Infrastructure - Task #4214

Story # 2829 (Closed): Structural changes to D1 Data Types schema for Version 2.0 release

PreferredMemberNode List should indicate ordinality

2014-01-07 09:20 - Robert Waltz

Status:	Rejected	**Start date:**	2014-01-07
Priority:	Normal	**Due date:**	
Assignee:	Ben Leinfelder	**% Done:**	0%
Category:	d1_schemas	**Estimated time:**	0.00 hour
Target version:	CCI-2.0.0		
Milestone:	CCI-2.0	**Story Points:**	
Product Version:	*		

Description

The PreferredMemberNode List is now considered to be a well ordered set. The order of the preferred nodes matters when determining the most appropriate node to replicate an object to.

We should provide an attribute or other property such that the order of the set may be traversed irrespective of container type.

History

#1 - 2014-01-07 09:21 - Robert Waltz

- Assignee changed from Matthew Jones to Robert Waltz

#2 - 2014-07-09 21:43 - Ben Leinfelder

- Status changed from New to Rejected

- translation missing: en.field_remaining_hours set to 0.0

This is not a schema change, but rather an implementation detail. Our structures - both in XML and in Java indicate an ordered list and should be preserved in the order we receive. Whether or not this has any meaning is still debatable, but we should at least respect the given order. Opening a ticket for Metacat to store this with order preserved in the backing table.

#3 - 2014-07-09 21:46 - Ben Leinfelder

- Estimated time set to 0.00

See replacement bug: https://projects.ecoinformatics.org/ecoinfo/issues/6576

#4 - 2014-07-09 23:02 - Robert Waltz

Wednesday July 9th, 2014
(16:55:22) benMac: hi robert - looking at https://redmine.dataone.org/issues/4214
(16:55:27) robert: yes?
(16:55:41) benMac: wondering how crucial ordering of pref nodes is
(16:57:26) benMac: skye, since you worked a lot on replication, was that a much-needed feature?
(16:57:52) robert: Don't remember how crucial it is. I think that would be a Chris J question. It came up when we were writing Tidy and noticed that we couldn't determine the exact order of the pref nodes in case there was a conflict that we needed to merge.
(16:59:14) benMac: yeah, there is no order guaranteed
(16:59:31) robert: right. But the order is important
(16:59:39) benMac: but I don't think it is
(16:59:49) benMac: it's a suggestion
(17:00:18) benMac: there's not even a guarantee that a replica will end up on any of those nodes
(17:00:38) robert: sure, but if we can send it to a node, then we should try to honor their preferences
(17:00:58) robert: but we are unable to determine what their 'most' preferred node is
(17:01:51) benMac: I don't think our current type really has a "most"
(17:02:39) benMac: it might happen to be ordered correctly, but there's nothing in the storage that would guarantee that
(17:02:41) robert: at any rate, I'm not the only one who worked on this. Rob and Chris were also on the discussion, so It is an issue that needs more than my input

(17:02:48) benMac: sure
(17:02:54) benMac: had to start somewhere
(17:03:16) robert: right. from my perspective, i think what I've stated above summarizes my understanding
(17:03:17) benMac: figured we could start the discussion in a chat room!
(17:03:29) benMac: and that was helpful
(17:04:20) robert: it mostly stemmed from conflicts in the systemMetadata and how to determine the correct manner to merge differences
(17:09:08) benMac: okay, so robert, I think the ordering issue is that we were not implementing it correctly
(17:09:36) robert: oh?
(17:09:43) benMac: since in the XML representation, the NodeRefereces are ordered in the order they appear in the document
(17:10:03) benMac: and jibx makes them List, which is ordered
(17:10:31) benMac: I just know that metacat shoves them all into a table with no ordering index
(17:10:48) benMac: and pulls them out with no order by (since we don't know it)
(17:10:52) benMac: so...
(17:11:43) benMac: I don't think we necessarily need urn:node:myNode as much as we need to ensure the serializations remain consistent
(17:20:39) robert: ReplicationPolicy is a complexType that contains a sequence of preferredMemberNode and then blockedMemberNode
(17:21:45) robert: though any item of preferredMemberNode must come before any item of blockedMemberNode, there is nothing to suggest that the first preferredMemberNode in the set of preferredMemberNodes is absolutely set in that position
(17:22:05) robert: i think we have the right implementation
(17:22:19) robert: it's the design that needs to be addressed.
(17:22:49) robert: So, my interpretation of the xsd may be incorrect
(17:28:13) robert: but, once again, lets get others in the discussion
(17:35:32) robert: maybe our documentation should just note that the set is unordered .

#5 - 2014-07-30 02:59 - Robert Waltz

- Assignee changed from Robert Waltz to Ben Leinfelder

#6 - 2014-10-01 16:57 - Robert Waltz

- Target version set to CCI-2.0.0