**Generate.py**

import time

start\_time = time.time()

with open('input.ssv', 'w') as out:

symbols = ['AUDUSD','EURUSD','GBPUSD','NZDUSD','USDCAD','USDCHF','USDJPY','USDCNY','USDHKD']

lines = []

for i in range(0,1\*1000\*1000):

q1, r1, q2, r2 = i//100000, i%100000, (i+1)//100000, (i+1)%100000

line = '{} {}.{:05d} {}.{:05d}'.format(symbols[i%len(symbols)], q1, r1, q2, r2)

lines.append(line)

out.write('\n'.join(lines))

print(time.time()-start\_time, i)

InfluxDBInsertSingleThread.py

import time

class Timer:

def \_\_enter\_\_(self):

self.start = time.time()

return self

def \_\_exit\_\_(self, \*args):

elapsed = time.time()-self.start

print('Imported in {:.2f} seconds or {:.0f} per second'.format(elapsed, 1\*1000\*1000/elapsed))

from influxdb import InfluxDBClient

client = InfluxDBClient('localhost', 8086, 'root', 'root', 'database01')

client.create\_database('database01')

with Timer() as t:

with open('input.ssv', 'r') as infile:

lines = infile.read().splitlines()

for line in lines:

json\_body = [

{

"measurement": "quotes",

"tags": {

"symbol": line[0:6]

},

"fields": {

"bid": float(line[7:14]),

"ask": float(line[15:])

}

}

]

client.write\_points(json\_body)

Imported in 5800.86 seconds or 172 per second