Federation Dispatcher

The *Federation Dispatcher* stage takes a job (most likely originating from the HTTP Feeder) and *dispatches* it to a number of applications (which normally include stages such as the FAST Query Builder and Fetch URL) that perform a query on a search engine and place the results back in their Aspire document.

The dispatcher then collects these federated results, but does not merge them. That is done by Merger.

Federation Dispatcher	
Factory Name	com.searchtechnologies.aspire:aspire-federation
subType	default
Inputs	Aspire Jobs
Outputs	Aspire Jobs

? Unknown Attachment

Detail

The dispatcher holds a list of *servers* to post queries to. These *servers* are organised in to *zones*. When the job is received, the stage attempts to find a *zone* to federate to by looking at a tag in the incoming document (or failing that may be configured with a default). Once the *zone* has been established, the dispatcher will get the *servers* for the *zone* and create a child job for each *server*. This job will be published via the branch handler, configured with a route to a *query application* that is associated with the *server*.

This *query application* will typically be search vendor specific and will do any conversion of parameters required in order allow the search to executed and the results loaded, typically using stages such as the FAST Query Builder, Fetch URL and XML Loader.

The *Federation Dispatcher* uses the job listener to see the completion of the child jobs and take the results from the individual search and adds then to the parent job

Servers

The list of configured servers held by the *Federation Dispatcher* defines the list of search applications that may be queried. The definition of a server consists of an *id* for identification (which should be unique), a *searchUrl* which defines the actual URL on which the search should be executed and a *query application* which is an Aspire application which performs the actual search and handles any conversion required (both before the query to convert parameters and after the query to load and convert results to the form required by the Merger.

A simplistic definition is shown below:

```
<server id="fast" searchUrl="http://host/cgi-bin/xsearch" queryApplication="/FederationFastQuery" />
```

Blacklisting

Search servers may be inaccessible at times. This can take time to establish and therefore slow the delivery of results to the client. When a job has been dispatched, the dispatcher will assume after a certain timeout (default 15s) that the server is unavailable and will return without that result set. When this happens, a *failure count* for the server is incremented. When this count reaches a certain threshold (default 3), the server will be *blacklisted* and no further jobs will be dispatched for this server for a period of time (the *blacklist period* - default 15 minutes). Once the *blacklist period* has passed, the failure count for the server will be reset and jobs dispatched to it once more.

Parameters can be set in the server definition thus:

```
<server id="fast" searchUrl="http://host/cgi-bin/xsearch" queryApplication="/FederationFastQuery"
blacklistThreshold="5" blacklistPeriod="180000" />
```

Boosting when merging

When merging by rank, you may require results from a certain server to be boosted. This can be achieved by setting the *boost* attribute.

```
<server id="fast" searchUrl="http://host/cgi-bin/xsearch" queryApplication="/FederationFastQuery" boost="2.5"
/>
```

Server parameter set

The full server parameter set is shown below:

Element	Type	Default	Description
server/@id	String		**Mandatory** The id of the server.
server/@searchUrl	String		**Mandatory** The URL of the search server application for this server.
server/@queryApplication	String		**Mandatory** The Aspire application for which jobs for this server should be routed.
server/@boost	Float	1.0	The boost factor for results from this server when the Merger is merging by rank.

server/@blacklistThreshold	int	3		The number of server failures (in a row) that will cause this server to be blacklisted.
server/@blacklistPeriod	long	900000 ms (=15 minutes)		The period in ms for which a server will remain blacklisted once it reaches the blacklist threshold.

Zones

Zones allow grouping of servers to perform federation. You need at least one *zone*, and a *zone* without any servers does not make sense. *Zones* are identified by an *id* (which should be unique) and reference servers by *id*s that should match those in the server configuration.

When the *Federation Dispatcher* receives a job, it looks in the attached document for a *zone* (or falls back to a default). Once the *zone* has been established, the dispatcher publishes jobs for each server configured in the *zone*

A simple zone definition would be:

```
<zone id="zoneOne">
   <server id="server1"/>
   <server id="server2"/>
</zone>
```

As mentioned above, servers may be unavailable, and the dispatcher times out requests after a certain period of time. This timeout may specified in the *zone* definition:

```
<zone id="zoneOne" timeout="15001">
   <server id="server1"/>
</zone>
```

Zone parameter set

The full zone parameter set is shown below:

Element	Type	Default	Description
zone/@id	String		**Mandatory** The id of the zone.
zone/@timeout	long	15000 ms (=15 seconds) but can be globally overriden	The time out for requests in this zone.
zone/@mergeType	String		The suggested method used for merging (merging in performed in the Merger).
zone/server/@id	String		**Mandatory** The id of the server to add to this zone (multiples allowed) .

Result Collection

The dispatcher does not merge the results from the federated *query applications* (this is done by the Merger) but it does collect them under a single tag in document passed to the stage. The primary reason for this action is that the jobs containing individual results may not be available when the job from this stage reaches the merger.

During result collection, the dispatcher looks for a named tag in the document from the child job from the federated query, and adds it as a child of the parent document, resulting in a single node in the parent document containing multiple children, where each child contains the results from a single search

Configuration

The following configuration items are supported:

Element	Type	Default	Description
federationResultTag	String	aspireFederationResult	The document tag to hold all of the federation result sets.
resultTag	String	SEGMENTS	The result tag in the federated results set.
federationEvent	String	onFederation	The event to publish the federation jobs on.
federationBranchEvent	String		The branch to set on the job if federation occurred.
noFederationBranchEvent	String		The branch to set on the job if federation did not occur.
zonePath	String	/doc/federationZone	The element in the document that hold the name of the *zone* to federate to.
defaultZone	String		The default *zone* if one is not found in the document.
timeout	long	15000 ms =15s	The default timeout for *zones*. After this period, outstanding federation queries will be deemed to have failed.
blacklistThreshold	long	3	The number of federation queries that can fail before the server is blacklisted.

blacklistPeriod	long	900000 ms =15m	The default period for which a blacklisted server will remain blacklisted.
servers	see *Servers* above		**Mandatory** One or more *servers* specifying where queries should be federated to.
zones	see *Zones* above		**Mandatory** One or more *zones* specifying how queries should be federated.

Example Configuration

```
<component name="Dispatcher" subType="default" factoryName="aspire-federation">
   <debug>${debug}</debug>
   <zonePath>${zonePath}</zonePath>
   <defaultZone>${defaultZone}</defaultZone>
   <servers>
     <server id="server1" searchUrl="http://myServer1/mySearchUrl1" queryApplication="/federateQuery1"
blacklistThreshold="11" blacklistPeriod="180001" boost="0.1" />
     <server id="server2" searchUrl="http://myServer2/mySearchUrl2" queryApplication="/federateQuery2"
blacklistThreshold="12"  boost="0.2" />
     <server id="server3" searchUrl="http://myServer3/mySearchUrl3" queryApplication="/federateQuery3" />
     <server id="server4" searchUrl="http://myServer4/mySearchUrl4" queryApplication="/federateQuery4" />
     <server id="server5" searchUrl="http://myServer4/mySearchUrl5" queryApplication="/federateQuery5" />
   </servers>
   <zones>
     <zone id="zoneNone" />
     <zone id="zoneOne" timeout="15001">
       <server id="server1" />
     </zone>
     <zone id="zoneTwoOne" />
       <server id="server1" />
       <server id="server2" />
     </zone>
     <zone id="zoneTwoTwo">
       <server id="server3" />
       <server id="server4" />
     </zone>
     <zone id="zoneAll" timeout="2500" mergeType="rank">
       <server id="server1" />
       <server id="server2" />
       <server id="server3" />
       <server id="server4" />
     </zone>
   </zones>
   <branches>
     <branch event="onFederation" pipelineManager="../FederationPipelineManager"/>
   </branches>
</component>
```