in SRM for the xrootd protocol. This would require a change only in dCache: however, it is agreed that it can wait until it has been verified that it will not break existing software;
- TURLs for the xrtood protocol MUST begin by either `root://` or `xroot://`. They MAY be `root://` if the SRM implementation is used by a community requiring TURLs of this form but SHOULD be `xroot://` otherwise. No change is required for WLCG;
- ROOT SHOULD be able to understand the `xroot://` TURL syntax by default; it is understood that even if it does not it is very easy to configure it to that purpose;
- `xroot` MUST be used as protocol *Type* in the information system for the xrootd protocol. This requires changes in the information providers for CASTOR and dCache and an amendment of the GLUE 2.0 schema specifications. dCache MAY wait until `xroot` is supported as protocol identifier in SRM.

-- AndreaSciaba - 11-Mar-2010

This topic: LCG > SrmXroot
Topic revision: r7 - 2010-05-12 - AndreaSciaba