Eyevory - An Application Monitor
Project Requirements *&* Analysis Artifacts

CS 387 - Spring 2022

Group Members

Name	Roll Number	Contact
Ankit Kumar Misra	190050020	ankitkmisra@cse.iitb.ac.in
Richeek Das	190260036	richeek@cse.iitb.ac.in
Rahul Prajapat	190050095	rahulprajapat@cse.iitb.ac.in
Raja Gond	190050096	rajagond@cse.iitb.ac.in

Objective *&* Description

To build an application monitor for analyzing and visualizing data of different resource consumed on different server machines over time.

`How will user/client interact with our application monitor?`

The user/client will start the application, a dashboard will be open and there will be option available for choosing bunch of server IP addresses from the list for which user/client wish to analysis and visualise different resources consumption over time. Application monitor will get data from the chosen server machines, store and analysis using influxDb and show it user/client a nice graphical view. Our application application will issue warnings(i.e by showing some notifications) if any of the chosen machines will low on some resources(will be setting some threshold for this). There will be also query panel available to user/client for typing flux query which will give user/client some flexibility for visualization and detection of resource usage on the server machines.

`Why is a database, What kind of database and why this is suitable for our project?`

For application monitor, we require to work with large volumes of data. Databases are searchable and sortable, so the getting subset of data we need for analysis or prediction can be found quick and easily. Database are concurrent; multiple users can query over them at the same time without corrupting the data. We will use **InfluxDB** a popular *Time-Series Database* and it is very efficient at storing, querying, visualizing, and taking action on streams of time series data, events, and metric.

Why *Time-Series*? We will be tracking hard resources consumed (CPU, memory, Disk, Network bandwidth) with time and may also track soft resources consumed (DB connections, file descriptors, threads, DB cache etc). Data collected from these resources consumption will be over time intervals, giving us the ability to track changes over time(we are thinking to use **telegraf/Nagios** to collect data from 4-5 hosts). Time is going to be a important component of our data hence, the *Time-Series Database.*

Data Collection Diagram

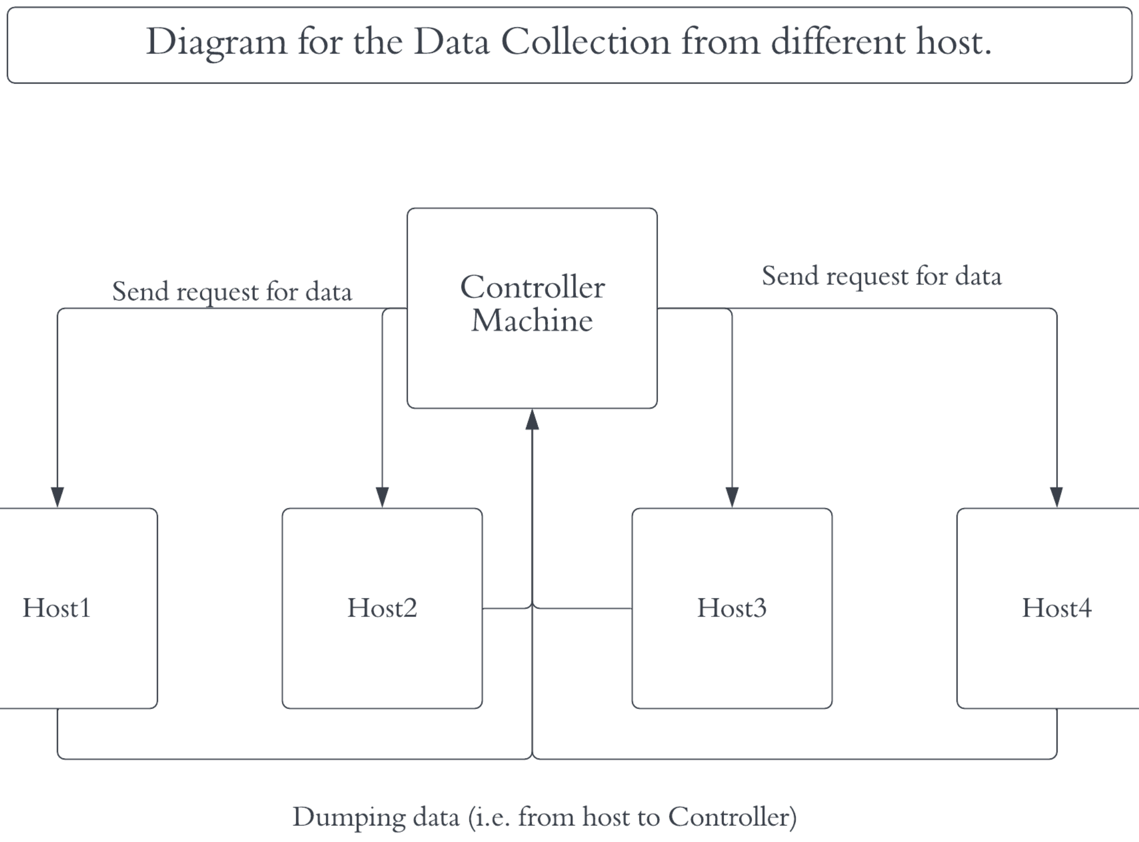
Diagram for the Data Collection from different host.
Send request for data
Controller Machine
Send request for data
Host1
Host2
Host3
Host4
Dumping data (i.e. from host to Controller)

Use cases

We list the possible use cases of our application monitor as below:

- The main use case of our application will be cluster analysis. We will have access to the complete picture of a group of servers (or machines). *Example:* Say we have 4 servers on 4 different machines, one running Django backend, one a Socket.IO server, one nginx and one a PostgreSQL server. We will run the application monitor from a controller machine and regularly query each of these 4 machines using their IP Addresses. We will take in the data from each of them and finally provide a unified view.
- It will provide failsafes. We will monitor the hard-resources on every host and allow end-users to set recommended levels on them. If any of the host servers are low on hard-resources we will issue a warning of possible crash/down-time soon. We will also predict crashes in the possible future along with the probable time of crash. *Example:* Host 1 with IP 172.14.5.xx will cross the recommended CPU bottleneck at 11:22PM UTC+05:30.
- It will also have a query panel, which will provide a low level view of the underlying Influx DB. In the query panel users will be able to directly write queries using Flux and get their desired results. It will have options to export the results into CSV and further visualize if it.

Attributes to track

- CPU
- Disk
- I/O
- Memory
- Network Connections
- Processes
- Database Connections

Dashboard and Start Screen

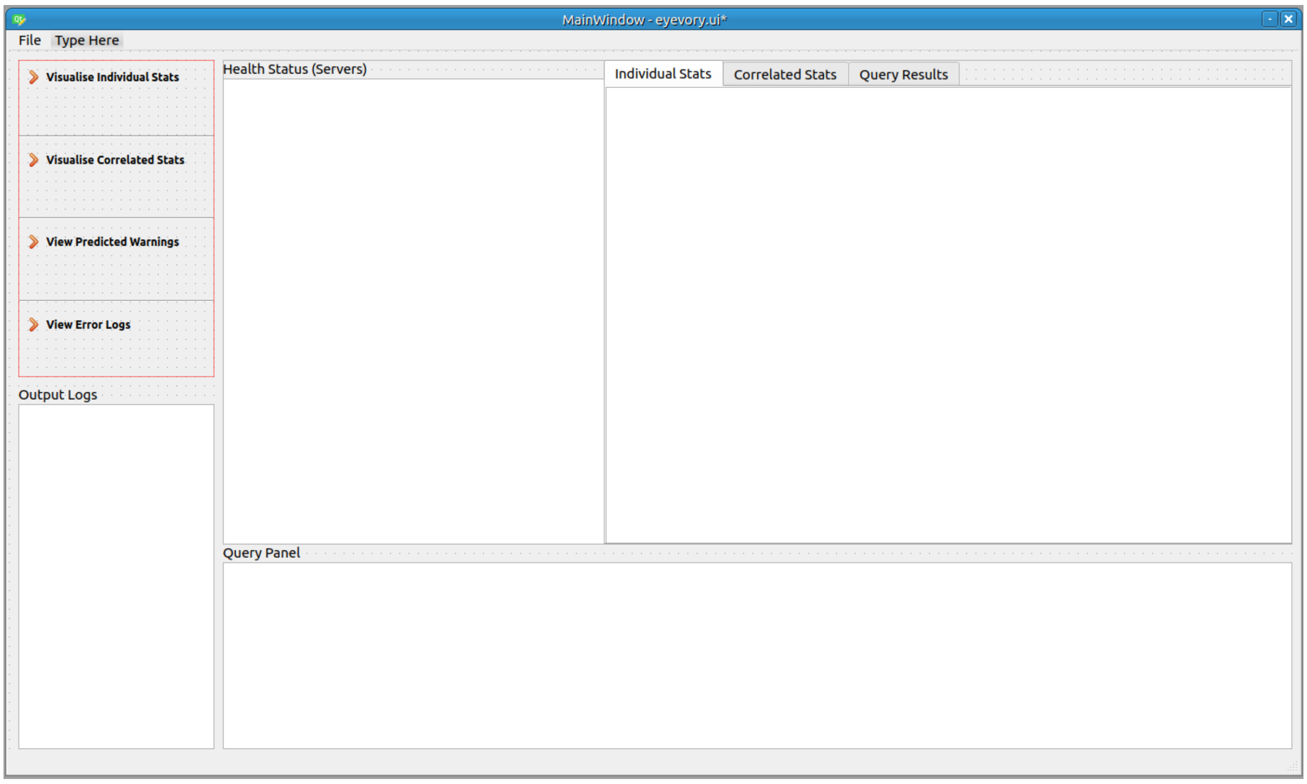

Figure 1: Eyevory Dashboard

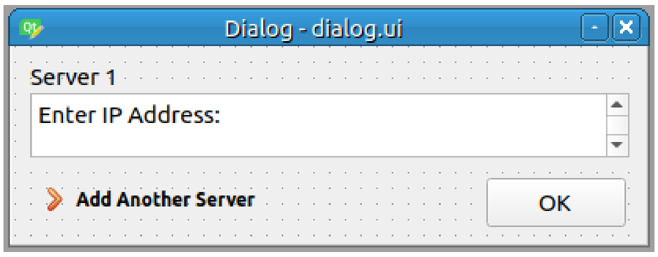

Figure 2: Start Screen

ER Diagram

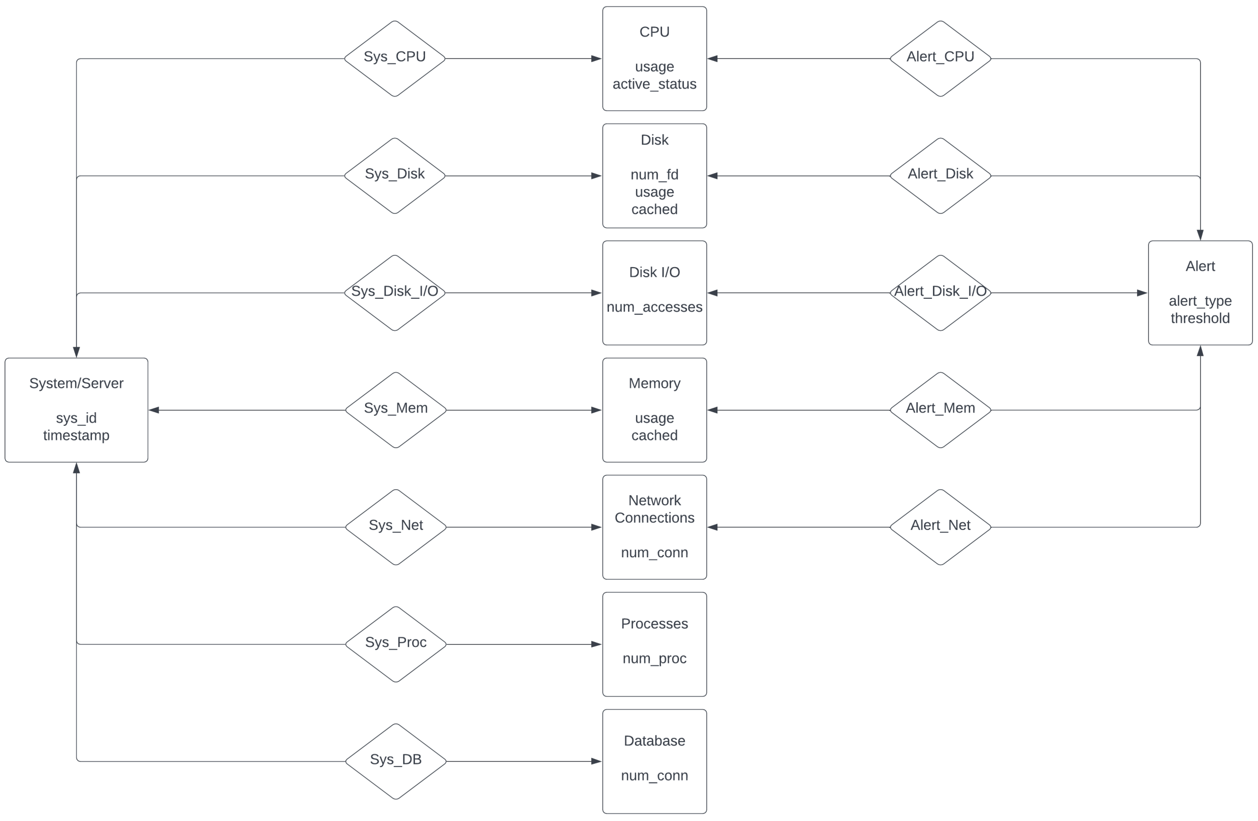

Figure 3: E-R Diagram