



Geolocation-based Load Balancing Algorithm for GSLB Members

Avi Technical Reference (v17.1)

Geolocation-based Load Balancing Algorithm for GSLB Members [view online](#)

The geolocation algorithm of the Avi DNS for GSLB directs client requests to the optimal site based on the longitude and latitude of the client and GSLB sites. Each SE in the Avi DNS configuration refers to its own local copy of the geolocation database (geo-DB) to make that determination.

Operation

Operation of the geolocation algorithm is best explained by example. Refer to figure 1.

A client with IP address = client_IP seeks service from a multi-site application implemented by four virtual services (VS-A1 through VS-A4) at four geographically separated data centers, DC1-DC4, one of which is a non-Avi load-balanced site (tersely referred to as an external site). The client's local DNS must resolve the FQDN for the application to one of four potential addresses, vip-1, vip-2, vip-3, or vip-4. The first step is to find a DNS capable of making that translation. Either DNS-1 or DNS-2 will do. Each DNS instantiation has access to its own, identical geo-DB. Using that database, either DNS-1 or DNS-2 can translate the five IP addresses into latitude-and-longitude-based locations.

Note: The selection of DNS-2 is outside the control of Avi Vantage, and thus may have nothing to do with geographical location. We have intentionally chosen a more distant DNS to underscore this fact.

```
{% include figure.html title="Avi DNS is authoritative for the FQDN of the multi-site application" url="img/DNS-is-queried.png" alt="Avi DNS is authoritative for the FQDN of the multi-site application" caption="Figure 1. An authoritative DNS for the FQDN is found" %}
```

The algorithm running in DNS-2 calculates four distances, determines that dist-1 is the shortest, and therefore responds to the DNS request with vip-1. As figure 2 shows, the client thereafter directs its application request to vip-1.

```
{% include figure.html title="Geolocation algorithm running in DNS-2 selects vip-1" url="img/client-directs-request-to-closest-site.png" alt="Geolocation algorithm running in DNS-2 selects vip-1" caption="Figure 2. Geolocation algorithm running in DNS-2 selects vip-1" %}
```

Assigning Geographic Locations to VIPs

Each GSLB service pool's algorithm is independently set. Current choices are the round-robin, consistent hash, and geolocation algorithms. To implement the geolocation algorithm, the Avi DNS for GSLB must have a way to assign locations to GSLB sites comprising a GSLB service. The choices on a per-GSLB-service basis are as follows:

```
<li><strong>Compute location from geo-DB (default)</strong>. This method requires a look-up in a database const
<li><strong>Inherit location from site</strong>. With this methodology, each participating GSLB site is assigned
<li><strong>Explicit configuration</strong>. This manual process gives the administrator full control on the mo
```

When the geolocation algorithm is chosen for a given GSLB service, member VSs having VIPs for which locations cannot yet be determined are essentially opaque. DNS responses will never include their VIPs. Instead, the geolocation algorithm will

chose a VIP from among the set of VSs whose locations *can* be determined. If not a single member VS has a VIP whose location can be determined, the Avi DNS for GSLB will fall back to the (weighted) round-robin algorithm.

Additional Details

```
<li>Since the distance calculation is based purely on a virtual service's VIP, both Avi and external sites are
<li>DNS-2 would have returned vip-2, or vip-3, or vip-4, in geo-proximity order, had health monitoring determin
<li>As with the round-robin algorithm, it is possible to assign <strong>weights</strong> to the VIPs. So far, w
<li>As mentioned in <a href="https://avinetworks.com/docs/17.1/gslb-architecture-and-object-model/">Avi GSLB A
```

1. There are times when a client's IP address does not yield an accurate client location. To address this, the administrator may choose to modify the DNS application profile to enable [EDNS](#). If this option is selected, and an EDNS subnet extension is present in the DNS request, then it will be used instead of the client's IP address. For more information, refer to the [Extension Mechanisms for DNS Client Subnet Option Insertion](#) article.

The Geo-DB

In the lifecycle of an Avi GSLB configuration, geographic information for geolocated services is first loaded into the GSLB leader Avi Controller, in bulk or on a location-by-location basis. The data are placed into Avi geo-DB format (specified below) for download into all Avi DNS SEs in the SE group. As a reminder, this group is dedicated to DNS; no application VSs are allowed in it. The Avi geo-DB is subsequently replicated to all follower Avi DNS instantiations, so that any one of them (DNS-1 and DNS-2 in the above example) can reply to client-local DNSs with the same optimally located VIP.

Geo-DB Location Record Format

Fields in an Avi geo-DB record are sufficient in number and kind to accommodate information defined in popular 3rd-party geolocation databases. A single record contains six comma-delimited fields. The first four are mandatory. A hyphen in the fifth or sixth position indicates the field is unspecified. Those fields are:

```
<li><strong>IP address range start</strong></li>
<li><strong>IP address range end</strong></li>
<li><strong>Latitude</strong></li>
<li><strong>Longitude</strong></li>
<li>Full name of city, e.g., USA/California/San Francisco</li>
<li>Tag is reserved for special uses, e.g., to store a customer-defined value for region, say "West"</li>
```

The 'name' and 'tag' fields are used for two purposes:

1. They appear in logs.
2. They can be used to merge location entries, to make the memory usage less and lookups faster, by making the lookups coarser.

Geo-DB File Format and Grammar Rules

1. Any line starting with a leading '#' is ignored as a comment.
2. Location record lines may not contain leading or embedded spaces.
3. The first line must contain V.XX, where XX is the version of the Avi DB. At the time of this writing, the only supported value is V.01.
4. The optional name field is comprised of three parts, and the parts are delimited by two slashes.
5. Any line with a parsing error is dropped. If there are more than 95% errors, the geo-DB loading operation fails.

- If there are lines with duplicate or overlapping addresses, the line last specified in the file dominates, and overwrites the other entries.

The following example illustrates proper syntax.

```
V.01
# start_ip,end_ip,latitude,longitude,country/region/city/tag
# Hyphens denote an unspecified region and city within Australia
1.0.0.0,1.0.0.255,-33.4940,143.2104,Australia/-/-,-
1.0.1.0,1.0.1.255,26.0614,119.3061,China/Fujian/Fuzhou,-
1.0.2.0,1.0.3.255,26.0614,119.3061,China/Fujian/Fuzhou,-
# Neither the name nor the tag fields are specified for the IP range 2.0.2.0 to 2.0.3.255
2.0.2.0,2.0.3.255,26.0614,119.3061,-,-
```

Multiple files can be specified as input for source-IP-address-to-location mapping. As described later in this article, a priority mechanism determines which data should dominate if and when multiple sources stipulate a distance for a given IP address.

Conversion from other formats to the Avi format is possible. Such data is merged into *one*, static, composite geo-DB file in Avi format. In the first release, MaxMind format will be importable. Over time, the design may support import of data from other sources.

Clients of multi-site applications are assumed to have public or private addresses. From the perspective of Avi GSLB, they have *direct* network access to either public or private VIPs, but not both simultaneously. In public networks, network address translation may engage private sites, but such private IPs are transparent to the geolocation algorithm that governs public-client access. Accordingly, the geo-DB is responsible to contain one or the other kind of address, not both simultaneously.

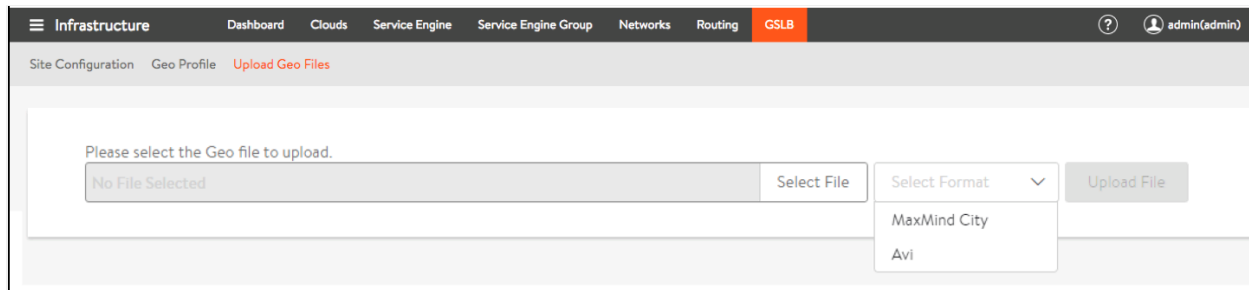
Impact on Avi Controller and Avi SE Configuration

To support the geolocation algorithm, special consideration must be given when configuring Controllers and the SEs implementing Avi DNS for GSLB.

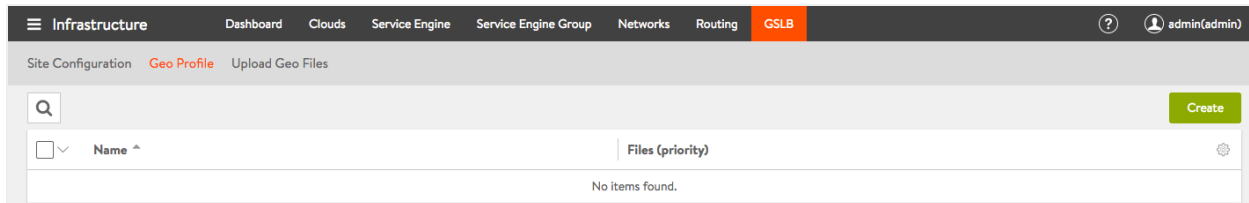
```
<li><strong>Memory Allocation</strong>. An Avi geo-DB can be very large (e.g., can easily contain 3 million entries).</li>
<a href="img/gslb.add-memory-to-accommodate-geo-db.png"></a>
<li><strong>Disk Allocation</strong>. An Avi geo-DB can be very large (e.g., can easily contain 3 million entries).</li>
<li>When the Avi geo-DB is large, the DNS Service Engine needs some time to load the entries into memory.</li>
<li>The CLI commands for these are discussed in the CLI section.</li>
```

Geo-DB GUI

Use the Upload Geo Files tab to select files that will become candidates for inclusion into the required geolocation profile. Note the format choices available in the pulldown.



To create the geolocation profile, navigate to Infrastructure > GSLB > Geo Profile as shown below.



Click Create to open the geo profile editor. * Give the profile a name. * Specify one or more previously uploaded files -- see below. For each files, specify a priority from 1 (lowest) to 100 (highest) and choose the file's format from the dropdown of supported formats. When determining the location of a given address or address range, Avi Vantage will search the files starting with the highest priority file, and pick the entry from the highest priority file.

New Geo Profile: ✕

Name*

Files

Filename ⓘ	Priority ⓘ	Format ⓘ	
Select File ▼	Priority	Avi ▼	🗑️

+ Add File

Save

CLI

When setting SE group properties in an environment that will use the geolocation algorithm, pay special att

```
[admin:10-10-24-207]: > configure serviceenginegroup mySEgrp
[admin:10-10-24-207]: serviceenginegroup> where
Tenant: admin
+-----+-----+
| Field | Value |
+-----+-----+
| name  | mySEgrp |
```

```
+-----+-----+
[admin:10-10-24-207]: serviceenginegroup> memory_per_se 8196
[admin:10-10-24-207]: serviceenginegroup> extra_shared_config_memory 2048
[admin:10-10-24-207]: serviceenginegroup> save
[admin:kh-c1]: serviceenginegroup>
```

Upload a geodb file(s)

```
[admin:10-10-24-207]: > upload gslbfiles [FILENAME]
```

Create a GSLB geolocation profile in vi editor-like fashion.

```
[admin:10-10-24-207]: > configure gslbgeodbprofile mygslbprofile
[admin:10-10-24-207]: gslbgeodbprofile> where
Tenant: admin
+-----+-----+
| Field | Value      |
+-----+-----+
| name  | mygslbprofile |
+-----+-----+
[admin:10-10-24-207]: gslbgeodbprofile> [TAB][TAB]
cancel      Exit the current submode without saving
description Help string not found for argument
do          Execute a show command
entries     (submode)
name       A user-friendly name for the geodb profile.
new        (Editor Mode) Create new object in editor mode
no         Remove field
save       Save and exit the current submode
show_schema show object schema
tenant_ref Help string not found for argument
watch      Watch a given show command
where      Display the in-progress object
[admin:10-10-24-207]: gslbgeodbprofile> where
Tenant: admin
+-----+-----+
| Field | Value      |
+-----+-----+
| name  | mygslbprofile |
+-----+-----+
[admin:10-10-24-207]: gslbgeodbprofile> new
# description: '<string> # Field Type: Optional'
# entries:
```

```
# - file:
#   checksum: '<string> # Field Type: Optional'
#   file_id: '<string> # Field Type: Optional'
#   filename: '<string> # Field Type: Optional'
#   format: '<choices: GSLB_GEODB_FILE_FORMAT_MAXMIND_CITY | GSLB_GEODB_FILE_FORMAT_AVI>
#     # Field Type: Optional'
#   timestamp: '<integer> # Field Type: Optional'
#   priority: '<integer> # Field Type: Optional'
#   name: '<string> # Field Type: Required'
#   tenant_uuid: '<string> # Field Type: Required'
#   uuid: '<string> # Field Type: Required'
#
name: mygslbprofile
```

Choose the geolocation load balancing algorithm for the GSLB service pool.

```
[admin:10-10-24-207]: > configure gslbservice gs-1
[admin:10-10-24-207]: gslbservice> new
# - algorithm: '<choices: GSLB_ALGORITHM_CONSISTENT_HASH | GSLB_ALGORITHM_ROUND_ROBIN
#   | GSLB_ALGORITHM_GEO> # Field Type: Optional'
#   consistent_hash_mask: '<integer> # Field Type: Optional'
#   members:
#     - cluster_uuid: '<string> # Field Type: Optional'
#       enabled: '(true | false) # Field Type: Optional'
#       fqdn: '<string> # Field Type: Optional'
#       hm_proxies:
#         - proxy_type: '<choices: GSLB_HEALTH_MONITOR_PROXY_ALL_MEMBERS> # Field Type:
#           Optional'
#           site_uuid: '<string> # Field Type: Optional'
#       ip:
#         addr: '<string> # Field Type: Required'
#         type: '<choices: V4 | DNS> # Field Type: Required'
#       location:
#         location:
#           latitude: '<float> # Field Type: Optional'
#           longitude: '<float> # Field Type: Optional'
#           name: '<string> # Field Type: Optional'
#           tag: '<string> # Field Type: Optional'
#           source: '<choices: GSLB_LOCATION_SRC_FROM_GEODB | GSLB_LOCATION_SRC_USER_CONFIGURED
#             | GSLB_LOCATION_SRC_INHERIT_FROM_SITE> # Field Type: Optional'
#           ratio: '<integer> # Field Type: Optional'
#           vs_uuid: '<string> # Field Type: Optional'
#       name: '<string> # Field Type: Optional'
#       priority: '<integer> # Field Type: Optional'
#   health_monitor_scope: '<choices: GSLB_SERVICE_HEALTH_MONITOR_ONLY_NON_AVI_MEMBERS
#     | GSLB_SERVICE_HEALTH_MONITOR_ALL_MEMBERS> # Field Type: Optional'
#   health_monitor_uuids: <string>
```



```

# name: '<string> # Field Type: Required'
# num_dns_ip: '<integer> # Field Type: Optional'
# tenant_uuid: '<string> # Field Type: Required'
# ttl: '<integer> # Field Type: Optional'
# use_edns_client_subnet: '(true | false) # Field Type: Optional'
# uuid: '<string> # Field Type: Required'
# wildcard_match: '(true | false) # Field Type: Optional'
#
groups:
- algorithm: GSLB_ALGORITHM_GEO
name: gs-1

```

A GSLB service using the geolocation algorithm will show the location (and any other values that might have

```

[admin:10-10-24-207]: > show gslbservice gs-1
-----+
Field  Value
-----+
uuid   gslbservice-90a4becd-0051-48b2-b19d-e3e2aa30f101
name   gs-1
domain_names[1] abcd.com
groups[1]
name   gs11
priority 10
algorithm GSLB_ALGORITHM_GEO
members[1]
ip 1.0.0.0
ratio 1
enabled True
controller_health_status_enabled True
health_monitor_scope GSLB_SERVICE_HEALTH_MONITOR_ALL_MEMBERS
enabled True
tenant_ref admin
-----+

[admin:10-10-24-207]: > show gslbservice gs-1 runtime
-----+
Field  Value
-----+
uuid   gslbservice-90a4becd-0051-48b2-b19d-e3e2aa30f101
name   gs-1
groups[1] | |
name   gs11
members[1]
ip 1.0.0.0
oper_ips[1] 1.0.0.0

```

Troubleshooting

```
<li>To verify correct operation from the perspective of clients, check Avi Vantage's logs.</li>  
<li>If a DNS SE does not have sufficient memory to host the geo-DB, the SE event SE_GEO_DB_FAILURE is generated
```

Other Related Articles

```
<li><a href="https://avinetworks.com/docs/17.1//avi-gslb-overview/">Avi GSLB Overview</a></li>  
<li><a href="https://avinetworks.com/docs/17.1//gslb-architecture-and-object-model/">Avi GSLB Architecture and  
<li><a href="https://avinetworks.com/docs/17.1//avi-gslb-site-configuration-and-operations/">Avi GSLB Site Con  
<li><a href="https://avinetworks.com/docs/17.1//avi-gslb-service-and-health-monitors/">Avi GSLB Service and He
```