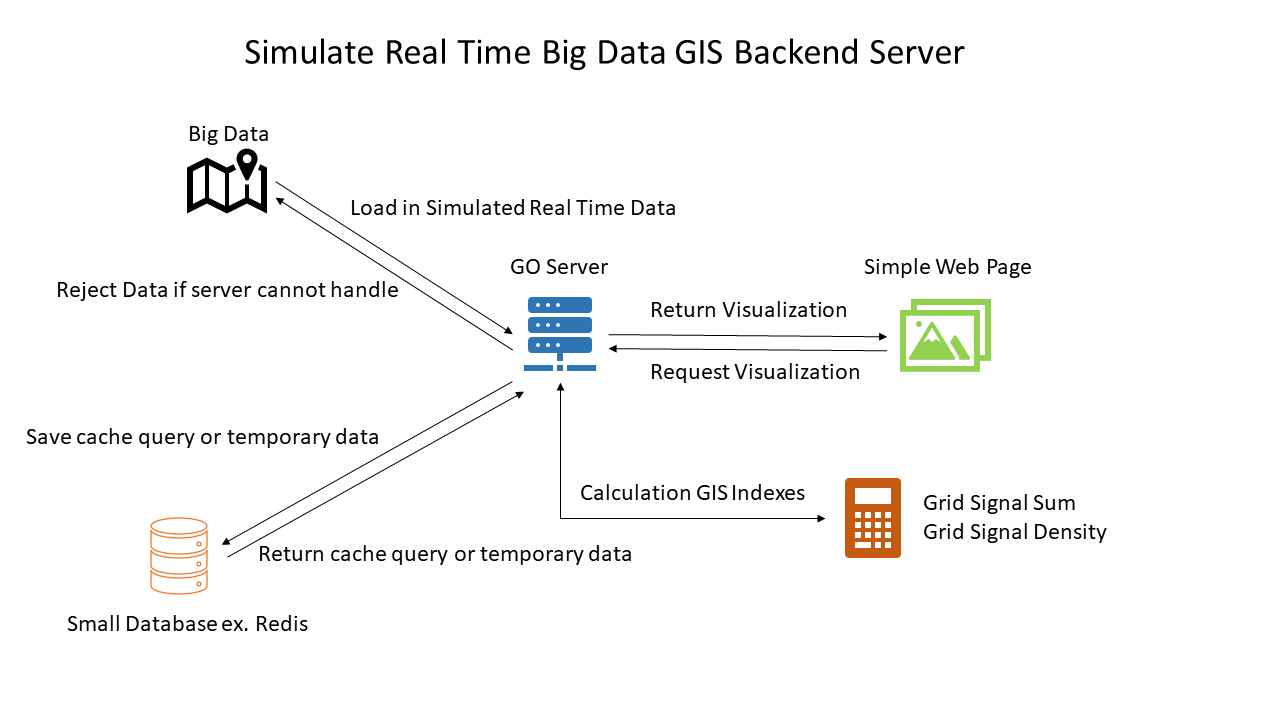
模擬巨量即時資料的GIS處理後端Server

1. Programming Language: GO / Python / Html / CSS / JavaScript

|  |  |
| --- | --- |
| Language | Purpose |
| GO | Backend Server |
| Python | Generate Fake Big Data |
| Html / CSS / JavaScript | Result Visualization |

1. Framework



1. 每筆資料均為json格式，包含user\_id / time / longitude / latitude
2. 模擬即時資料: 設計時間總長為10分鐘，每5秒鐘傳送一組資料集至Server進行處理，共120組資料集。每組資料集之資料總數均不相同，範圍為一萬至五萬之間，ex. 資料集1有25000筆，資料集2有15000筆。
3. 打算利用Redis來做為快取儲存，目前Server不傾向於連接mysql、postgresql等資料庫儲存所有資料。
4. GIS計算: 網格內的訊號點總量和密度，網格大小暫定為二公里

🡪 總量和密度超過某閥值發出警示

(想一下怎麼快速判斷位於哪個網格內)

1. 如果4完成後，可新增一些重點區域，如車站附近，這些區域的閥值可以自訂。
2. 最後於網頁上顯示網格地圖，以不同顏色區分不同密度。原本考慮兩種方案:
   1. Garfana
   2. Plotly + Dash (Python)

最後選定第二種因為免費而且考慮到可視化2維圖像的rows,cols不是太多，python處理起來不會太慢。Garfana免費版有metrics限制，若資料量過大時可能有影響。

1. GOAL: 於網頁顯示可即時更新的地圖
2. 時間安排

|  |  |  |
| --- | --- | --- |
| 項目 | 預計時間 | 預計日期 (2023) |
| 產生大量不同時間的假資料 | 1 - 2天 | 4/4 – 4/5 |
| GO 計算GIS Index | 1 - 2天 | 4/6 – 4/7 |
| GO 連接Redis | 1 - 2天 | 4/8 – 4/9 |
| 建置簡易網頁 | 1 - 2天 | 4/10 – 4/11 |
| 整合GO server | 3 - 5天 | 4/12 – 4/16 |
| 總開發時間 | 7 - 13天 | 4/4 – 4/17 |

實際開發時間

|  |  |  |
| --- | --- | --- |
| 項目 | 時間 | 日期 (2023) |
| 產生大量不同時間的假資料 | 1天 | 4/4 – 4/4 |
| GO 計算GIS Index | 2天 | 4/6 – 4/8 |
| GO 連接Redis | 2天 | 4/12 – 4/13 |
| 建置簡易網頁 | 3天 | 4/17 – 4/20 |
| 整合GO server | 1天 | 4/21 – 4/21 |
| 總開發時間 | 9天 | 4/4 – 4/21 |

