ResourceSync
Towards a Web-Based Approach for Resource Synchronization

Herbert Van de Sompel
Los Alamos National Laboratory
@hvdsomp

ResourceSync is funded by
The Sloan Foundation
ResourceSync

Problem Perspective

Technical Directions

An Experiment

Q&A
ResourceSync Core Team – NISO & OAI

Cornell University & OAI:
    Berhard Haslhofer, Carl Lagoze, Simeon Warner

Old Dominion University & OAI:
    Michael L. Nelson

Los Alamos National Laboratory & OAI:
    Martin Klein, Robert Sanderson, Herbert Van de Sompel

NISO:
    Todd Carpenter, Nettie Lagace, Peter Murray
ResourceSync Technical Group

- Manuel Bernhardt, Delving B.V.
- Richard Jones, JISC
- Graham Klyne, JISC
- Stuart Lewis, JISC
- Kevin Ford, Library of Congress
- David Rosenthal, LOCKSS
- Christian Sadilek, Red Hat
- Shlomo Sanders, Ex Libris, Inc.
- Sjoerd Siebinga, Delving B.V.
- Jeff Young, OCLC Online Computer Library Center
ResourceSync

Problem Perspective

Technical Directions

An Experiment

Q&A
ResourceSync Problem

• Consideration:
  • **Source** (server) A has resources that change over time: they get created, modified, deleted, moved, …
  • **Destination** (servers) X, Y, and Z leverage (some) resources of Source A.

• Problem:
  • Destinations want to keep in step with the resource changes at Source A: resource synchronization.

• Task of ResourceSync effort:
  • Design an approach for resource synchronization aligned with the Web Architecture that has a fair chance of adoption by different communities.
    • The approach must scale better than recurrent HTTP HEAD/GET on resources.
ResourceSync Use Cases

• arXiv mirroring.
• Metadata and content transfer from Europeana partners to central Europeana hub.
• Synchronization, local caching of Linked Data content.
• Recurrently collecting Memento metadata from IIPC web archives to central aggregator.
• Keeping up-to-date with resources that reside on a Web server.

• Use cases have different requirements regarding synchronization accuracy:
  • Synchronization coverage: perfect …… good enough
  • Synchronization speed: fast ……… fast enough
Web Context

- The context in which resource synchronization will take place is the Web.
- Resources considered for synchronization:
  - Are identified by dereference-able URIs.
  - Are cacheable.

=> Caveat: It is understood that resources that require client-side processing (e.g. hashbang URIs) cause problems.
3 Synchronization Needs

• Overall, there are 3 distinct needs regarding resource synchronization. All 3 are considered in scope for the effort:

1. **Baseline matching**: An approach to allow a Destination that wants to start synchronizing with a Source to perform an initial catch up – **Dump**.

2. **Incremental resource synchronization**: An approach to allow a Destination to remain up-to-date regarding changes at the Source.

3. **Audit**: An approach to allow checking whether a Destination is in sync with a Source – **Inventory**.
Change Notification – Content Transfer

• Distinguish between two aspects related to incremental resource synchronization:

  2.a. Change notification: An approach to allow a Destination to understand that a Source’s resource has changed; and what the nature of the change is.

  2.b. Content transfer: An approach to allow a Destination to update its holdings to reflect the change the resource underwent at the Source.

=> For both, PUSH or PULL approaches are conceivable.
Selective Synchronization

• There is a need to be able to synchronize with a limited set of a Source’s resources: selective synchronization.

• This leads to the notion of channels for resource synchronization.
• The need for channels is most apparent for resource synchronization but may have to be carried through in baseline matching and audit.

=> This raises an issue:
• The definition of channels by the Source is straightforward, yet may not meet the Destination’s needs.
• The definition of channels by a Destination is less straightforward.
Memory

• For robustness, both change notification and content transfer may need a memory function to allow for synchronization after a Destination has missed updates:
  
  • Catching up on change notifications - Digests.
  • Catching up on resource versions - Mementos.

⇒ However, it would be beneficial if resource synchronization could operate in a memory-less mode in cases where less than perfect sync is acceptable.

⇒ An important architectural consideration is where that memory should reside: Source, intermediary.
Changes Only – Entire Resource

• Two approaches for content transfer:

  2.b.1. Synchronization is achieved by transfer of the entire resource.
  2.b.2. Synchronization is achieved by transfer of resource changes only.

⇒ Focus is on (2.b.1) because of complexities re content-type-specific resource segment description and re-assembly instructions involved in (2.b.2).
⇒ Focus re (2.b.1) is on content transfer using a PULL, not PUSH, approach.
Event Types

• **Event types** to be considered for change notifications:

• Tier 1:
  • Regarding resources: create ; delete ; update ; new dump ; new inventory
  • Regarding channels: removed from channel ; added to channel

• Tier 2:
  • Regarding resources: move ; copy

⇒ Focus is on Tier 1 and on events pertaining to single resources.
⇒ Tier 2 and consideration of groups of resources (e.g. URI templates, URI regex) is postponed.
Resource Representations

• Resource representations are subject to synchronization:
  • But limited to those that can be requested using protocol parameters for the resource’s URI.

=> Caveats:
  • It is understood this causes problems in light of resource representation *personalization*, e.g. geo-dependent representations.
  • If the change notification function is provided by 3rd party instead of Source, change notifications may be *subjective* to the 3rd parties perspective.
Discovery

• Several Discovery needs arise:

  • Regarding change notification channels:
    • How to find channels that provide change notifications for a given Source Server?
    • How to find information about the nature of the changes that are communicated on such channels?
  
  • Regarding dumps and inventories:
    • Where to obtain the most recent dump?
    • Where to obtain the most recent inventory?
  
  • Regarding memory:
    • Where to obtain digests?
    • How to access Mementos?
And Also …

• **Authorization**: For baseline matching, resource synchronization (change notification, content transfer), and audit a distinction may be required between Destinations that are authorized or unauthorized to perform these functions.

• **Embargo**: How soon after a resource change are parties allowed to know about it, to act upon it?

• **Trust** in a change notification channel:
  • Can a change notification channel that provides information about a given Source Server be trusted?
ResourceSync

Problem Perspective

Technical Directions

An Experiment

Q&A
ResourceSync Architectural Challenges

1. Initial Baseline Synchronization ~ Dump
2. Incremental Synchronization
   1. Change Notification (CN)
      • Alert that something happened (create, update, delete)
   2. Content Transfer (CT)
      • Transfer of just the change or the full resource
3. Audit to ensure Consistency
Incremental Synchronization - Architectures

Trivial Approach:
- Retrieve every resource and compare to current copy
  - Not scalable: too many wasted, large transactions
  - No way to discover newly created resources
Incremental Synchronization - Architectures

Theoretically Optimal Solution:
- Whenever a resource changes, push only the change to the appropriate Destinations
- No wasted transactions, only as much data transferred as needed
- Newly created resources are discovered
- But overly burdens the source! Not economically viable
- Q: Sweet point between Trivial and Optimal?
Incremental Synchronization - Architectures

Trivial Approach plus Conditional GET (If-Modified-Since):
  • Retrieve every resource if it has changed
  • Still not scalable: too many wasted transactions
  • Still no way to discover newly created resources
Incremental Synchronization - Architectures

Simplest Workable Model:
- Introduce a Feed of change notifications for all resources
- Atom, RSS, OAI-PMH, SiteMaps, etc.

- Significantly reduces wasted transactions
- Newly created resources are discovered
- But still not very efficient as Destination doesn’t know when to pull
Incremental Synchronization - Architectures

Feed Extension Solution:

- Continue the Feed paradigm, but introduce an aggregating Service, and have the Source ping the Service when there’s a change (simulated push)

- No wasted transactions, but pull will get already-seen notifications
- Only advantageous if Source already supports a Feed
Incremental Synchronization - Architectures

Push Solution:
- Instead of ping+pull, Source can push the change notification

- No wasted transactions, no wasted data transfer
- Service maintains subscriber list, not Source
- Change Notification from Source is easier than a Ping+Feed if no feed is already available, and minimally harder than just a Ping
Change Notification - Protocols

- Atom PubSubHubbub (PuSH)
- XMPP
  - PubSub extension
  - BoSH (XMPP over HTTP)
- Comet / HTTP Streaming
  - Open an HTTP connection and keep reading from it
  - Bayeux Protocol
- Long Polling
  - Keep HTTP connection open until a message, then reopen
  - BoSH, Bayeux option
- WebSockets
  - NullMQ / ZeroMQ
  - XMPP over WebSockets?
Ongoing Investigations

- Change Notification - XMPP & XMPP PubSub & bleeps
  - LANL
  - Ongoing Experiment with LiveDBPedia
- Change Notification - Comet / HTTP Streaming & bleeps
  - ODU
  - Bayeux Protocol via Faye Implementation
- Change Notification - Change Simulator
  - Cornell U
  - Generate configurable change notifications
  - Use as standardized input to different systems for testing
- Baseline Matching & Audit
  - Cornell U
  - Looking into Sitemap protocol extension
ResourceSync

Problem Perspective

Technical Directions

An Experiment

Q&A
LiveDBpedia Synchronization Experiment

- Propose a synchronization approach:
  - Push for resource change notification: XMPP PubSub
  - Pull for resource synchronization: HTTP GET

- Use LiveDBpedia’s ad-hoc synchronization tool to create a LiveDBpedia-LANL.

- Test the proposed approach to allow third parties:
  - To be aware of changes to subject-URIs of LiveDBpedia-LANL
  - To synchronize with subject-URIs of LiveDBpedia-LANL
<XMPP Intermezzo>
XMPP

Extensible Messaging and Presence Protocol - XMPP: Client to Client(s) communication with help of intermediate servers (cf IM)

- Based on Jabber (1999)
- See http://xmpp.org
- 3 Core RFCs:
- Multitude of extension specifications, see http://xmpp.org/xmpp-protocols/xmpp-extensions/
- Extensive toolkit: clients, servers, libraries – see http://xmpp.org/xmpp-software/
XMPP Architectural Diagram

message to jane@bar.org

joe@foo.org → foo.org → bar.org → jane@bar.org
XMPP PubSub

XMPP Publish-Subscribe: Client to Subscription Service, Subscription Service to Client(s) communication (cf Twitter)

- Apple Notifications based on XMPP PubSub
- Available tools, see http://xmpp.org/about-xmpp/technology-overview/pubsub/#impl-client
  - XMPP Servers with PubSub support:
    - ejabberd, OpenFire, Tigase, SleekXMPP
  - XMPP libraries with PubSub support:
    - Strophe (C, JavaScript), XMPP4R (Ruby), SleekXMPP (Python), PubSub Client (Python)
XMPP PubSub Architectural Diagram
</XMPP Intermezzo>
LiveDBpedia-LANL Experiment:

Testing a Push/Pull ResourceSync Approach
LiveDBpedia-LANL Experiment:

Testing a Push/Pull ResourceSync Approach

Experiment conducted by:

Luydmilla Balakireva
Martin Klein
Michael L. Nelson
Robert Sanderson
Harihar Shankar
Herbert Van de Sompel

D-Lib Magazine paper forthcoming
Characteristics of Push/Pull Synchronization Approach

• Overall approach:
  • Change Notification – Push - XMPP PubSub
  • Resource Synchronization – Pull – HTTP GET
XMPP PubSub for ResourceSync Experiment

- LiveDBpedia-LANL sends change notifications using XMPP PubSub.
  - Has multiple PubSub Nodes on a 3rd party PubSub server, e.g.:
    - DBpedia_all
    - DBpedia_business
    - DBpedia_music
    - ...
  - LiveDBpedia-LANL sends change notifications to appropriate PubSub Node
  - PubSub Node passes change notifications on to Node Subscribers
Characteristics of Push/Pull Synchronization Approach

- Change types supported:
  - Create / Update / Delete for HTTP-URI-identified resources
  - No Move ; No Copy ; No Fragment Update
  - No HTTP-URI wildcarding
Characteristics of Push/Pull Synchronization Approach

- Notification language - Inspired by Tweets (but for machines)
  - General-purpose `<bleep>` element carried in XMPP `<item>`
  - Content of `<bleep>`:
    - Text that is writeable & readable by machine and human
    - In the experiment
      - *URI created at=* "time-of-create" #hashtag
      - *URI updated at=* "time-of-update" #hashtag
      - *URI deleted at=* "time-of-delete” #hashtag
    - *time* of bleep and *publisher* of bleep in message headers

http://megalodon.lanl.gov/dbpedia/data/Paris
updated at="2012-03-05T19:54:39Z”#dbpedia
Change Notification
<item>
<bleep time="time-of-bleep" publisher="XMPP-client">
subject-URI updated at="time-of-update" #dbpedia $resync
</bleep>
</item>
Bleeps Streaming into Browser

http://megalodon.lanl.gov/strophe/
Resource Synchronization
<item>
<bleep time="time-of-bleep" publisher="XMPP-client"> subject-URI updated at="time-of-update" #dbpedia $resync</bleep>
</item>
Characteristics of Push/Pull Synchronization Approach

- Robustness - notification memory:
  - Zero memory at source/intermediate/destination.
  - Notification memory provided by autonomous 3rd party service:
    - Separate XMPP PubSub server with support for Offline Delivery and Message Persistence;
    - Consumes bleeps from source and archives them;
    - Publishes an hourly digest of archived bleeps;
    - Sends out a $resync notification re the availability of a new digest.
Notification Memory
<bleep time="time-of-bleep" publisher="XMPP-client">
digest-URI created at="time-of-creation"
digestOf="XMPP-URI-of-channel" $resync
</bleep>
Digests

Index of /digest

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_all/</td>
<td>31-Jan-2012 16:00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_applied_sciences/</td>
<td>30-Jan-2012 17:22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_arts/</td>
<td>31-Jan-2012 15:25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_business/</td>
<td>30-Jan-2012 17:22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_culture/</td>
<td>31-Jan-2012 15:25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_history/</td>
<td>30-Jan-2012 17:22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dbpedia_music/</td>
<td>31-Jan-2012 15:25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Apache/2.2.17 (Ubuntu) Server at abacus.seven.research.odu.edu Port 8080

http://abacus.seven.research.odu.edu:8080/digest/
Digest

http://megalodon.lanl.gov/dbpedia/data/Wycombe_Wanderers_F.C._Scott_Rendell__1_deleted_at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/List_of_Super-Heroes_publications updated at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/FBOC_Motorsports_Club updated at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Yoo_Ji-tae updated at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Strong_Enough_to_Bend updated at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Wycombe_Wanderers_F.C. updated at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Kyoto_School updated at="2012-01-31T22:00:00Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Kyoto_design_declaration updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Kyoto_design_declaration updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Portal:Contents/Outlines/History_and_events updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/WikiProject_Hungary/Participants updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Yann_M%27Villa updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Yoo_Ji-tae__1 updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Nicholas_Tomaszewski updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/First_Crusade updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/First_Crusade updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Category:2001_in_sports updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync
http://megalodon.lanl.gov/dbpedia/data/Airspeed_Courier updated at="2012-01-31T22:00:11Z" feature="memento" #dbpedia $resync

http://abacus.seven.research.odu.edu:8080/digest/dbpedia_all/most_recent.txt
Resource Version Memory
<item>
<bleep time="time-of-bleep" publisher="XMPP-client">
	subject-URI updated at="time-of-update"
	feature="memento" dbpedia $resync
</bleep>
</item>
Synchronizing Resource Versions
Discovery of Notification Channels
Discovery of Notification Channels – well-know URI

```
{
    "xmpp:pubsub.xmpp.org?node=dbpedia_applied_sciences": {
        "description": "DBPedia updates from the Applied Sciences Category tree",
        "language": "resync"
    },
    "xmpp:pubsub.xmpp.org?node=dbpedia_music": {
        "hasDigest": "xmpp:pubsub.xmpp.org?node=dbpedia_music_digest",
        "description": "DBPedia updates from the Music Category tree",
        "language": "resync"
    },
    "xmpp:pubsub.xmpp.org?node=dbpedia_business": {
        "description": "DBPedia updates from the Business Category tree",
        "language": "resync"
    },
    "xmpp:pubsub.xmpp.org?node=dbpedia_culture": {
        "description": "DBPedia updates from the Culture Category tree",
        "language": "resync"
    },
    "xmpp:pubsub.xmpp.org?node=dbpedia_history": {
        "description": "DBPedia updates from the History Category tree",
        "language": "resync"
    },
    "xmpp:pubsub.xmpp.org?node=dbpedia_arts": {
        "description": "DBPedia updates from the Arts Category tree",
        "language": "resync"
    },
    "xmpp:pubsub.xmpp.org?node=dbpedia_all": {
        "hasDigest": "xmpp:pubsub.xmpp.org?node=dbpedia_all_digest",
        "description": "All of the DBPedia updates",
        "language": "resync"
    }
}
```

http://megalodon.lanl.gov/.well-known/bleep.txt
Discovery of Notification Channels – Link Header

- curl -I http://megalodon.lanl.gov/dbpedia/data/Mexico

HTTP/1.1 200 OK
Date: Mon, 30 Jan 2012 23:59:41 GMT
Server: Apache/2.2.21 (Unix) DAV/2 mod_fcgid/2.3.7-dev mod_wsgi/3.3 Python/2.4.3
Link: <http://megalodon.lanl.gov/dbpedia/timegate/http://megalodon.lanl.gov/dbpedia/data/Mexico>; rel="timegate", <xmpp:pubsub.xmpp.org;xmpp:pubsub.xmpp.org;xmpp:pubsub.xmpp.org>; node=dbpedia_all>; rel="resyncNode"
Content-Length: 48932
Content-Type: text/html
And it actually syncs!
Average of 2 Change Notifications per Second
Bleeps Versus Queue Size

![Graph showing Bleeps Versus Queue Size with two plots]

- Top plot: 
  - x-axis: Time (2012-02-12T17:41:51Z to 2012-02-13T01:36:51Z)
  - y-axis: Number of messages and max queue size

- Bottom plot: 
  - Similar to the top plot

ResourceSync – Herbert Van de Sompel
CNI Membership Meeting, April 2 2012, Baltimore MD
<table>
<thead>
<tr>
<th>Run</th>
<th>Total</th>
<th>Diff</th>
<th>MaxQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LANL-LIV</td>
<td>LIV</td>
</tr>
<tr>
<td>1</td>
<td>13,819</td>
<td>4 (0.03%)</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>32,453</td>
<td>2 (0.006%)</td>
<td>389</td>
</tr>
<tr>
<td>3</td>
<td>6,910</td>
<td>1 (0.01%)</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>24,400</td>
<td>5 (0.02%)</td>
<td>175</td>
</tr>
<tr>
<td>5</td>
<td>11,850</td>
<td>0 (0.0%)</td>
<td>84</td>
</tr>
<tr>
<td>6</td>
<td>14,937</td>
<td>4 (0.03%)</td>
<td>98</td>
</tr>
<tr>
<td>total</td>
<td>104,369</td>
<td>16 (0.015%)</td>
<td></td>
</tr>
</tbody>
</table>
## Diffs After 8-Hour Runs

<table>
<thead>
<tr>
<th>Run</th>
<th>Total</th>
<th>Diff</th>
<th>MaxQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LANL-LIV</td>
<td>LIV</td>
</tr>
<tr>
<td>1</td>
<td>13,819</td>
<td>4 (0.03%)</td>
<td>120</td>
</tr>
<tr>
<td>2</td>
<td>32,453</td>
<td>2 (0.006%)</td>
<td>389</td>
</tr>
<tr>
<td>3</td>
<td>6,910</td>
<td>1 (0.01%)</td>
<td>93</td>
</tr>
<tr>
<td>4</td>
<td>24,400</td>
<td>5 (0.02%)</td>
<td>175</td>
</tr>
<tr>
<td>5</td>
<td>11,850</td>
<td>0 (0.0%)</td>
<td>84</td>
</tr>
<tr>
<td>6</td>
<td>14,937</td>
<td>4 (0.03%)</td>
<td>98</td>
</tr>
<tr>
<td>total</td>
<td>104,369</td>
<td>16 (0.015%)</td>
<td></td>
</tr>
</tbody>
</table>
Content Transfer Volume in 24 Hours

• LiveDBpedia changesets:
  • Compressed: 149 Mb
  • Uncompressed: 4.2 Gb

• HTTP GET from SFI to LiveDBpedia-LANL:
  • Uncompressed: 1.8 Gb
  • Compressed: 180 Mb
ResourceSync
Towards a Web-Based Approach for Resource Synchronization

Herbert Van de Sompel
Los Alamos National Laboratory
@hvdsomp

ResourceSync is funded by
The Sloan Foundation