

Taming System with Python

A sysadmin perspective



Taming System with Python

- Motivation
- Decision
- GIL? How do we tame GIL?
- Concurrency: CSP Model
- A very brief tutorial about threading, process and queue
- Using concurrency in system administration
 - Case 1 (using Redis)
 - Case 2 (using Queue)
- Video Demo
- Conclusion
- Questions

Who am I

1. Lead system engineer for ROKKI Avionics Sdn. Bhd.
2. System and Network administrator for ROKKI Sdn. Bhd.
3. 2017 PyCon APAC committee member
4. 2018 PyCon MY committee member
5. Creator of TravelBot on LINE platform.



ROKKI

1. We are the in-flight entertainment and connectivity service provider to AirAsia aircraft.
2. Currently we have 50 + 1 (MAA + IAA) aircrafts installed with ROKKI system
3. PK-AXV is the first Indonesian AA aircraft installed with ROKKI system.
4. Python is used in ROKKI to handle some low level signaling between file server and satellite communication, as well as backend API for the captive portal.





Source: <https://www.jetphotos.com/photo/8768205>

Motivation

- In one boring, hot, sleepy and boring Monday afternoon....
- I decided to print a paper to read....
- But I cannot print....
 - Because the office had changed a new printer without my notice.

Motivation

Try to find the driver for the printer

[OpenPrinting](#)
[Database](#)
[Printers](#)
[Drivers](#)
[FAQ](#)
[Foomatic](#)
[Articles](#)
[Projects](#)
[Documentation](#)
[Developers](#)

[The Linux Foundation](#) · [OpenPrinting](#) · [Printers](#) · [KONICA MINOLTA](#) · [bizhub C360](#)

KONICA MINOLTA bizhub C360

Color laser printer, max. 1800x600 dpi, this is a **Paperweight**

User-contributed Printer Entry

This printer entry was contributed by a user but was not yet verified or proofread by the site administrators. Therefore it is not included in the [Foomatic](#) packages.

Discussion Forum


Look for help in our [forum for printers from Konica Minolta, Minolta, and QMS](#).

Comments

1 Comment [OpenPrinting.org](#)


News & Announcements

Latest Comments




[John Gilmour](#) drivers were located and installed by linux ubuntu 12 but no more tried the hplp, but the connection fails over and over i'm going to put my project on a thumb drive, and take it to a real print...

http://www.openprinting.org/printer/HP/HP-LaserJet_P1005 · 3 days ago



[M Mahender](#) i have hp laserjet printer P1007 and hp 1020plus. But I couldn't install in BOSS(LINUX) operating PC. I need help.



Motivation

Well...The outcomes can be quite passive:

1. You ask your colleague to print for you
2. You wait for community to submit a printer driver

With no resource in hand, I decided to choose 1. Option 2 might take years.

Motivation

Motivated by Jessica McKellar's Talk: *The programmer's Mind* during PyCon APAC 2017:

- *Programming changes the way you think about and debug and interact with the world*
- *Programmers master a system they know they can change*

Decision

- Why don't I write a printer driver?
 - PCL (Printer Command Language) vs Postscript...
- Do I have a simple solution?
 - Simple Server-Client model

Decision: Simple Server-Client Model

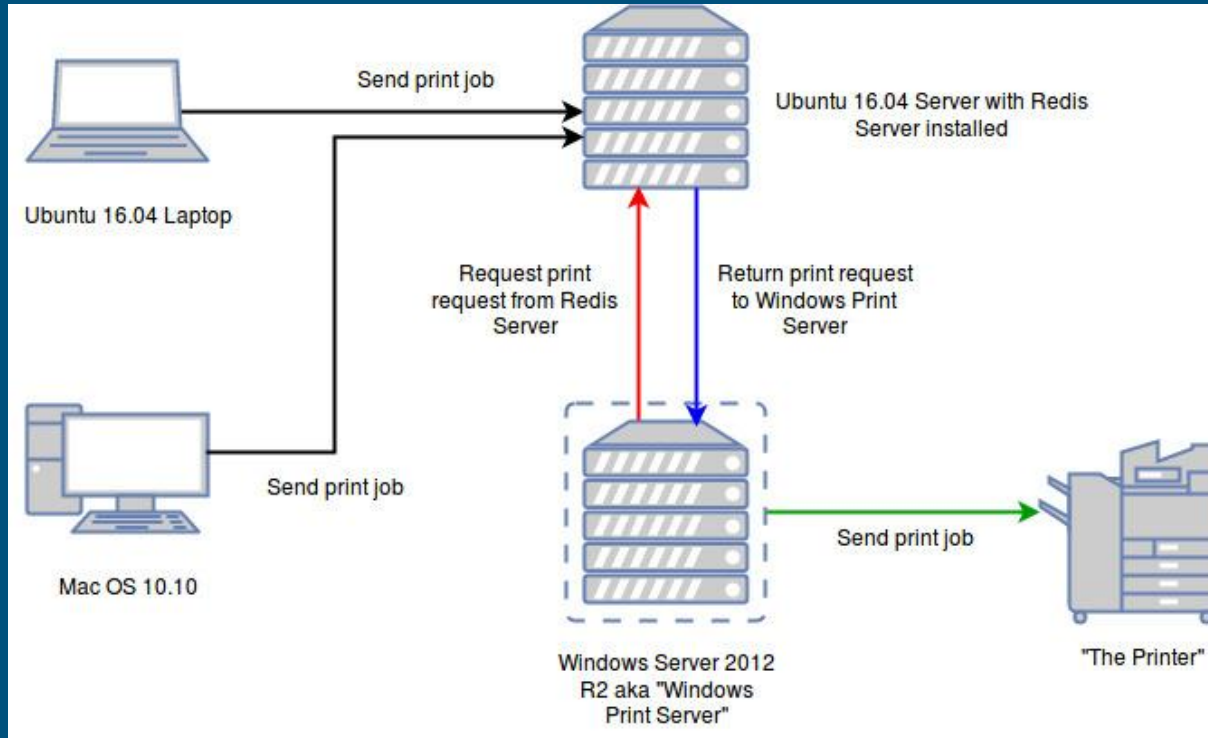
- Server:
 - A window 2012 RC2 VM with printer driver, pdf reader and python installed
- Client:
 - A python client
- Queue:
 - Redis server

Decision: Simple Server-Client Model

Version 1:

1. Client sends file metadata to Redis Server
2. Client copies the file to NFS directory
3. Windows mounts the NFS directory (YES, Windows Server 2012 has NFS client)
4. Server in Windows 2012 reads from Redis Server,
5. Server sends the print job to printer.

Decision: Simple Server-Client Model



Decision: Simple Server-Client Model

Reflection for Version 1:

1. This work around seems to be ok.
2. This is a multi single-threaded processes.
3. Setting up NFS mount point between Windows 2012 and GNU/Linux is not easy
4. Can we improve this?

Decision: Simple Server-Client Model

What can we improve?

1. Use HTTP POST to upload the file to WEB Server (flask APP)
2. Run SERVER to do the printing

Better version:

But can we run both in one application?

Concurrency?



Source: <https://www.nostarch.com/erlang>

Global Interpreter Lock

David M Beazley, “Understanding GIL”

The Unwritten Rules of Python

1. *You do not talk about the GIL.*
2. *You do NOT talk about the GIL.*
3. *Don't even mention the GIL. No seriously.*

Global Interpreter Lock

- What is GIL?
 - Global Interpreter Lock is a mutex lock to prevent multiple threads from executing python bytecodes at once.
 - This is due to the CPython's memory management which is not thread-safe.
- Conclusion: Multithreaded program may perform badly compared single-threaded program due to GIL.
- GIL seems to be taboo in python. But can we improve our solution using threading package?

Global Interpreter Lock: Taming GIL?

Few questions about GIL:

1. Does it mean that we cannot use threading or multiprocessing since GIL has constrained the behaviour of python bytecode?
2. Is there a way that we can tame GIL?
3. Can we eliminate GIL?

Global Interpreter Lock: Taming GIL?

Few questions about GIL:

1. Does it mean that we cannot use threading or multiprocessing since GIL has constrained the behaviour of python bytecode? (NO!)
2. Is there a way that we can tame GIL? (Depends on what you want to do)
3. Can we eliminate GIL? (Not at the moment)

Global Interpreter Lock: Taming GIL?

Third Law of Thermodynamics:

It is impossible for any process, no matter how idealized, to reduce the entropy of a system to its absolute-zero value in a finite number of operations.

In another word: There is no way for us to achieve absolute-zero.

Global Interpreter Lock: Taming GIL?

Physicists in MIT managed to cool atoms down to 0.0000000005 K

Cooling Bose-Einstein Condensates Below 500 Picokelvin

A. E. Leanhardt,* T. A. Pasquini, M. Saba, A. Schirotzek, Y. Shin,
D. Kielpinski, D. E. Pritchard, W. Ketterle

Spin-polarized gaseous Bose-Einstein condensates were confined by a combination of gravitational and magnetic forces. The partially condensed atomic vapors were adiabatically decompressed by weakening the gravito-magnetic trap to a mean frequency of 1 hertz, then evaporatively reduced in size to 2500 atoms. This lowered the peak condensate density to 5×10^{10} atoms per cubic centimeter and cooled the entire cloud in all three dimensions to a kinetic temperature of 450 ± 80 picokelvin. Such spin-polarized, dilute, and ultracold gases are important for spectroscopy, metrology, and atom optics.

Source: Science Vol 301, 12 Sept 2003

Global Interpreter Lock: Conclusion

My experience:

- Multithreading - parallelism or concurrency?
 - If you treat multithreading as parallelism, GIL is going to hold your back.
- But can we do many things at one time using threading package?

Concurrency: CSP model

- CSP model - Communicating Sequential Process
 - Concept: Let threads communicate through channels, not by memory.
- An inspired talk by Mosky's *Elegant Concurrency* at PyCon APAC 2017
- Rob Pike (Golang creator):

Do not communicate by sharing memory; instead, share memory by communicating.

- Do we have similar concept in python? (YES. We can use Queue)

A brief tutorial about threading

```
import time
import threading

def wait_print():
    thread_name = threading.currentThread().getName()
    print("%s reporting!\n" % thread_name)
    time.sleep(5)
    print("%s is inside Ecto-1\n" % thread_name)

def main():

    ghostbuster = [ "Peter Venkman", "Egon Spengler", "Raymond Stantz", "Winston Zeddemore"]
    threads = [threading.Thread(name=gb, target=wait_print, args=()) for gb in ghostbuster]

    for thread in threads:
        thread.start()

if __name__ == "__main__":
    main()
```


A brief tutorial about multiprocessing

```
import time
from multiprocessing import Process

def wait_print(process_name):
    print("%s reporting!\n" % process_name)
    time.sleep(5)
    print("%s is inside Ecto-1\n" % process_name)

def main():

    ghostbuster = [ "Peter Venkman", "Egon Spengler", "Raymond Stantz", "Winston Zeddemore"]
    processes = [Process(target=wait_print, args=(gb,)) for gb in ghostbuster]

    for process in processes:
        process.start()

if __name__ == "__main__":
    main()
```

A brief tutorial on Queue

- Python's queue is quite a generic data structure.
- It consists of FIFO, LIFO and Priority
- It can be used in anywhere.
- Interesting things:
 - multiprocessing also has Queue object, which actually uses Queue library

A brief tutorial on Queue

Code Demo:

- `queue_thread.py`
- `queue_mp.py`

Concurrency in System Administration

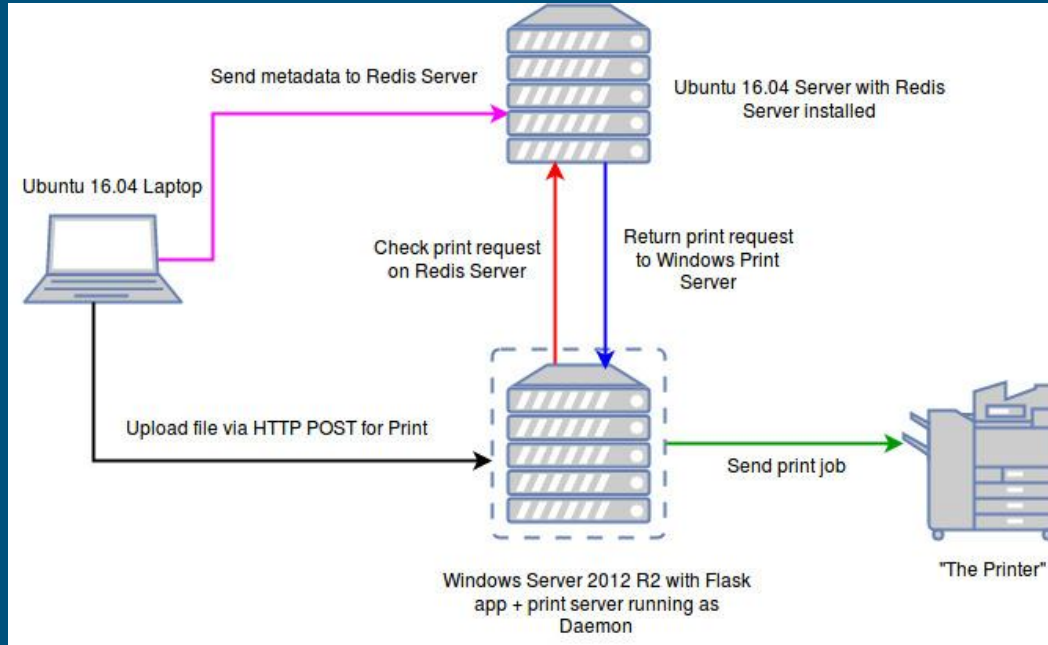
Case 1: (Use Redis Server)

The windows printing issue: How do we solve it using *threading*?

1. Client uploads file via HTTP POST
2. Client uploads file metadata to Redis Server
3. Server receives file via HTTP POST
4. Server checks file metadata on Redis Server
5. Server sends print jobs according to file metadata

Concurrency in System Administration

Case 1: (Use Redis Server)



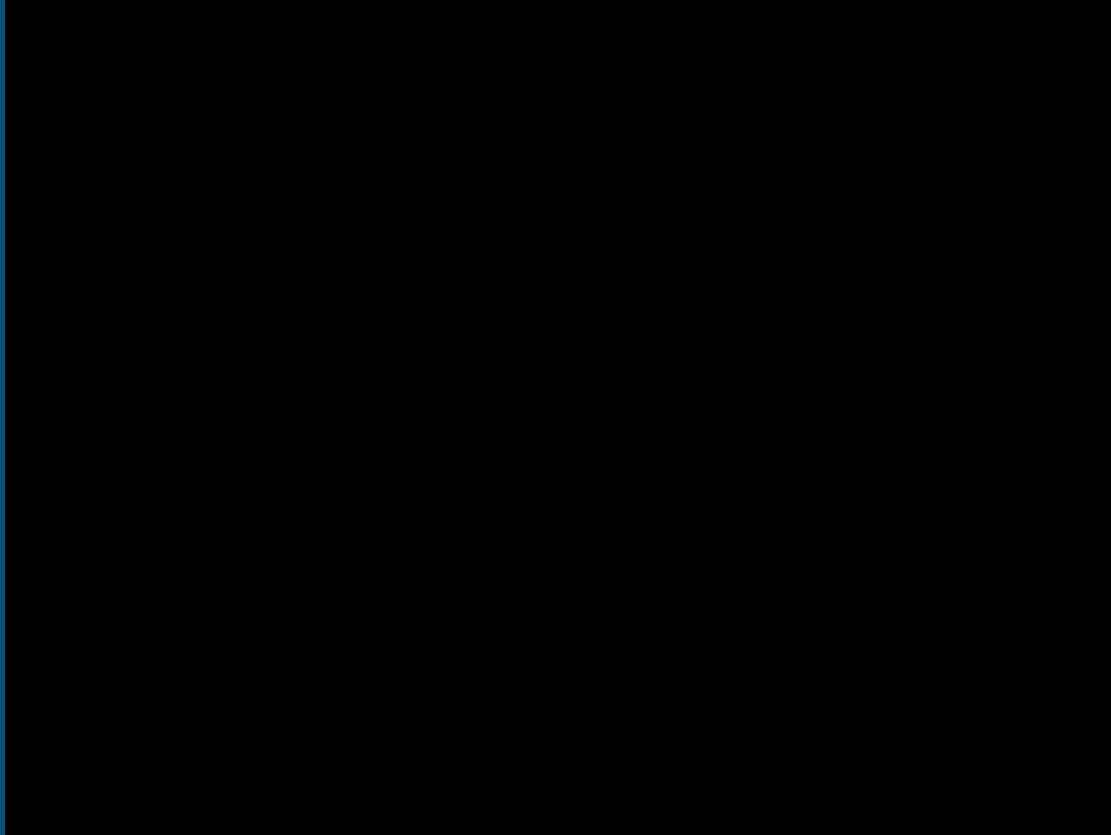
Concurrency in System Administration

Case 2: (Use Queue)

Monitoring your network: How do you determine the quality of your network?

1. Use pyspeedtest (<https://github.com/fopina/pyspeedtest>)
2. But it is slow, even it is multi-threaded.
3. Can we make it a little faster? (YES)
 - We can utilize multiprocessing and Queue

Video Demo & Demo



Conclusion

What have I learned?

- Threading is not as scary as it is.
 - It can be used as daemon
- Multiprocessing seems to be more promising than Threading.
 - Bad point: It requires large memory (expensive to use)
- Be a creative sysadmin
- Read more
 - Both books + github repo. You will learn others' creativity.
- Do not afraid to get your hands dirty.
- Do not worry if your github repo is starred.
 - The purpose of programming is to solve problems, even small problems

Note before end:

- Tutorial: (https://www.github.com/tangingw/pycon_id_tutorial.git)
- Python_winprint (https://www.github.com/tangingw/python_winprint.git)
- Python_network (https://www.github.com/tangingw/python_network.git)

Thank you!

Questions?
