Content Oriented Printer AutoSelection

OpenPrinting, The Linux Foundation

Deepak Patankar
Mentor : Till Kamppeter

About the project

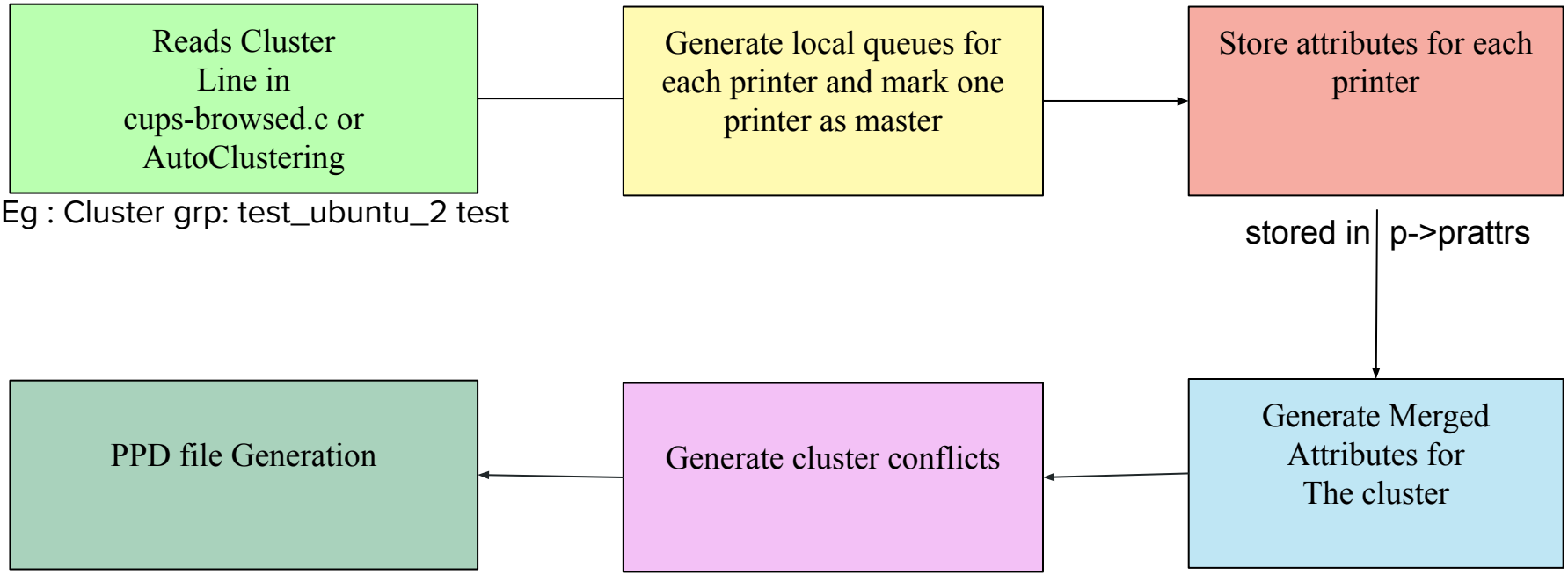
To print a document, one first needs to select a printer and then adjust its settings to fit their requirement. This project aims at enhancing user experience by building a module which automatically selects a printer based on printing options given by the user. The user need not worry about what all printers are available and their capabilities. Depending on printing options set by the user such as color scheme, paper size, the system will automatically send the job to the best available printer and will notify the user about the same.

Earlier we were able to cluster only same printers, this module will allow clustering different printers too.

Tasks:

1. PPD Generation for Remote Cups Queue
2. Clustering remote cups queue and IPP Network Printer
3. Generating Merged Attributes for the cluster
4. Generating conflicts for the cluster
5. Selecting the default attributes of the printer
6. Generating ppd file for the cluster
7. Selecting the printer for the job
8. Filtering job on the server

Generating Cluster:



Tasks:

1. PPD Generation for Remote Cups Queue
    a. Earlier for remote cups queues, the ppd file was received from the server.
    b. Now the ppd file for remote cups queue is generated using the IPP Response message sent by the server which contains all the printer attributes.

Tasks:

2. Clustering Remote Cups Queue and IPP Network Printer
    a. Earlier the clustering was only implemented for the remote cups queue, now we can make a cluster of remote cups queue and IPP Network Printer.
    b. For facilitating the same the following changes were done
        i. Changed implicitclass backend, so that the job is directly sent to the printer, rather than queuing job on the local cups server. Since ipp backend works for both remote cups queue and IPP Network Printer, the implicitclass backend now sends the job to the ipp backend.
        ii. Changed cups-browsed.c, now it sends the uri of the printer to the implicitclass backend.
        iii. For Load Balancing Type *queue_on_server*, changed cupsGetJob function so that number of jobs is directly queried to the printer.

Tasks:

3. Generating Merged Attribute for the cluster

Printer A

- **Color Suppported :** Yes
- **Duplex:** One-sided
- **PageSize:** A3, A4, A5, Letter,Legal
- **Resolution :** 300dpi,600dpi
- **ColorModel :** SRGB, Gray

Printer B

- **Color Suppported :** No
- **Duplex:**one-sided, DuplexNoTumble
- **PageSize:** A4, A5
- **Resolution :** 600dpi

Cluster

- **Color Suppported :** Yes
- **Duplex:** one-sided, DuplexNoTumble
- **PageSize: A3,**A4, A5, Letter,Legal
- **Resolution :** 300dpi, 600dpi
- **ColorModel:** SRGB, Gray

Tasks:

4. Generating conflicts for the cluster

Printer A

- **Color Supported :** Yes
- **Duplex:** One-sided
- **PageSize:** A3, A4, A5, Letter,Legal
- **Resolution :** 300dpi,600dpi
- **ColorModel :** SRGB, Gray

Printer B

- **Color Suppported :** No
- **Duplex:**one-sided, DuplexNoTumble
- **PageSize:** A4, A5
- **Resolution :** 600dpi
- **Media-type:**auto,,envelope

Cluster

Conflicts pair

A3	DuplexNoTumble
Letter	DuplexNoTumble
Legal	DuplexNoTumble
Envelope	Letter
Envelope	Legal
……	
…….	

Tasks:

5. Selecting the default attributes for the cluster

   One way of selecting the default value for the cluster could be setting the value which is supported by maximum number of printers.

	Printer A	**Printer B**	**Printer C**	**Printer D**
Default PageSize	A4	A5	A4	Letter
Default MediaType	Auto	Envelope		Envelope

So the cluster will have default Pagesize **A4** and default MediaType **Envelope**.
But this we way we will generate a constraint for A4, Envelope. But we can't have constrint between default options.
So this method won't work...

Tasks:

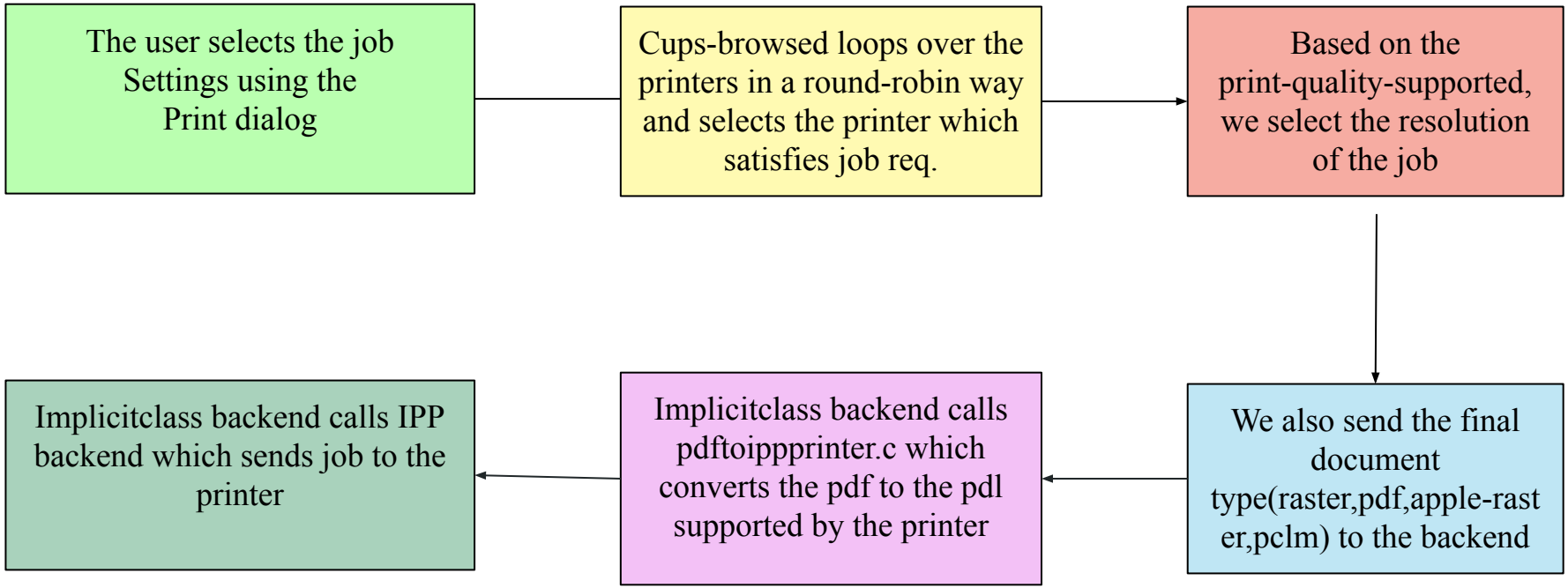
5. Selecting the default attributes for the cluster
    a. The printer with the maximum throughput (pages/min) is selected as the default printer.
    b. The default attributes of the cluster will be the attribute of this printer.

Tasks:

6. Generating ppd file for the cluster

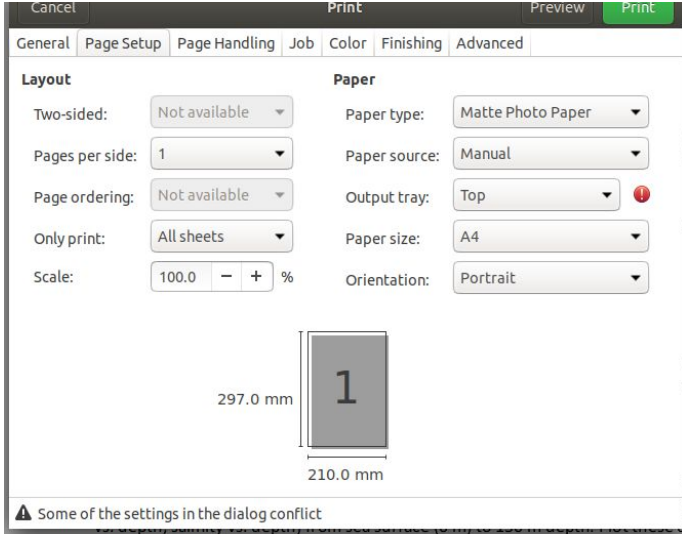
   cups-browsed sends the merged attributes to the ppdCreateFromIPP function in ppdgenerator.c which generates the ppd file.

Printing the job:



Tasks:

7. Selecting the job for the cluster
    a. One can use print dialog to set the job settings.
    b. If there is a option conflict, the user will get see red circle and he must not print that job.



Tasks:

7. Selecting the job for the cluster
    a. Next the cups-browsed reads the settings requested by the user, and selects the printer From the list which supports the requested options.

List of job options checked:

1. Document-format-supported
2. Job-sheets
3. Multiple-document-handling
4. MediaType
5. StapleLocation
6. FoldType
7. PunchMedia
8. ColorModel
9. Duplex
10. Orientation-requested
11. PageSize
12. PrintQuality

Tasks:

8. Filtering jobs on the server
    a. Created pdftoippprinter.c (copy of sys5ippprinter.c) which takes pdf file as input and converts in into the pdl supported by the printer.

Thank You

Content Oriented Printer Autoselection : GSoC 2018

Content Oriented Printer Autoselection : GSoC 2018

Content Oriented Printer AutoSelection

OpenPrinting, The Linux Foundation