Graph Engine Service

FAQs

Issue   01
Date   2023-06-02

Contents

1 Customer Consultation

1.1 What Is GES?

Graph Engine Service (GES) facilitates query and analysis of multi-relational graph data structures. It provides distributed, large-scale, integrated query and analysis capabilities and built-in extensive graph analysis algorithms to meet the application requirements in multiple domains.

- Graph size

  GES supports super-large graphs, with tens of billions of vertices and hundreds of billions of edges.

- Performance

  With three core patents, the high-performance kernel can respond to high-concurrency, multi-hop, and real-time queries in seconds.

- Application scenarios

  GES is widely applied to social networking, targeted marketing, credit insurance, and network and path design scenarios.

- Relationship with Eywa

  Eywa is a self-developed graph engine kernel, and GES is a cloud service based on Eywa.

1.2 What Are the Differences Between GES and Image Processing Services?

- GES

  GES is mainly used for relationship analysis. It abstracts a relationship network as a data graph. For example, assume if person is a single vertex, the relationship between two persons is an edge. GES analyzes the graph and explores potential relationships. GES has extensive built-in algorithm libraries and applies to social networking, precision marketing, credit insurance, and network and path design.

- Image processing

Image processing services are used to process images. For example, image recognition is one of the functions of image processing.

1.3 What Algorithms Can Be Used on GES?

To meet application requirements of various scenarios, GES provides extensive basic graph algorithms, graph mining algorithms, and graph metrics algorithms.

You can use algorithms to analyze graphs on the query editor page. The following table lists the supported algorithms.

Table 1-1 Algorithm list

Algorithm	Description
PageRank	PageRank, also called web page rank, is a hyperlink analysis algorithm used to rank websites in their search engine results and display the relevance and importance of web pages (nodes).
PersonalRank	PersonalRank is also called Personalized PageRank. It inherits the idea of the classic PageRank algorithm and uses the graph link structure to recursively calculate the importance of each node. Unlike the PageRank algorithm, to ensure that the access probability of each node in the random walk can reflect user preferences, the PersonalRank algorithm returns each hop to the source node at a (1-alpha) probability during random walk. Therefore, the relevance and importance of network nodes can be calculated based on the source node (the higher the PersonalRank value, the higher the correlation/importance of the source node).
K-core	K-core is a classic graph algorithm used to calculate the number of cores of each node. The calculation result is one of the most commonly used reference values for determining the importance of a node so that the propagation capability of the node can be better understood.
K-hop	This algorithm is used to search all nodes in the k layer that are associated with the start node through Breadth-First Search (BFS). The found sub-graphs are the start node's ego-net. The K-hop algorithm will return the number of nodes in the ego-net.
Shortest Paths	This algorithm is used to find the shortest path between two vertices in a graph.
All Shortest Paths	This algorithm is used to find all shortest paths between two vertices in a graph.
Filtered Shortest Path	This algorithm searches for the shortest path that meets the filter criteria between vertices. If there are multiple shortest paths, any one of them is returned.

Algorithm	Description
SSSP	This algorithm finds the shortest paths from a specified node (source) to other nodes.
Shortest Path of Vertex Sets	This algorithm discovers the shortest path between two vertex sets. It can be used to analyze the relationships between blocks in scenarios such as Internet social networking, financial risk control, road network transportation, and logistics delivery.
n-Paths	This algorithm is used to find the n paths between two vertices on the k layer of a graph. It applies to scenarios such as relationship analysis, path design, and network planning.
Closeness Centrality	Closeness centrality is the average distance from a node to all other reachable nodes. It can be used to measure the time for transmitting information from this node to other nodes. The smaller the node's closeness centrality is, the more central the node location will be.
Label Propagation	This algorithm is a graph-based semi-supervised learning method. Its basic principle is to predict the label information about unlabeled nodes using that of the labeled nodes. This algorithm can create graphs based on the relationships between samples. Nodes include labeled data and unlabeled data, and the edge indicates the similarity between two nodes. Node labels are transferred to other nodes based on the similarity. Labeled data is like a source used to label unlabeled data. The greater the node similarity is, the easier the label propagation will be.
Louvain	This algorithm is a modularity-based community discovery algorithm with high efficiency and effect. It can discover hierarchical community structures and aims to maximize the modularity of the entire community network.
Link Prediction	This algorithm is used to calculate the similarity between two nodes and predict their relationship based on the Jaccard measurement method.
Node2vec	By invoking the Word2vec algorithm, the Node2vec algorithm maps nodes in the network to the Euclidean space, and uses vectors to represent the node characteristics. The Node2vec algorithm generates random steps from each node using the rollback parameter **P** and forward parameter **Q**. It combines BFS and DFS. The rollback probability is proportional to 1/P, and the forward probability is proportional to 1/Q. Multiple random steps are generated to reflect the network structures.
Real-time Recommendation	This algorithm is based on the random walk model and is used to recommend nodes that are similar (have similar relationships or preferences) to the input node. This algorithm can be used to recommend similar products based on historical purchasing or browsing data or recommend potential friends with similar preferences.

Algorithm	Description
Common Neighbors	Common Neighbors is a common basic graph analysis algorithm. It can be used to obtain the neighboring nodes shared by two nodes and further speculate the potential relationship and similarity between the two nodes. For example, it can intuitively discover common friends in social occasions, or commodities that interest both nodes in the consumption field.
Connected Component	A connected component stands for a sub-graph, in which all nodes are connected with each other. Path directions are involved in the strongly connected components and are not considered in the weakly connected components. **NOTE** This algorithm generates weakly connected components.
Degree Correlation	This algorithm calculates the Pearson correlation coefficient between the start vertex degree and the end vertex degree of each edge. It is used to indicate whether the high-degree nodes are connected to other high-degree nodes in a graph.
Triangle Count	This algorithm counts the number of triangles in a graph without considering the edge directions. More triangles mean higher node association degrees and closer organization relationships.
Cluster Coefficient	The cluster coefficient describes the node aggregation degree in a graph. In real or some specific networks, nodes with high cluster coefficient always tend to form close organizational relationships.
Common Neighbors of Vertex Sets	This algorithm obtains vertex set neighbors, that are, the intersection of two vertex sets (groups). They are objects that are associated with both sets, for example, common friends, common products of interest, and persons contacting with both communities. You can use neighbors to further speculate potential relationships and the degree of the connection between two vertices.
All Shortest Paths of Vertex Sets	This algorithm is used to discover all shortest paths between two vertex sets. It can be used to analyze the relationships between blocks in scenarios such as social networking, financial risk control, road networks and transportation, and logistics delivery.
Filtered Circle Detection	This algorithm searches for all circles that meet a specified filter criteria in the graph. It is applicable to scenarios such as detecting round-trip transfers and anti-money laundering for financial risk control, locating abnormal links in network routes, and risk identification in enterprise finance guarantee.
Subgraph Matching	This algorithm is used to find all subgraphs of a given small graph that is isomorphic to a given large graph. This is a basic graph query operation and is intended to explore important substructures of a graph.

Algorithm	Description
Filtered All Pairs Shortest Paths	This algorithm is used to search for the shortest path between any two vertices in the graph that meets the condition. In a specific application scenario, you need to set a start vertex set (**sources**) and end vertex set (**targets**) as input for this algorithm. This algorithm returns the required shortest paths between the start and the end vertex sets.
Filtered All Shortest Paths	This algorithm allows you to search query results of the Shortest Path algorithm for the paths that meet the conditions between two vertices in a graph.
TopicRank	The TopicRank algorithm is one of commonly used algorithms for ranking topics by multiple dimensions. For example, this algorithm is applicable to rank complaint topics obtained through a government hotline.
Filtered n-Paths (2.2.22)	The filtered n-Paths algorithm is used to find no more than n k-hop loop-free paths between the source and target vertices. The start vertex (source), end vertex (target), number of hops (k), number of paths (n), and filter criteria (filters) are the parameters for the algorithm.

2 API Use

2.1 How Do I Import Data to GES?

If you did not import data when creating a graph, you will need to import data to the graph after creating it. In other scenarios, you will need to import incremental data to GES.

To learn how to import data incrementally, see **Importing Data**.

📖 **NOTE**

- Currently, only graphs of version 1.1.8 and later support this function.
- To prevent failures in restoring the imported graph data during system restart, do not delete the data stored on OBS when the graph is in use.
- The default separator of data columns is comma (,). You cannot define a separator.

2.2 When Binding an EIP to an Existing Instance or an Instance Being Created, What Should I Do to Deal with Insufficient Permissions for Creating Agencies?

Assign the security administrator role to the user group the user belongs to.

Step 1 Click **Service List** and choose **Identity and Access Management**.

Step 2 On the **Users** page, click ⌄ next to the username in the user list to view the user's user group name.

Step 3 On the **User Groups** page, click **Assign Permissions** in the **Operation** column of the user group. The **Group Permissions** page is displayed.

Step 4 Click **Attach Policy** in the **Operation** column of **Global service**. The **Attach Policy** page is displayed.

Step 5 In the **Available Policies** list on the left, select **Security Administrator** and click **OK**.

----End

2.3 When Binding an EIP to an Existing Instance or an Instance Being Created, What Should I Do to Deal with Insufficient Quotas for Creating Agencies?

You need to be authorized when using GES the first time. During the authorization process, an agency (for example **ges_admin_trust**) is created on the Identity and Access Management (IAM) page. You can create a maximum of 10 agencies. If you create more than 10 agencies, the system prompts that the agency quota is insufficient.

Perform the following steps to delete unnecessary agencies and implement authorization again.

Step 1 Log in to the Huawei Cloud official website or GES product page. In the drop-down list of the username in the upper right corner, click **My Account**. The **My Account** page is displayed.

Step 2 Select **Identity and Access Management** from the drop-down list under the username in the upper right corner. The IAM page is displayed.

Step 3 Choose **Agencies** in the left navigation pane. Then, delete any unnecessary agencies to ensure that GES has a sufficient quota for you to create an agency.

Step 4 After the agencies are deleted, perform authorization again and go to the GES page to create an instance.

----End

2.4 Can I Run Several Gremlin/Cypher Commands At a Time?

- You can run multiple Gremlin commands at a time. You need to separate the commands by semicolons (;).

  Example:
  ```
  graph = EywaGraph.open('ges_6715');g = graph.traversal();g.V().limit(1)
  ```
- However, you cannot run multiple Cypher commands at a time.

For details about the Cypher syntax, see **Cypher Query**. For details about the Gremlin syntax, see **Gremlin Query**.

2.5 If a Vertex Is Deleted, What Will Happen to Edges Formed Based on the Vertex?

GES imports graph data based on the property graph. A property graph is a directed graph consisting of vertices, edges, labels, and properties.

A vertex is also called a node, and an edge is also called a relationship. Nodes and relationships are the most important entities.

- Vertices in the data model represent entities, such as vehicles on the transportation grid, sites on a communications network, users and products on an e-commerce network, and web pages on the Internet.
- Edges in the data model represent relationships, such as friend relationships on a social network, user ratings and purchasing behavior on an e-commerce network, cooperative relationships between authors of a paper, and index relationships between articles.

If a vertex is deleted, edges formed based on the vertex are automatically deleted as well.

2.6 What Do I Do If an Error Message Indicating a Vertex Does Not Exist Is Displayed When Properties of the Vertex Whose ID Contains Chinese Characters Are Modified?

If such error message is displayed, perform the following operations:

1. Add **UTF-8** to **headers**: **"Content-Type":"application/json;charset=UTF-8"**
2. Use the URL encoding format to encode Chinese characters in the URL.

3 GUI

3.1 How Do I Execute a Gremlin Command and View the Execution Results?

- Executing a Gremlin command

On the graph editor page, enter a Gremlin command in the Gremlin text box in the lower part of the page and press **Enter** to analyze and query the current graph.

- Viewing the execution result

After running a Gremlin command, you can view the command running status in **Running Record** and the running result in **Query Result**. When querying vertices and edges, you can view the graphical display in the drawing area.

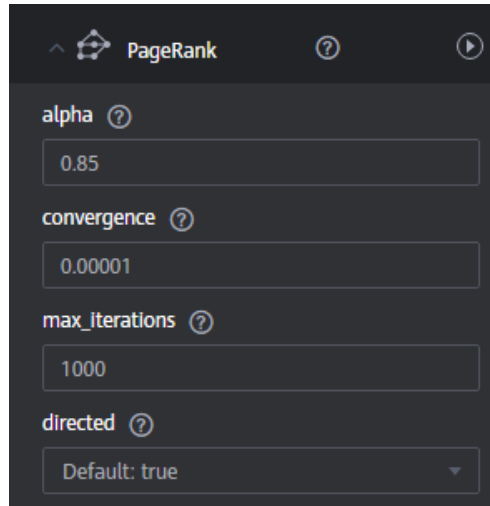**Figure 3-1** Executing a command and viewing the execution results
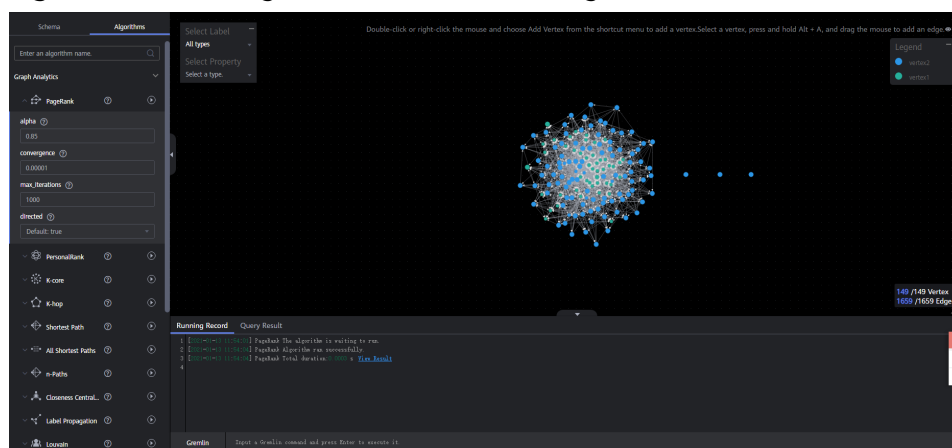
3.2 Why Do Some Algorithms Not Respond When I Click the Execution Button?

The execution button works after the algorithm parameters are set.

Click ⌄ on the left, set the required parameters, and click ⊙ to execute the algorithm.

Figure 3-2 Setting algorithm parameters

4 Others

4.1 What Are Regions and AZs?

Concepts

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are divided from the dimensions of geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.

- An AZ contains one or more physical data centers. Each AZ has independent cooling, fire extinguishing, moisture-proof, and electricity facilities. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.

Figure 4-1 shows the relationship between the regions and AZs.

Figure 4-1 Regions and AZs
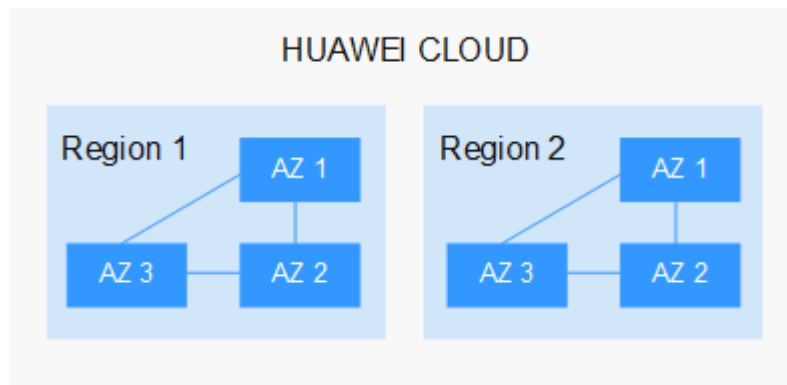
Services of HUAWEI CLOUD are available in many regions around the world. You can select a region and AZ as needed.

How to Select a Region

When selecting a region, consider the following factors:

- Location

  You are advised to select a region close to you or your target users. This reduces the network latency and improves the access speed. Regions within the Chinese mainland provide the same infrastructure, BGP network quality, as well as resource operations and configurations. Therefore, if your target users are in the Chinese mainland, you do not need to consider the network latency differences when selecting a region.

  - If you or your target users are in Asia Pacific excepting the Chinese mainland, select the **CN-Hong Kong**, **AP-Bangkok**, or **AP-Singapore** region.

  - If you or your target users are in Africa, select the **AF-Johannesburg** region.

  - If you or your target users are in Europe, select the **EU-Paris** region.

- Relationship between cloud services

  When using multiple cloud services, pay attention to the following restrictions:

  - ECSs, RDS instances, and OBS buckets in different regions cannot communicate with each other through an internal network.

  - ECSs in different regions cannot be bound to the same load balancer.

- Resource price

  Resource prices may vary in different regions.

How to Select an AZ

When determining whether to deploy resources in the same AZ, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs in the same region.

- For low network latency, deploy resources in the same AZ.

Regions and Endpoints

Before using an API to call resources, specify its region and endpoint. For more details about regions and endpoints, see **GES Regions and Endpoints**.