Multi-Node Installation

If you need more network bandwidth, CPU, or RAM than can be achieved with a single machine, you can run Aspire Enterprise across multiple machines in a cluster.

This page discusses distributed processing for Aspire Enterprise (distributed processing is not available for Aspire Community).

Why Run on Multiple Machines?

The primary reason to run Aspire across multiple machines is to increase available network bandwidth.

In order to index all documents across a large corporation, a search engine must actually download all documents, extract the text, and index every word in a large index. Naturally, this is a very large amount of data, and so downloading it through a single machine can often take a very long time.

Distributing the crawling process across multiple machines means more machines downloading more documents simultaneously through more network cards, allowing Aspire to crawl many more documents per second.

Two Types of Jobs

There are two types of jobs in Aspire: Scanning Jobs and Document Download Jobs. Both types of jobs can be distributed.

Scanning Jobs

A "scanning" job scans directories (or update tables) looking for documents to be processed. For a full scan, this means scanning a content source for all possible documents to be indexed. For an incremental scan, this means looking only for documents which have been added, updated, or deleted.

Many of the connectors implement incremental scans using a "snapshot" file. This is a file which holds a listing of all documents in the content source (plus ACLs, typically). Incremental scanning, therefore, usually involves scanning the content source a second time, and comparing the new set of documents against the old snapshot to identify documents which are new, modified, or deleted.

Therefore, snapshot files will need to be preserved as persisted data from scan to scan.

Document Download Jobs

A document download job fetches the full content of the document from the remote server and extracts text and basic metadata. Since downloading large, multiple-megabyte documents can take a long time, it is most important that document download jobs be distributed across multiple machines in the network.

Levels of Distributed Processing

File:CWSDistributedProcessing.png
Example of Distributed Processing

Aspire Enterprise can be set up to automatically use all available nodes in an Aspire cluster. Distribution of processing jobs occurs at two levels:

- The CS Manager can send scanning jobs to any appropriate connector in the cluster
  For example, if a SharePoint scan is required, the CS Manager will send the job to any "/SharePointConnector" application installed on any node in the cluster.
- Each scanner can send document download jobs to any appropriate connector in the cluster
  For example, the SharePoint scanner will create a sub-job for each individual document to be downloaded. If the local SharePoint connector application (installed at "/SharePointConnector") is full, then jobs will be automatically distributed to any other "/SharePointConnector" installed on any Aspire node in the cluster.

Rules for Distributed Processing

Once you have multiple Aspire machines in your cluster, using these machines requires following three simple rules:

1. Make sure a search engine publisher is installed on *every* machine.
2. Install multiple copies of the connectors on multiple machines.
3. Snapshot directories need to be shared.

Each of these items is discussed in detail in the following sub-sections.

Rule 1: Install A Search Engine Publisher on Every Machine

Once a document is fetched, the connector will automatically extract text and metadata from the document. These records can be large and contain a lot of metadata. Transferring them to another machine before they are sent to the search engine does not make sense.

Therefore, Aspire requires a search engine publisher to be installed on every machine which contains a connector to a content source.

This prevents these large records from having to be unnecessarily transmitted from Aspire Node to Aspire Node before being sent to the search engine.

Rule 2: Install Multiple Copies of your Connectors

In Aspire, any SharePointConnector installed on any machine in a cluster will be able to handle any SharePoint job (directory scanning or document download). This is true of all connectors.

Therefore, if you wish to distribute your jobs across multiple machines, simply install a copy of the same connector on each Aspire node. Jobs will be sent to any node which supports the connector with the same name.

For example, if your content source specifies "/SharePointConnector" as its connector, the scanning job can be sent to any node on which a " /SharePointConnector" application has been installed. If multiple nodes have the application installed, then jobs will be sent, round-robin, to each node in turn.

Rule 3: Snapshot Directories Need to be Shared

Incremental crawling for most connectors requires scanning all documents in a content source and then comparing these documents against a previous snapshot.

In a distributed system, the following situation can occur:

1. When the content source is processed the first time, the scan job goes to node X.
   This creates a snapshot, call it SNAPSHOT_1.
2. When the content source is crawled a second time (an incremental crawl) the scan job now goes to node Y.
   Node Y will need access to SNAPSHOT_1 in order to determine what documents are new, updated, or deleted.

Therefore, in a distributed system, all snapshot directories need to be stored on shared storage which is available to all nodes in the network. This can be done with NFS mounts or using SAN storage.

Note that no two jobs will ever be reading or writing a snapshot at the same time. The Aspire Scheduler carefully controls scanning jobs such that a snapshot for a content source can only be processed by a single job (running on a single node/computer) at a time.

Every connector in Aspire has an option for specifying the location of the snapshot directory. In a distributed environment, this directory should be modified so that it refers to shared storage across all machines in the cluster.