Balanceamento de Carga na Rasterização de Documentos PDF

Maiquel Breitenbach, Carolina Fonseca, Mateus Raeder,
Mariana Kolberg, Luiz Gustavo Fernandes

Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Av. Ipiranga, 6681 - Prédio 32 - PPGCC
{maiquel.breitenbach, carolina.fonseca, mateus.raeder}@acad.pucrs.br,
{mariana.kolberg, luiz.fernandes}@pucrs.br

Resumo

Com o surgimento das impressoras digitais, procedimentos automatizados para a criação e transformação de documentos personalizados tornaram-se necessários, a fim de suprir a demanda existente pelas *Print Shops*. Algumas estratégias foram introduzidas para aumentar o desempenho na fase de rasterização através do uso de técnicas relacionadas ao processamento paralelo e distribuído. Porém, estas estratégias apresentam alguns problemas, como a impossibilidade da garantia de um bom balanceamento de carga. Em trabalhos anteriores, foram propostas estratégias para aumentar o desempenho desta fase. Tais estratégias apresentaram ganhos, entretanto, algumas otimizações ainda podem ser aplicadas. Neste contexto, este trabalho propõe novas formas de divisão dos *jobs* para obtenção de melhor balanceamento de carga, levando em conta características importantes em documentos PDF (*Portable Document Format*), tais como a transparência de objetos.

Introdução

Na etapa de renderização, alguns trabalhos envolvendo processamento paralelo já foram realizados para otimizar o desempenho [GFT 06] [NRK 09]. Já na etapa de rasterização, trabalhos mais recentes vem sendo realizados. O foco do presente estudo é a etapa da rasterização, que é descrita em maiores detalhes a seguir. A fase de rasterização ou *RIPping* (*Raster Image Processing*) consiste na conversão do documento em um formato conhecido pelas impressoras, uma vez que estas não são capazes de interpretar linguagens de alto nível (como PDF) para descrição de documentos.

Através do uso de técnicas relacionadas ao processamento paralelo e distribuído, algumas estratégias já foram aplicadas para aumentar o desempenho desta etapa [NGK 09], permitindo a utilização das impressoras em sua máxima capacidade. Porém, tais estratégias apresentam alguns problemas, como a impossibilidade de garantir um balanceamento de carga justo para quaisquer sequências de *jobs* contendo documentos personalizados. Assim, o objetivo deste trabalho é encontrar diferentes tipos de estratégias para um bom balanceamento de carga entre os RIPs na quebra de documentos PDF, levando em consideração diferentes características, como transparência e reusabilidade.

Contextualização do Problema

Os documentos PDF são formados por objetos [PDF 03], os quais possuem algumas características importantes que podem influenciar no balanceamento dos RIPs. Estes objetos podem ser classificados como:

- ***Path Objects***: representam formas arbitrárias , trajetórias, regiões e *paths*;

- ***External Objects***: representam elementos que podem ser reusáveis. Estão contidos nestes objetos as imagens transparentes. Três sub-tipos de *xObjects* são conhecidos:
  - *Images xObjects*: descrevem imagens *bitmap* que representam seu conteúdo através de uma matriz de *pixels*;
  - *Postscript xObject*: objetos que definem seu conteúdo através de PS *commands*. Estes objetos estão deixando de ser usados em PDFs;
  - *Groups (forms) xObjects*: grupo de objetos gráficos, usados para definir propriedades comuns dentro de um grupo;
- ***Inline Image Objects***: definem uma imagem diretamente dentro do PDF (não pode ser reusável). Esses objetos contêm diversas limitações, sendo a mais relevante o tamanho da imagem;
- ***Shading Pattern Objects***: descrevem uma forma geométrica (que tem sua cor definida por uma função arbitrária);
- ***Text Objects***: descrevem as porções de texto do documento, incluindo o formato e suas características, tais como fonte, tamanho, espaçamento entre letras, etc.

Em [NGK 09], foi apresentada uma ferramenta chamada PDF *Profiler*, capaz de analisar e fornecer todas as informações dos objetos contidos em cada página de um documento PDF. Com o uso desta ferramenta, foi possível realizar a análise do perfil dos *jobs*, concentrando-se no formato PDF (por ser um formato difundido na descrição de documentos personalizados). A partir das informações fornecidas pelo PDF *Profiler*, pode-se obter um balanceamento de carga mais justo para cada RIP.

Estratégias e Discussão

Em um ambiente de rasterização tradicional, as estratégias existentes baseiam-se em sistemas paralelos e distribuídos para aumentar a vazão e o desempenho de tal fase. Assim, diversos RIPs são aplicados em conjunto para rasterizar uma dada fila de *jobs* de forma paralela. Desta forma, através da análise de cada estratégia é possível verificar as vantagens e desvantagens existentes. Deste modo, são identificadas três estratégias que são aplicadas para acelerar a rasterização dos *jobs*:

1. **alocar um RIP por *job***: cada RIP existente irá processar um *job* inteiro. Existem duas situações que podem decorrer: a distribuição da carga é injusta ou vários RIPs ficam ociosos. Este tipo de cenário funciona bem para *jobs* pequenos, com alta reusabilidade;

2. **alocar todos RIPs para um único *job***: esta é a abordagem força-bruta, onde cada RIP irá processar uma porção de um dado *job*. Esta estratégia funciona bem com *jobs* que apresentam baixa reusabilidade e demandam alto poder computacional;

3. **alocar um número fixo de RIPs por *job***: configuração baseada no fato de que os *jobs* padrões das PSPs (*Print Service Providers*) necessitam de um determinado número de recursos para manterem as impressoras continuamente trabalhando. Uma desvantagem é o fato de que um conjunto de RIPs somente será alocado para um novo *job* assim que todos os RIPs estiverem livres, ou seja, no momento em que terminar o processamento do *job* atual (aplica-se a estratégia 2 também).

Nenhuma das estratégias discutidas garante o melhor balanceamento de carga entre os RIPs. Porém, as estratégias 2 e 3 apresentam uma eficiência maior do que a estratégia 1. Como exemplo, imagina-se o seguinte cenário: na primeira estratégia, 3 RIPs e 3 *jobs*.

O primeiro *job* é executado em 100s, o segundo em 2s e o terceiro em 50s. Neste exemplo, pode-se perceber facilmente que o segundo RIP ficará ocioso por um tempo muito grande (98s).

Balanceamento

Esta seção aborda a importância do balanceamento de carga para o ganho de tempo no processamento dos *jobs*. Como citado na seção anterior, as estratégias 2 e 3 foram as que se mostraram mais eficientes no balanceamento de carga. Cabe ressaltar que o grão mínimo de cada parte do *job* é uma página, sendo assim, não poderá ocorrer a existência de um *job* com grão menor que um nem páginas quebradas. Outro fator de alta influência no processamento dos *jobs* pelos RIPs são os objetos que cada *job* possui. Em [NRK 09] foi observado que *jobs* de tamanhos iguais podem demorar tempos diferentes dependendo de suas características, isso é, objetos contidos em cada página.

Desta forma, foram criadas algumas políticas para balancear a divisão dos grãos dos *jobs* entre os RIPs. Em algumas destas políticas foram levadas em conta as características de cada documento PDF.

Uma primeira abordagem foi a divisão dos *jobs* de forma direta entre os RIPs, na qual cada RIP fica com um intervalo de páginas igual (sem páginas quebradas), não levando em conta as características de cada *job*. Assim, um *job* com 10 páginas deveria ser distribuído em 2 RIPs. A distribuição seria de 5 páginas para cada RIP (sendo as 5 primeiras para o RIP1, e as demais para o RIP2). Esta política, entretanto, mostrou-se ineficiente, pois poderia ocorrer uma situação na qual as 5 primeiras páginas do *job* do exemplo contivessem imagens transparentes, que são as imagens que mais consomem processamento nos RIPs [NUN 09]. Por outro lado, as 5 páginas finais poderiam conter somente texto, que são rapidamente processados pelos RIPs (mais rápido que imagens comuns e transparentes). O tempo total do processo é o tempo que o último RIP leva para encerrar seu processamento, ou seja, o tempo do processamento do RIP com as imagens transparentes. Enquanto isso, o RIP que processou os objetos de texto ficaria ocioso.

Outra alternativa foi utilizar as características dos documentos PDF para um melhor balanceamento de carga. A característica que mais influencia no tempo de processamento de um *job* é a presença ou não de imagens transparentes. Pensando nisso, desenvolveu-se um algoritmo que leva em conta estas características. O algoritmo separa as páginas transparentes e as distribui entre os RIPs disponíveis, de forma que os RIPs fiquem com uma carga similar de transparências. Para melhor entendimento dessa política, considere o exemplo anterior. Neste algoritmo (que leva em conta a transparência), o primeiro RIP ficaria com 3 páginas transparentes e 2 não-transparentes (podendo ser texto, imagens comuns ou outros objetos). O segundo RIP, por sua vez, ficaria com 2 transparentes e 3 não-transparentes. Deste modo, o processamento é otimizado, tendo um ganho significativo de tempo na execução do *job*.

Para a demonstração deste ganho de desempenho, foram feitos alguns testes com PDF de características diferentes para as duas abordagens descritas. A seguir serão descritos as características dos PDFs utilizados.

- **PDF 1**: Contém 9 páginas ao total, tendo as 4 primeiras imagens transparentes e as outras 5 imagens comuns;
- **PDF 2**: Contém 15 páginas ao total, tendo as 6 primeiras imagens transparentes e as outras 9 imagens comuns;
- **PDF 3**: Contém 9 páginas ao total, tendo as 3 do meio (páginas 4, 5 e 6) imagens transparentes e as demais páginas objetos texto.

A Tabela 1 demonstra os tempos, em segundos, para o processamento dos arquivos PDF supracitados, para uma distribuição entre 3 RIPs. Para realizar estes testes, utilizou-se um processador *Intel* Pentium 4 1.8 GHz com 376MB de memória e 40GB de disco.

Tabela 1: *Tempos em segundos do processamento do RIPs*

PDF	RIP	BALANC.	Tempos (s)
1	1	SEM	39.26
1	2	SEM	21.06
1	3	SEM	12.60
1	1	COM	30.06
1	2	COM	21.73
1	3	COM	21.00

(a)

PDF	RIP	BALANC.	Tempos (s)
2	1	SEM	64.66
2	2	SEM	29.20
2	3	SEM	20.70
2	1	COM	38.00
2	2	COM	37,90
2	3	COM	37,90

(b)

PDF	RIP	BALANC.	Tempos (s)
3	1	SEM	4,16
3	2	SEM	39,13
3	3	SEM	6,63
3	1	COM	17
3	2	COM	16,60
3	3	COM	16,20

(c)

Como pode ser observado na Tabela 1, quando os *jobs* que não fazem o balanceamento levando em conta a transparência, eles perdem tempo de processamento, pois um dos RIPs fica sobrecarregado com as imagens transparentes enquanto os outros RIPs ficam ociosos aguardando o termino do processamento do RIP que contém as páginas transparentes. Observe o PDF2 da Tabela 1 (b), o *job* (sem o balanceamento) com maior tempo de processamento foi no RIP1, com 64,66 segundos. Isso se deve por conseqüência do maior número de imagens transparentes contido nele. Já no *job* (com balanceamento) o maior tempo de processamento foi o RIP1, com 38 segundos (ganho de quase 50%).

Conclusão

Neste trabalho foi realizado o balanceamento das cargas dos *jobs* levando em consideração suas características. Concluiu-se que balancear as cargas por transparência é vantajoso e proporciona uma melhoria significativa no processamento dos *jobs*. Ainda há outras características a serem aplicadas no escalonamento, como a reusabilidade. Como trabalhos futuros, pretende-se implementar algoritmos utilizando esta característica.

Referências

[NUN 09] Nunes, T. Aplicando Estratégias de Escalonamento Através da Análise do Peril de jobs para Ambientes de Impressão Distribuídos. Dissertação (Mestrado) — Pontifícia Universidade do Rio Grande do Sul. Porto Alegre, Brasil, 2009.

[PDF 03] Adobe Systems. PDF Reference. 4th. ed. San Jose, USA, 2003.

[NRK 09] Nunes, T. ; Raeder, M. ; Kolberg, M. ; Fernandes, L. G. ; Cabeda, A. ; Giannetti, F. . High Performance Printing: Increasing Personalized Documents Rendering through PPML Jobs Profiling and Scheduling. In: 12th IEEE CSE - International Conference on Computational Science and Engineering, 2009, Vancouver (Canadá). Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE). Los Alamitos, CA (EUA) : IEEE Computer Society, 2009. p. 285-291.

[GFT 06] Giannetti, F. ; Fernandes, L. G. ; Timmers, R. ; Nunes, T. ; Raeder, M. ; Castro, M. . High Performance XSL-FO Rendering for Variable Data Printing. In: 21st ACM SAC - Symposium on Applied Computing, 2006, Dijon (França). Proceedings of the 21st ACM Symposium on Applied Computing (ACM SAC). Nova Iorque, NY (EUA): ACM Press, 2006. v.1. p.811-817.

[NGK 09] Nunes, T. ; Giannetti, F. ; Kolberg, M. ; Nemetz, R. ; Cabeda, A. ; Fernandes, L. G. . Job Profiling in High Performance Printing. In: 9th ACM Symposium on Document Engineering, 2009, Munique. Proceedings of the 9th ACM Symposium on Document Engineering (ACM DOCENG). Nova Iorque, NY (EUA) : ACM Press, 2009. p. 109-118.