### 一、测试环境

|  |  |
| --- | --- |
| 硬件环境 | CPU: Intel Core i5-6500  内存：12G |
| 软件环境 | 系统： Window 10  TopJS 3.3.1  PostgreSQL 10.12 |

### 二、测试方式

通过JS脚本批量插入数据到本地PG数据库，主要通过TopJS数据库接口batchInsert()批量插入数据及COPY命令从文件插入数据2种方式，脚本由TopJS执行，采用多进程同时执行多个脚本的方式测试数据插入效率。

测试表为oee\_machine\_log\_v2，表结构如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 参考值 | 备注 |
| id | int8 | 200320100000000004 | 非空，自增 |
| workcenter\_id | int4 | 25 | 非空 |
| log\_time | timestamp | 2020-03-19 13:15:33 | 非空 |
| log\_time2 | timestamp | 2020-03-19 13:15:33 |  |
| log\_type | varchar | info |  |
| lot\_no | varchar | 123456 |  |
| partnumber | varchar | abc123 |  |
| subpart | varchar | test\_part |  |
| lot\_serial | int4 | 1122 |  |
| station | varchar | test\_station |  |
| state | varchar | on |  |
| programe\_name | varchar | test\_program |  |
| daq\_time | timestamp | 2020-03-20 14:16:45 |  |
| analysis\_flag | bool | t |  |
| log\_data | jsonb | {“a”: “b”,”c”: 1} |  |

注意：测试数据由脚本随机生成，其中log\_data为包含100个键值对的Map构成。

脚本在gitlab仓库：<http://gitlab.topibd.net/clownce.deng/pg-batch-insert-test>

### 测试结果

记录各进程执行数据插入过程所用时间（秒）。

##### 1. batchInsert方式

a ) 单进程

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据量（行） | 10 | 50 | 100 | 500 | 1000 | 2000 | 5000 |
| 耗时（秒） | 0.107 | 0.292 | 0.518 | 2.297 | 4.425 | 8.622 | 21.159 |

b ) 5进程

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据量（行）  进程  耗时（秒） | 10 | 50 | 100 | 500 | 1000 | 2000 | 5000 |
| 进程1 | 0.117 | 0.39 | 0.681 | 3.11 | 5.891 | 11.833 | 36.228 |
| 进程2 | 0.112 | 0.347 | 0.692 | 3.112 | 6.06 | 11.952 | 35.811 |
| 进程3 | 0.125 | 0.355 | 0.69 | 3.059 | 6.234 | 12.54 | 35.867 |
| 进程4 | 0.115 | 0.366 | 0.702 | 3.137 | 9.728 | 19.672 | 37.783 |
| 进程5 | 0.12 | 0.296 | 0.537 | 5.274 | 6.034 | 11.847 | 36.192 |
| **平均** | **0.118** | **0.351** | **0.66** | **3.538** | **6.789** | **13.567** | **36.376** |

c ) log\_data字段类型为json时的效率（json写入更快，jsonb读取更快）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据量（行）  进程  耗时（秒） | 10 | 50 | 100 | 500 | 1000 | 2000 | 5000 |
| 进程1 | 0.136 | 0.336 | 0.676 | 3.085 | 5.914 | 11.988 | 32.646 |
| 进程2 | 0.12 | 0.358 | 0.66 | 3.031 | 6.169 | 19.935 | 32.645 |
| 进程3 | 0.109 | 0.368 | 0.701 | 3.037 | 6.061 | 11.989 | 33.297 |
| 进程4 | 0.128 | 0.388 | 0.705 | 3.072 | 5.972 | 12.118 | 33.643 |
| 进程5 | 0.121 | 0.284 | 0.539 | 5.162 | 9.541 | 12.021 | 39.535 |
| **平均** | **0.123** | **0.347** | **0.656** | **3.477** | **6.731** | **13.61** | **34.353** |

##### COPY命令方式

COPY命令是从文件读取数据并写入数据库，这里以CSV文件为例。

a ) 单进程

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据量（行） | 10 | 50 | 100 | 500 | 1000 | 2000 | 5000 |
| 耗时（秒） | 0.068 | 0.077 | 0.078 | 0.124 | 0.187 | 0.315 | 0.698 |

b ) 5进程

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据量（行）  进程  耗时（秒） | 10 | 50 | 100 | 500 | 1000 | 2000 | 5000 |
| 进程1 | 0.064 | 0.087 | 0.078 | 0.151 | 0.216 | 0.39 | 1.213 |
| 进程2 | 0.072 | 0.115 | 0.073 | 0.171 | 0.217 | 0.388 | 1.192 |
| 进程3 | 0.082 | 0.089 | 0.082 | 0.226 | 0.209 | 0.524 | 1.276 |
| 进程4 | 0.07 | 0.07 | 0.083 | 0.148 | 0.223 | 0.438 | 1.239 |
| 进程5 | 0.065 | 0.07 | 0.103 | 0.149 | 0.215 | 0.428 | 1.163 |
| **平均** | **0.07** | **0.086** | **0.084** | **0.169** | **0.216** | **0.434** | **1.217** |

c ) log\_data字段为json时的效率（json写入更快，jsonb读取更快）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 数据量（行）  进程  耗时（秒） | 10 | 50 | 100 | 500 | 1000 | 2000 | 5000 |
| 进程1 | 0.08 | 0.072 | 0.078 | 0.124 | 0.254 | 0.331 | 1.031 |
| 进程2 | 0.07 | 0.083 | 0.079 | 0.114 | 0.161 | 0.348 | 1.018 |
| 进程3 | 0.073 | 0.068 | 0.073 | 0.186 | 0.171 | 0.345 | 1.043 |
| 进程4 | 0.069 | 0.071 | 0.078 | 0.124 | 0.165 | 0.319 | 0.996 |
| 进程5 | 0.067 | 0.171 | 0.075 | 0.155 | 0.273 | 0.349 | 1.026 |
| **平均** | **0.072** | **0.093** | **0.077** | **0.141** | **0.205** | **0.338** | **1.023** |