XSEDE Canonical Use Cases 8 & 12 - Search for and Publish Resource Information

XUAS Architectural Response

Table of contents
Introduction
  Structure of this document
Canonical use cases 8 and 12
Architectural response
  Search for resource information
  Publish resource information
  Variation CAN8.a - Large search results returned in chunks
Quality attributes
  CAN8.a - Search response time
  CAN8.b - Aggregate search rate
  CAN8.c - Search service availability
  CAN8.d - Search response integrity
  CAN12.a - Publishing response time
  CAN12.b - Aggregate publishing rate and volume
  CAN12.c - Information storage capacity
  CAN12.d - Validation rejection accuracy
  CAN12.e - Validation acceptance accuracy
  CAN12.f - Storage persistence

Introduction

This document describes the realization of Canonical Use Cases 8, Search for Resource Information, and Canonical Use Case 12, Update Resource Information, using the XSEDE XUAS architectural components. See http://hdl.handle.net/2142/46550 for the use cases.
It is assumed that the reader has already read and is familiar with the XSEDE Architecture Level
3 Decomposition (L3D), in particular §3 (Access Layer) and §6 (XUAS). The authors suggest
the L3D document be open or on hand when reading this document.

**Structure of this document**

This document comprises two sections. §2 reviews the use cases and §3 describes how the
XSEDE components are used to implement the use cases from §2.

**Canonical Use Cases 8 and 12**

Canonical use case 8 is “Search for Resource Information.” Canonical use case 12 is “Update
Resource Information.” The descriptions for the two use cases are as follows.

- 8 - Synchronously search or query for resource information from the information system.
- 12 - Synchronously add or update resource information in the information system.

The two use cases begin with a number of assumptions, specifically:
1. The Searcher and the Publisher both know how to contact and communicate with the
   Information System.
2. The term “information system” does not imply a single central service.
3. When required, either the Searcher or the Publisher authenticates with the Information
   System and is authorized.
4. Authentication and Authorization are consistent with UCCAN 9.0 and the relevant
   security architecture.
5. Searchers and Publishers will comply with XSEDE wide information policies.
6. The Publisher is responsible for the quality of the published information (e.g., it is
   accurate, it is up to date, it conforms to any specified schemas).
7. Publishers include appropriate timestamps and validity time (time to live) in the resource
   information they update.
8. Publishers must be able to communicate securely with the information system when they
   wish to.

Canonical use case 8 describes how a searcher submits a search to the information system and
receives a response.

Canonical use case 12 is a bit more complicated. A publisher obtains information about a
resource, sends the resource information to the information system, and receives an
acknowledgement from the information system. On explicit acknowledgement of acceptance, the
publisher can be assured that the resource information has been updated in the information system. Optionally, the information system may validate the resource information against previously defined validation rules and reject resource information that doesn’t conform to the rules.

Table 1 lists the single variation for the two use cases.

<table>
<thead>
<tr>
<th><strong>UCCAN 8 &amp; 12</strong></th>
<th><strong>Search for and update resource information</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>VAR-CAN8.a</td>
<td>Large search or query responses may be returned in pages/chunks.</td>
</tr>
</tbody>
</table>

Table 2 lists the quality attributes for the two use cases.

<table>
<thead>
<tr>
<th><strong>UCCAN 8 &amp; 12</strong></th>
<th><strong>Search for and update resource information</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>QAS-CAN8.a</td>
<td>A Searcher will receive a response to a search/query request returning 1 MB or less in less than 5 seconds 90% of the time. [source: A&amp;D]</td>
</tr>
<tr>
<td>QAS-CAN8.b</td>
<td>The minimum aggregate rate at which all Searchers can perform searches. 2 searches per second.</td>
</tr>
<tr>
<td>QAS-CAN8.c</td>
<td>Search requests to the Information System will succeed 99.9% of the time [source: A&amp;D]</td>
</tr>
<tr>
<td>QAS-CAN8.d</td>
<td>Query results are delivered to the searcher without modification (integrity) 99.9% of the time [source: A&amp;D]</td>
</tr>
<tr>
<td>QAS-CAN12.a</td>
<td>When the information system is providing acknowledgements to the Publisher, the amount of time it takes to publish information and receive acknowledgement is less than 5 second. [source: A&amp;D]</td>
</tr>
<tr>
<td>QAS-CAN12.b</td>
<td>The aggregate rate at which all Publishers can update information. 2 updates/sec and 100 KB/sec.[source: job state updates rates]</td>
</tr>
<tr>
<td>QAS-CAN12.c</td>
<td>The amount of resource information that can be stored by the Information System: 100 million entries, 100 GB of data total. [source: job volume, accounting volume]</td>
</tr>
<tr>
<td>QAS-CAN12.d</td>
<td>Information that does not conform with specified schemas and integrity rules will be rejected by the Information System 99.999% of the time. [source: A&amp;D]</td>
</tr>
<tr>
<td>QAS-CAN12.e</td>
<td>Update requests that conform with specified schemas and integrity rules will succeed 99.9% of the time [source: A&amp;D]</td>
</tr>
<tr>
<td>QAS-CAN12.f</td>
<td>Once information has successfully been accepted by the system, it will be available for retrieval in the system for one year 99% of the time unless it is explicitly deleted. [source: A&amp;D]</td>
</tr>
</tbody>
</table>
Architectural response

Prerequisites (Dependencies) & Assumptions

1. The searcher and the publisher know the URL to the appropriate XSEDE information service. The Search for Information System URLs sub-section below describes how the XUAS architecture supports the discovery of information system URLs.
2. The publisher is authorized to publish information about the resource.
3. The searcher and publisher have access to the Internet.
4. All search and update interactions follow the Web/Cloud approach (e.g., REST, HTTP, SSL/TLS, OAuth) described in L3D §6.2.
5. All information responses return JSON content by default, and some may optionally return other encodings (like XML, HTML, and CSV) determined by user requirements and each implementation.
6. The searcher and publisher will authenticate to XSEDE per UCCAN 6/9 Authentication in order to a) publish information, b) search for sensitive/restricted information, or c) retrieve user customized/specific information.
7. The term user in this document primarily refers to applications/software that search or update resource information, but may in some cases mean literal persons (developers, staff, or expert users) who search or update resource information using a browser.

Search for Resource Information

Using a URL discovered thru the methods described in Search for Information System URLs below, the Searcher invokes the GET method, optionally specifying supported HTTP parameters, that represent the desired Search.

The Information System will respond in one of the following ways:

1. With an error if the specified URL or parameter aren’t supported by the interface
2. With a response indicating NO information satisfies the specified Search parameters
3. With all the information that satisfies the specified Search parameters
4. With some of the information that satisfies the specified Search parameters, plus an indication that additional information is available and the address to use to obtain the additional information.
The following sub-section describes a special case search: search for information systems and their URLs.

**Search for Information System URLs**

XSEDE provides a collection of information interfaces that Search and Publish a broad set resource information. A specific type of resource information may be available thru multiple interfaces (URLs) distinguished by whether the user is Searching or Publishing, interface quality attributes, the type of user accessing the interface, and whether information is combined with other information. For example, an interface that provides direct read-write access to XSEDE accounting information in XDCDB may be restricted to a limited set of users, while a read-only summary of that same accounting information may be cached in an information service and available to a broader set of users.

The XUAS architecture provides a special case interface that enables searching and discovery of information systems and interface URLs. This special search enables Searchers and Publishers to discover URLs that access the type of resource information of interest and that satisfy the quality attributes of interest.

User Steps:

1. The user (person) may launch a web browser and connect to the XUAS Directory Service Interface at [http://info.xsede.org/service/directory/](http://info.xsede.org/service/directory/) (described in L3D §6.1.3.1) to search for information about XSEDE XUAS information services and interfaces/URLs.

2. The user may access the [http://info.xsede.org/service/directory/v1/usage](http://info.xsede.org/service/directory/v1/usage) URL for a complete description of the Directory Service Interface, including (sub)queries, available methods (GET, POST, PUT, DELETE), available information encoding formats (JSON, XML, HTML, and CSV), and details about relevant information schemas.

3. The user may follow Directory Service returned URLs to navigate to the content of other information system, or copy the URL of those interfaces to use in software that needs to search or publish information.

Software Steps:

1. Software may Search via RESTful GET the XUAS Directory Service Interface at [http://info.xsede.org/service/directory/v1/](http://info.xsede.org/service/directory/v1/) (described in L3D §6.1.3.1) to retrieve information about all XSEDE XUAS information services and interfaces/URLs.

2. Software may use Directory Services returned URLs to navigate to the content of other information systems.
Update Resource Information

Using a URL discovered thru the methods described in *Search for Information System URLs* above, the Publisher executes the appropriate RESTfull PUT, POST, or DELETE method, optionally specifying supported HTTP parameters, and providing the content (in the case of PUT or POST) to be published in a supported encoding.

The Information System will respond in one of the following ways:
1. With an error if the specified URL or parameters aren’t supported by the interface
2. With a response indicating that the specified resource to be updated (PUT), or deleted (DELETE), doesn’t exist.
3. That the supplied information was successfully Published to the information system.

Quality attributes

Note: Currently, all XUAS information services cache information on the service host to respond to searches. The cache is populated using back-end processes that are decoupled from the service interfaces. All of the responses to quality attributes below assume that this is the case. However, we know that for some of the planned (not yet implemented) search services--such as those that access data in the XDCDB--caching will not be practical because the data sizes and security requirements will be prohibitive. The responses provided here will not be adequate for those services. We believe that the best way to address this is for us to explicitly state in service documentation and registries for these (future) services that they do not utilize caching and thus can not guarantee these quality attributes.

Variations

**CAN8.a - Large search results returned in chunks**

This variation is described in the response above to “Search for resource information.” Specifically, response type (4) is used to break the response into pieces and return the first “chunk,” and the user is then able to choose when to request each piece.

**CAN8.a - Search response time**

Attribute: A Searcher will receive a response to a search/query request returning 1 MB or less in less than 5 seconds 90% of the time. [source: A&D]
Response: Requests to the search service can be broken down into the following steps.

1. User’s web browser (or REST client) transmits HTTP request over Internet to search service.
2. Search service decodes the request.
3. Search service performs the search locally on cached data and encodes the response.
4. Search service responds to client via HTTP over Internet.

XSEDE does not have control over the user’s connection to the Internet, which is used for the initial request and final response transmission. We assume that the time for steps 1 and 4 above is not included in the one second. However, we also note that Step 2 requires only a fraction of the second, leaving most of the time for Step 3.

Step 3 requires the search service to perform a search on its local cache, which is populated by other software components. With the current hosting arrangements, both our testing and our continuous operations demonstrate that the time required to complete the local file access and query is far less than one second.

**CAN8.b - Aggregate search rate**

**Attribute:** The minimum aggregate rate at which all Searchers can perform searches. 2 searches per second.

Response: The search service implements transfer requests using a REST-style interaction between the client and the search service. As described above, each REST interaction involves communication between the REST client and server and the REST server and the search service’s local cache. The round trip for each request is far less than a second. The REST service is multi-threaded and engineered to support dozens of simultaneous HTTP connections. The system is easily able to handle two requests/second from the aggregate of these connections.

**CAN8.c - Search service availability**

**Attribute:** Search requests to the Information System will succeed 99.9% of the time [source: A&D]

Response: We assume that “succeed” means that the service will respond to a properly formed query with a result—a set of zero or more matching “hits”—as opposed to no response from the service or a response indicating that the service is unavailable. We also assume that the
maximum aggregate search rate (2 searches per second) is not exceeded.

Examining the steps listed above in CAN8.a, we have no control over the user’s connection to the Internet, which is used for steps 1 and 4. Faults at these points are subject to TCP/IP’s ability to guarantee transmission and receipt of messages. (TCP/IP does not guarantee the ability to establish a connection from client to service.) Steps 2 and 3 are carried out entirely within the service code and its host environment, which includes a host operating system, HTTP server, Web service container, and any information service provider utilities that run on the host system. As long as the host operating system and HTTP server are reasonably stable (built on commercial or widely used open source products and professionally installed and operated), the chance of the OS or HTTP service running out of resources or otherwise preventing the service code from running is very, very small. (Much smaller than 0.1%) It is vital that the host operator use appropriate quotas (storage) and limits (CPU, memory) on information service provider utilities so that it is impossible for them to disrupt the host system.

The remaining possibility is a host operating system failure (a power failure or a system “crash”) or a network partition that prevents contact with the host system. In order to ensure a 99.9% availability of the service, we recommend that the host operating system be managed according to the “high availability” practices described in [L3D §9.3].

XSEDE’s XUAS information services are currently being operating in the manner described. Specifically, there are two host instances running at two separate physical sites. A dynamic DNS entry points to the currently “active” instance. Changing the dynamic DNS entry takes only moments, but it must be initiated by a human operator when a failure is detected. Failures are detected by XSEDE’s Inca system. Failure notifications are delivered to human operators via email.

**CAN8.d - Search response integrity**

*Attribute:* Query results are delivered to the searcher without modification (integrity) 99.9% of the time [source: A&D]

*Response:* The REST interface relies on TCP to ensure that the data transmitted from the service to the client is not modified in transit. TCP provides considerably higher than a 99.9% integrity rate. This leaves only the service implementation itself as a source of modifications to the search results. The correctness of the service implementation itself is the subject of unit tests, which are developed and executed by the software developers as a routine part of code maintenance. In order for code to be delivered to XSEDE, all unit tests must pass, which means that all known failure conditions are properly handled.
CAN12.a - Publishing response time

Attribute: When the information system is providing acknowledgements to the Publisher, the amount of time it takes to publish information and receive acknowledgement is less than 5 second. [source: A&D]

Response: Requests to the publishing service can be broken down into the following steps.

5. User’s web browser (or REST client) transmits HTTP request over Internet to publishing service.
6. Publishing service decodes the request.
7. Publishing service performs the search locally on cached data and encodes the response.
8. Publishing service responds to client via HTTP over Internet.

XSEDE does not have control over the user’s connection to the Internet, which is used for the initial request and final response transmission. We assume that the time for steps 1 and 4 above is not included in the one second. However, we also note that Step 2 requires only a fraction of the second, leaving most of the time for Step 3.

Step 3 requires the publishing service to validate the data against relevant schema and write the data its local cache. With the current hosting arrangements, both our testing and our continuous operations demonstrate that the time required to complete the validation and local storage is far less than five seconds.

CAN12.b - Aggregate publishing rate and volume

Attribute: The aggregate rate at which all Publishers can update information. 2 updates/sec and 100 KB/sec.[source: job state updates rates]

Response: The publishing service implements transfer requests using a REST-style interaction between the client and the search service. As described above, each REST interaction involves communication between the REST client and server and the REST server and the publishing service’s local cache. The round trip for each request is far less than a second. The REST service is multi-threaded and engineered to support dozens of simultaneous HTTP connections. The system is easily able to handle two requests/second from the aggregate of these connections.
CAN12.c - Information storage capacity

Attribute: The amount of resource information that can be stored by the Information System: 100 million entries, 100 GB of data total. [source: job volume, accounting volume]

Response: It is critical that the host system on which the information services are run have sufficient storage to contain the maximum amount of data expected for the system. This is an operational issue, not an architectural issue. 100GB of data is easily stored on a hard disk drive or even a solid-state storage device (SSD).

CAN12.d - Validation rejection accuracy

Attribute: Information that does not conform with specified schemas and integrity rules will be rejected by the Information System 99.999% of the time. [source: A&D]

Response: There is no architectural component to this quality attribute. Data validation is a feature of the code that implements the service, and this feature is tested by developer unit tests. The service should not be deployed unless all unit tests pass, implying that all known validation errors are tested and avoided.

CAN12.e - Validation acceptance accuracy

Attribute: Update requests that conform with specified schemas and integrity rules will succeed 99.9% of the time [source: A&D]

Response: There is no architectural component to this quality attribute. Data validation is a feature of the code that implements the service, and this feature is tested by developer unit tests. The service should not be deployed unless all unit tests pass, implying that all known validation errors are tested and avoided.

CAN12.f - Storage persistence

Attribute: Once information has successfully been accepted by the system, it will be available for retrieval in the system for one year 99% of the time unless it is explicitly deleted. [source: A&D]

Response: The concern here is the likelihood of data being lost from the system without a way to restore it. (Again, correctness of the software implementing the information services is verified by unit tests, and the system is not deployed unless all unit tests pass.) In order to ensure that data is never unintentionally lost, the services must be operated on host systems with routine backups and a verified restore capability.
Note that quality attribute CAN8.c requires operation of redundant services. Both host instances should be backed up. In the unlikely event that one of the instances is lost or corrupted and the backup can not be restored, the other instance's backup can be used. Also note that CAN.12c requires a minimum of 100GB of active storage. This implies that the backup systems must each be able to retain multiple copies of the 100GB of active storage. This is well within the capacity of off-the-shelf backup systems.

Using these methods, 99% persistence is extremely likely.