CUPS Synchronizer

Contents

[Overview 1](#_Toc412631121)

[Scope 1](#_Toc412631122)

[Existing 1](#_Toc412631123)

[Proposed 2](#_Toc412631124)

[Problem 2](#_Toc412631125)

[Solution 2](#_Toc412631126)

[Limitations and TODO(s) 5](#_Toc412631127)

[Performance 5](#_Toc412631128)

[OS, Software, Tools & Technologies Used 8](#_Toc412631129)

# Overview

CUPS is a very commonly used Linux based printing system. CUPS can work on various printing protocols like IPP (Internet Printing Protocol). As of now we have only one instance of CUPS. Objective of this document is to provide an overall idea and implementation plan to provide high availability (failover & load balancing) of this printing system (CUPS).

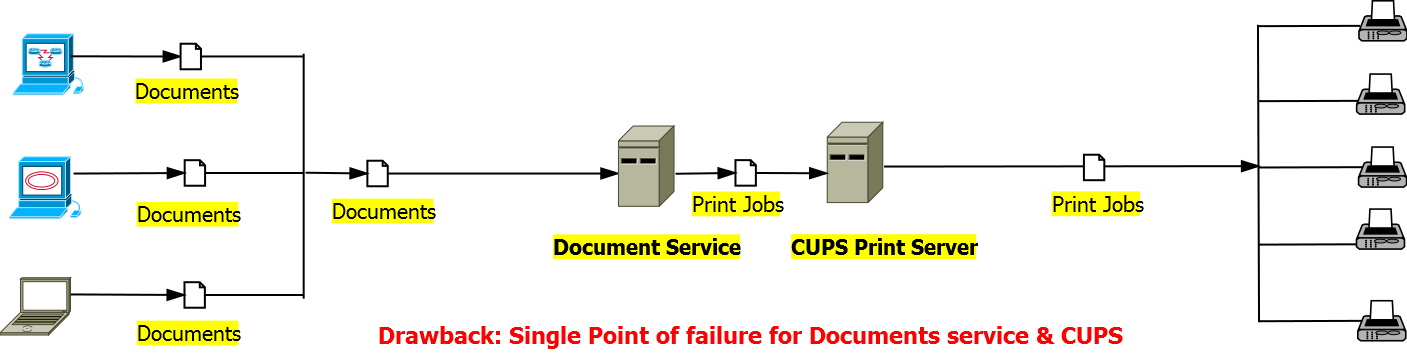
# Scope

* Scope of this document is to provide a brief overview of the problem and proposing a potential solution.
* Consider this is a POC document and the implementation plan may change deepening on the final decision/discussion.
* Actual implementation and details of the program/code is not in the scope of this document.

# Existing

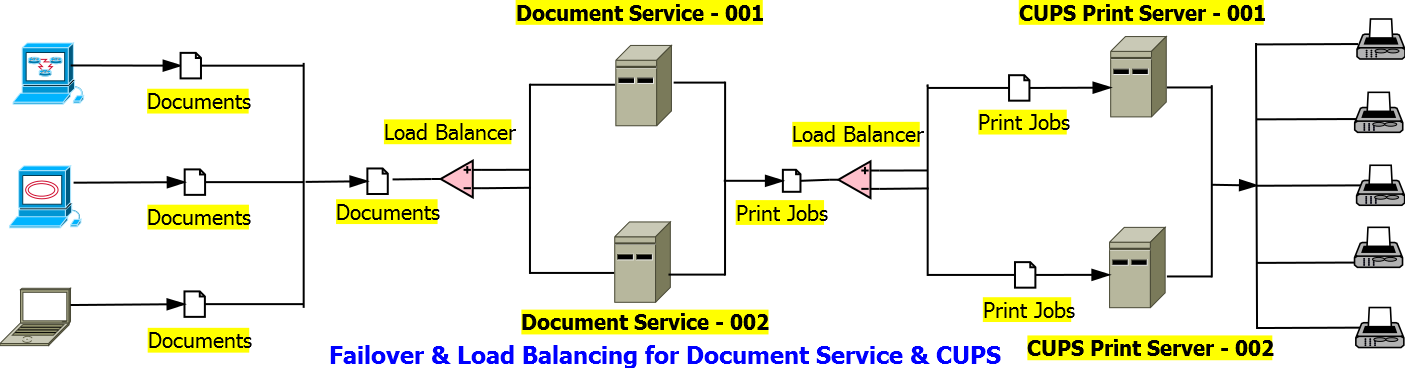
CUPS is a popular print server. Document Service connects to CUPS for sending print jobs. As of now we have only one instance of Document Service and CUPS. Hence our printing system has single point of failure.

As you see in the below diagram of the existing system, Document Service and CUPS have single point of failure.



# Proposed

Below diagram depicts a proposed solution for providing Failover/Load Balancing for Document Service and CUPS. How to synchronize printers among tow CUPS system will be discussed in the Implementation section.



# Problem

In case if we have another failover server for CUPS the problem is how to synchronize set of printers among two CUPS instances running in different machine. So far there is no readymade tool or utility available for achieving this.

# Solution

To achieve synchronization among two CUPS instances (running in different boxes), we created a java utility which will run as a UNIX background process in each CUPS server.

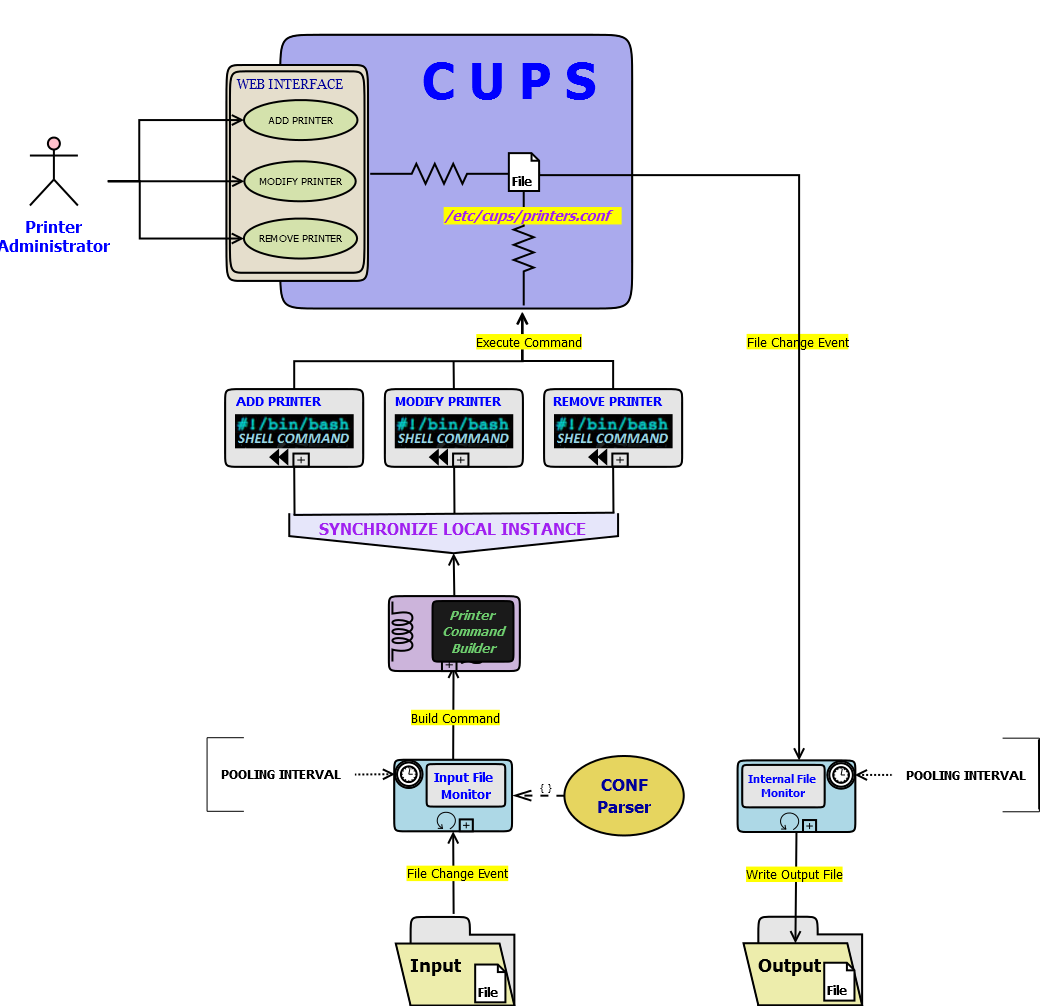
This java utility mainly consists of four components:

* **Printer Class** – Abstract representation of a Printer configuration object.
* **CONF Parser** – A java based parser for reading and parsing CONF files (CONF is a standard config file format for UNIX).
* **Printer Command Builder** – Java Class for building “lpadmin” command (lpadmin is a UNIX shell command for manipulating CUPS).
* **Input File Monitor (Thread)** – This is Java based Thread which is continuously monitoring an input file. If the last access timestamp of that file changes, it will use the CONF Parser to parse and create a unique list (Set) of Printer objects. Then it will read and parse internal CUPS configuration file: “/etc/cups/printers.conf” to create a unique list (Set) of Printer objects, same as before.   
  These two lists will be compared to identify:
  + Unique List (Set) of Printers **Added**.
  + Unique List (Set) of Printers **Removed**.
  + Unique List (Set) of Printers **Modified**.

Once the modifications (ADD, UPDATE, and DELETE) are identified, with the help of Printer Command Builder this will generate add, update & delete UNIX Commands (*lpadmin*) and will execute these shell command one by one.

* **Internal File (printers.conf) Monitor -** This is also Java based Thread which is continuously monitoring internal CUPS configuration file: “/etc/cups/printers.conf”. If the last access timestamp of that file changes then it will copy internal printers.conf file to the output directory.

System architecture & process flow of CUPS synchronizer:

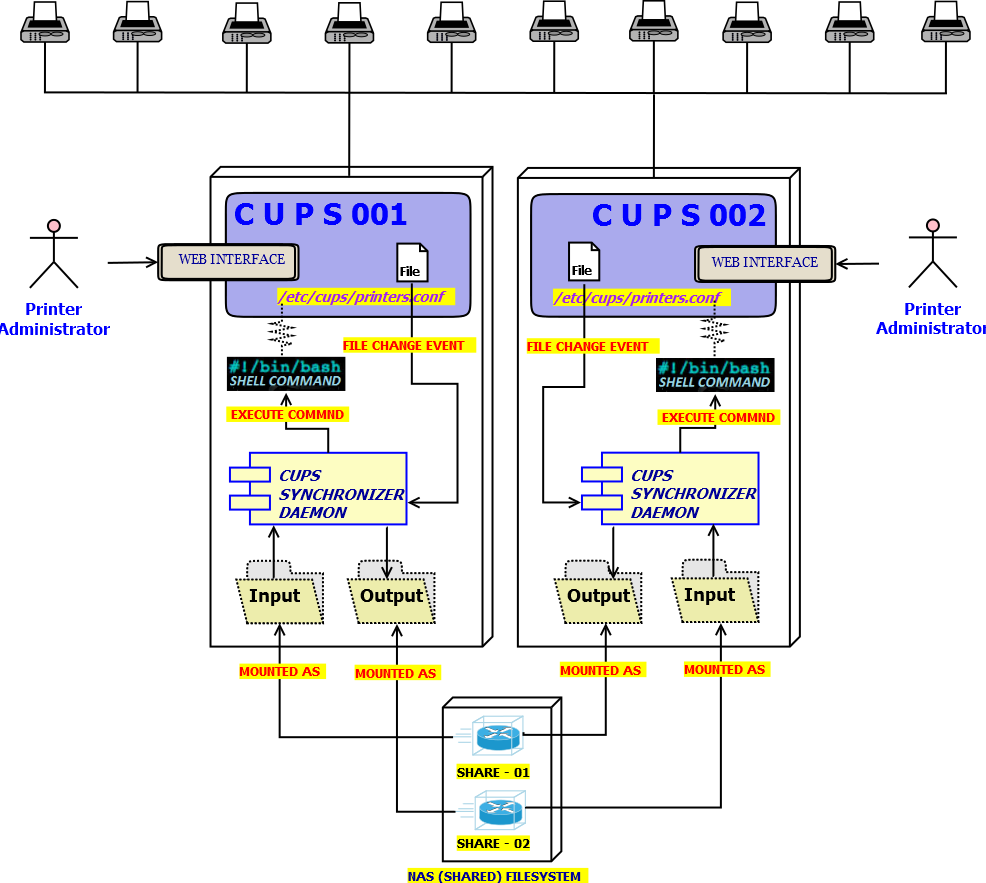


The java utility is packaged as jar and can be used as standalone or inside an app server.

As per proposed design, each of the CUPS servers will be having two shared file system (NAS or any other shared file system) mounted as “Input” and “Output”. “Input” folder for one CUPS Server is “Output” folder for another CUPS server. Same way “Output” folder for one CUPS Server is “Input” folder for another.

The above Java program will monitor “Input” folder to synchronize Printers with locally running CUPS instance. It will also copy the internal configuration file to the “Output” folder (in case and changes found in the internal config file) which will be the input for second CUPS server. Second server will also synchronize Printers in the same way.

Logical diagram of CUPS Synchronizer:



# Limitations and TODO(s)

Below are the known limitations or TODO items of this POC project. Addressing these limitations, TODO(s) will be part of actual project (Not in the scope of POC).

1. This POC has been done using Ubuntu 14.04 and CUPS 1.7.2. We need to configure two REDHAT Linux (Fedora Core 6) systems for implementation, integration and testing. -- Implementation will be part of actual project.
2. The process (CUPS Synchronizer) needs to be running by “**root**” user, because the CUPS internal configurations file: “/etc/cups/printers.conf” has only access for the “**root**” user. –- Needs to evaluate more on this and it will be part of actual project.
3. Synchronizing printer property update is not properly implemented. Only Add and Remove operations are implemented and tested so far. –- Update implementation will be part of actual project.
4. “Input”, “Output” folder paths and pooling interval for both input and output monitors, needs to be externalized. -- Implementation will be part of actual project.
5. Logging, Monitoring, Notification needs to be implemented for easy maintenance. -- Implementation will be part of actual project.
6. Synchronization takes some time as a print job can be send to a specific printer in the mean time. Availability of a specific Printers need to be determined before sending a job to a specific CUPS instance. –- Needs to evaluate more on this and it will be part of actual project.

# Performance

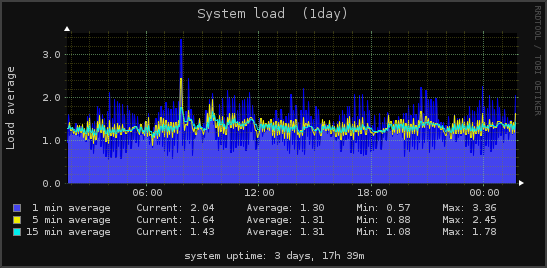
As per the implementation plan there will be a demon (cups-synchronizer), running in each Linux boxes (CUPS server). To ensure the performance impact we used Monitorix to profile the VM (Ubuntu).

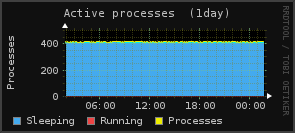
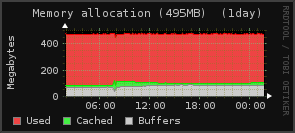
System configuration used for each VM is given below:

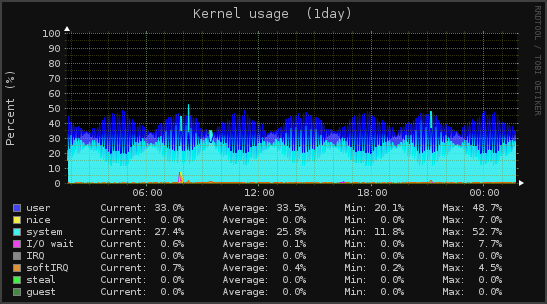
* **Processor:** 1 CORE Allocated from - [Intel® Core (TM2) Duo @2.66 GHz].
* **RAM:** 512 MB.
* **Storage:** 8:00 GB.
* **Network:** Bridge Mode [Intel® 82567LM-3 Gigabit network Connection].

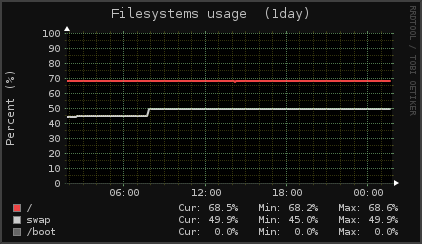
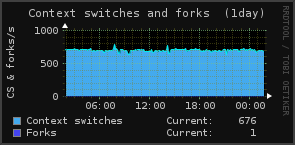
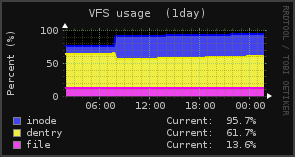
System Usage summary given below:

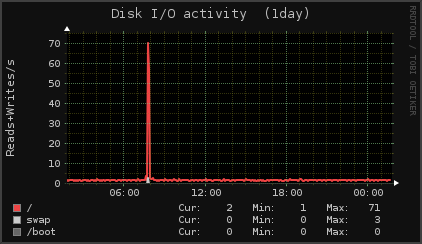
* **System Load:** 1.31% AVG.
* **Processor:** 400 AVG.
* **RAM:** 495 MB AVG.

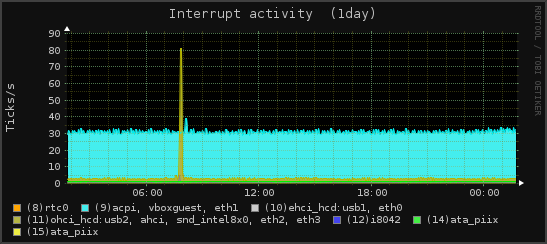
Screenshots given below: [](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/system1z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))

[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/system2z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/system3z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))

[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/kern1z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))

[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/fs01z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/kern2z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/kern3z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))

[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/fs02z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))

[](javascript:void(window.open('http://10.74.173.68:8080/monitorix/imgs/int1z.1day.png','','width=915,height=400,scrollbars=0,resizable=0')))

# OS, Software, Tools & Technologies Used

* **VM Platform:**
  + Oracle VirtualBox (4.3.20). <https://www.virtualbox.org/>
* **OS Used**:
  + Ubuntu (14.04 LTS). <http://www.ubuntu.com/>
* **Software**:
  + CUPS (1.7.2). <http://cups.org/> <http://www.computerhope.com/unix/ulpadmin.htm>
  + OpenJDK (1.7.0.75). <http://openjdk.java.net/>
  + Monitorix (3.0.6). <http://www.monitorix.org/>
* **Programming Languages**:
  + Java (1.7). <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
  + UNIX shell script. <http://www.cs.columbia.edu/~sauce/tutorial/shell.html>