# LSGi Web Services

Tere Gonzalez, Janneth Rivera  
Aug 26, 2016

Contents

[LSGi Web Services 1](#_Toc461204216)

[Overview 1](#_Toc461204217)

[1. Architecture 1](#_Toc461204218)

[2. Datasets 2](#_Toc461204219)

[3. Web services 2](#_Toc461204220)

[3.1. Launch the inference 2](#_Toc461204221)

[3.2. Stop a job 3](#_Toc461204222)

[3.3. Inference Results by Iteration 3](#_Toc461204223)

[3.3.1. Statistics Only 3](#_Toc461204224)

[3.3.2. All Results 3](#_Toc461204225)

[3.3.3. By Vertex Id 4](#_Toc461204226)

[4. Console Configuration 4](#_Toc461204227)

[4.1.1. Get execution configuration 4](#_Toc461204228)

[4.1.2. Load specific configuration file 5](#_Toc461204229)

[4.1.3. Edit specific configuration file 5](#_Toc461204230)

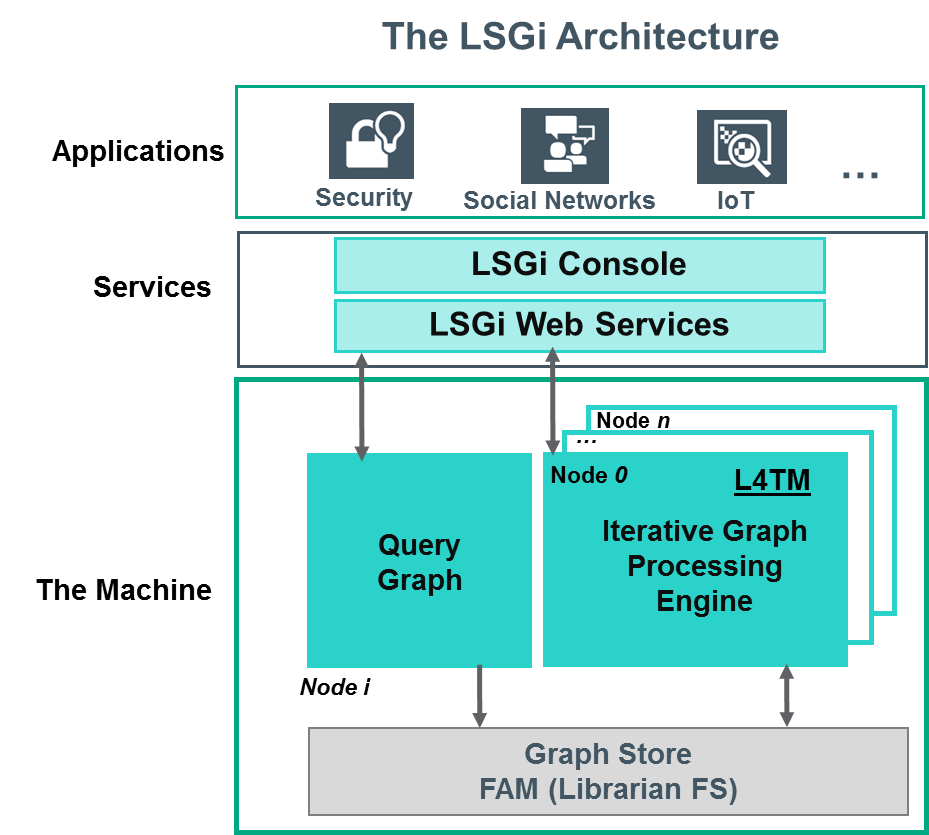
# Overview

This document describes the web services and data elements exposed to execute inference algorithms using Large Scale Graphical Inference (LSGi) engine as service across multiple deployed environments, datasets and computing nodes. The goal of this document is to provide the detail specification for developers that will use the web services.

# Architecture

The web services assume that the LSGi package has been already deployed and setup on the execution environment as is explained in LSGiDemoReadme.pdf document included in the package. Developers can build applications such malware detection, social interest, road accident detection, or brand reputation using the web services that enable the execution of LSGi engine on memory centric machines.

LSGi web services are consumed by LSGi Console as the figure shows. The services connect via SSH to the execution environments and launch inference jobs using the parameters selected (dataset/number of computing nodes) in the web console. The LSGi launch script starts N jobs and each node computes the inference for a K iterations. Then, LSGi Console can request the status of the results produced by the engine. The inference engine keeps only the latest iteration results on shared states. The web service retrieves iteration results by connecting to the graph query service (via TCP/IP) and retrieves them to the Console for visualization.



# Datasets

For this architecture, we assumed that datasets are stored already in the execution environments due to the datasets may be large and data transfer is expensive. In our design, the web service uses the relative path for the LSGi deployed package whi**ch should in**clude the datasets are stored either on the local folder, NFS or soft link that allows direct access to the file.

# Web services

# Launch the inference

Description:  
Launches Job and QueryService in specific environment. Verifies if "Done!" string was returned in order to notify success state to the UI. It also makes the assumption that if the environment to run is "localhost" (or 127.0.0.1) then a tunneling process is needed to access the real destination environment (for example, when accessing a virtual machine without public IP)

METHOD: POST

"/theMachine""/launch

Example:

INPUT:  
{environment: "build-l4tm-2.u.labs.hpecorp.net", lsgi\_home: "/home/gomariat/graphs/code/LSGi", nodes: "1", dataset: "/data/inputGraphs/dns\_graph.alchemy.factors.bin"}  
OUTPUT:  
{"status":"success"}

# Stop a job

Description:

Stop Job and QueryService in specific environment. Verifies if "Done!" string was returned in order to notify success state to the UI. It also makes the assumption that if the environment to run is "localhost" (or 127.0.0.1) then a tunneling process is needed to access the real destination environment (for example, when accessing a virtual machine without public IP).

METHOD: POST

"/theMachine""/stop

Example:

INPUT:  
{environment: "build-l4tm-2.u.labs.hpecorp.net", lsgi\_home: "/home/gomariat/graphs/code/LSGi"}  
OUTPUT:  
{"status":"success"}

# Inference Results by Iteration

# Statistics Only

Description:

Gets inference results - only statistics - by Probability Threshold at the last iteration run  
  
METHOD: POST

"/theMachine""/queryByProbThresholdStats

Example:

OUTPUT:  
{"iterationResults":{"stats":{"nonconverge":1,"iteration":1000,"time":224,"totalVertices":2077057,"state0":2057000,"convergence":93}}}

# All Results

Description:  
Gets inference results –vertices info and stats by Probability Threshold at the last iteration run

METHOD: POST

"/theMachine""/queryByProbThreshold

Example:

INPUT:  
{"environment": "build-l4tm-2.u.labs.hpecorp.net", "queryServicePort":58000, "probability":51}  
OUTPUT:  
{"iterationResults":{"topK":[{"vid":0,"p\_0":0.863,"state":"0","iteration":1000},{"vid":2,"p\_0":0.996,"state":"0","iteration":1000},{"vid":6,"p\_0":1.0,"state":"0","iteration":1000},{"vid":7,"p\_0":1.0,"state":"0","iteration":1000},{"vid":8,"p\_0":1.0,"state":"0","iteration":1000}],"stats":{"nonconverge":1,"iteration":1000,"time":224,"totalVertices":2077057,"state0":2057000,"convergence":93}}}

# By Vertex Id

Description:  
Gets inference results by VertexId at the last iteration run  
METHOD: POST

"/theMachine""/queryByVertexId

Example:

INPUT:  
{"environment": "build-l4tm-2.u.labs.hpecorp.net", "queryServicePort":58000, "vId":0}  
OUTPUT:  
{"iterationResults":{"vId":{"vid":0,"p\_0":0.863,"state":"0","iteration":1000},"stats":{"nonconverge":1,"iteration":1000,"time":224,"totalVertices":2077057,"state0":2057000,"convergence":93}}}

# Console Configuration

# Get execution configuration

Description:  
Get execution configurations for LSGi Console available in the configuration files. Please see web service configuration docs.  
METHOD: GET

"/theMachine""/getExecutionConfiguration"

Example:

OUTPUT:

{"nodes": [{"image": "Infrastructure.png","name": "1 node","value": "1"}], "environments": [{"image": "Servers.png","hostname": "build-l4tm-2.u.labs.hpecorp.net", "lsgi\_home": "/home/gomariat/graphs/code/LSGi","name": "l4tm-selfhosted-2"}], "datasets": [{"image": "Big\_data.png","loadingTime": 10000,"name": "1 Billion", "pollingTime": 500,"value": "/data/inputGraphs/1billion"}]}

# Load specific configuration file

Description:  
Loads a configuration file content.  
METHOD: POST

"/settings""/load"

Example:

INPUT:  
{"filename": "environments.json"}  
OUTPUT:  
{"environments": [{"id": 2,"name": "Single Node-L4TM","hostname": "build-l4tm-2.u.labs.hpecorp.net","image": "Servers.png","lsgi\_home": "/home/gomariat/graphs/code/LSGi","description": "The l4tm-selfhosted-2 is a single node (DL580) running L4TM and librarian self hosted version.","availableDatasets": [0,1,2,3,4],"availableNodes": [0,1,2,3,4,5,6]}]}

# Edit specific configuration file

Description:  
Saves changes in configuration file.  
METHOD: POST

"/settings""/save"

Example:

INPUT:  
{"filename": "environments.json", "content": [{"id": 1,"name": "Single Node-L4TM","hostname": "build-l4tm-2.u.labs.hpecorp.net","image": "Servers.png","lsgi\_home": "/home/gomariat/graphs/code/LSGi","description": "The l4tm-selfhosted-2 is a single node (DL580) running L4TM and librarian self hosted version.","availableDatasets": [0,1,2,3,4],"availableNodes": [0,1,2,3,4,5,6]}]}  
OUTPUT:  
{"status":"success"}