

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Bacharelado em Ciência da Computação

Paralelização do Processo de Renderização de Documentos XSL-FO

– Tópicos em PPD I –

Mateus Raeder
Thiago Tasca Nunes

Professor: Prof. Dr. Luiz Gustavo Leão Fernandes

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Apresentação

- Introdução
- Objetivos
- Renderização
- FO+Processor Sequencial
- FO+Processor Paralelo
- Resultados Obtidos
- Considerações Finais
- Trabalhos Publicados

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Introdução

- Crescente demanda por documentos personalizados
- Por exemplo, empresas que precisam personalizar mensagens para cada cliente na promoção de materiais ou campanhas de *marketing*
- Utilização de *Variable Data Printing* (VDP)
- **FO+Processor** e **FOP** trabalham em conjunto para transformar documentos constituídos por VDP em um formato PDL (*Page Description Language*)
 - PDF
 - PS
 - SVG
- PDL então pode ser enviado para a impressão

FO+Processor identifica porções variáveis
FOP renderiza uma porção variável

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Objetivos

- A transformação para PDL depende da **quantidade de conteúdo variável a ser renderizado em cada documento** ⇒ **Alto custo computacional**
- Velocidade das impressoras em paralelo é maior do que a velocidade de renderização ⇒ **Gargalo**
- **FOP**: transforma uma porção de conteúdo variável em PDL por vez
(Tempo de renderização de uma única porção variável é pequeno)
- **FO+Processor**: permite a renderização de diversas porções variáveis
(Tempo de renderização de várias porções variáveis é grande)
- Estudou-se, então, este processo de renderização, visando a proposta de uma nova versão através das técnicas de programação paralela

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Renderização

- Renderização de documentos é o processo de interpretação de determinados tipos de dados que objetiva transformá-los em um conteúdo visualizável

VDP (Variable Data Printing)

- impressão de documentos personalizados de uma só vez
- 10 ou 1000 documentos em uma única demanda
- documentos contêm elementos com informação variável

nome de um cliente
nome de um produto
figuras

Template

Banco X
< nome >
< endereço >
< cep >
< cidade > < estado >

Banco X
João da Silva
Av. Ipiranga, 55555
00000-000
Porto Alegre RS

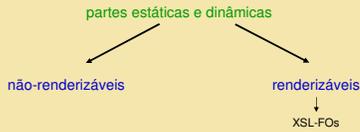
contas de bancos
contas de telefone
catálogos de marketing personalizados

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Renderização

PPML (Personalized Print Markup Language)

- linguagem padrão para impressão digital baseada em XML (*eXtensible Markup Language*)
- permite utilizar VDP em documentos de alta qualidade

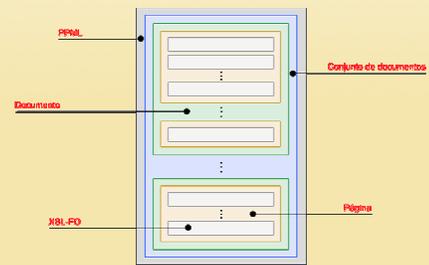


- serve de entrada para o FO+Processor

Paralelização do Processo de Renderização de Documentos XSL-FO
Makoto Suzuki | Thiago Tasso Nunes

Renderização

Composição do documento PPML

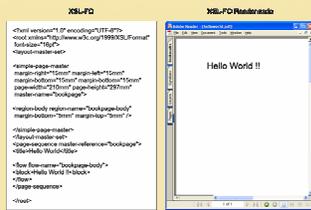


Paralelização do Processo de Renderização de Documentos XSL-FO
Makoto Suzuki | Thiago Tasso Nunes

Renderização

XSL-FO (eXtensible Stylesheet Language - Formatting Objects)

- linguagem de formatação de objetos com muitos recursos visuais
- áreas de trabalho que podem ser preenchidas com:
 - texto
 - imagens
 - espaço vazio
 - outro XSL-FO aninhado



- porções renderizáveis do documento

↓
transformadas em conteúdo visualizável

Paralelização do Processo de Renderização de Documentos XSL-FO
Makoto Suzuki | Thiago Tasso Nunes

Renderização

PDL (Page Description Language)

- linguagem de comando de alto nível de abstração



- otimiza o processo de impressão
- é independente do dispositivo

Alguns tipos de PDL

- PS
- PDF
- SVG
- DVI

Paralelização do Processo de Renderização de Documentos XSL-FO
Makoto Suzuki | Thiago Tasso Nunes

Renderização

SVG (Scalable Vector Graphics)

- padrão baseado em XML com grande suporte empresarial
- descreve gráficos bidimensionais
- foi desenvolvido com o intuito de ser utilizado em páginas da Internet
- versões posteriores passaram a se preocupar não apenas com a Internet
- destaca-se a área de impressão
- independência de resolução da impressão
- integração ao padrão PPML para a impressão de dados variáveis



Paralelização do Processo de Renderização de Documentos XSL-FO
Makoto Suzuki | Thiago Tasso Nunes

Renderização

FOP (Formatting Objects Processor)

- engine de renderização mais popular e difundida no mercado
- aplicação Java
- lê um objeto de entrada e transforma-o em PDL
- o arquivo de entrada deve ser formatado pelo padrão XSL-FO



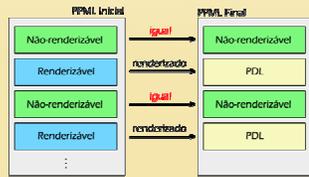
FO+Processor (Formatting Objects Plus Processor)

- a engine FOP é complementada pela ferramenta FO+Processor
- busca as porções compostas por XSL-FO nos documentos PPML de entrada
- extrai do documento PPML de entrada suas partes variáveis e encaminha-as à engine de renderização
- aplicação Java

Paralelização do Processo de Renderização de Documentos XSL-FO
Makoto Suzuki | Thiago Tasso Nunes

FO+Processor Sequencial

- O arquivo de saída do FO+Processor é formado pela mesma estrutura do documento PPML de entrada



- Pode ser subdividido em dois módulos principais: FOExtractor e FOIterator

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Bueder | Thiago Tasso Nunes

FO+Processor Sequencial

FOExtractor

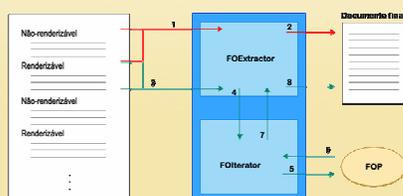
- realiza o processo de *parsing* (análise) do documento de entrada
- identifica
 - não-renderizáveis
 - renderizáveis
- encontra uma porção renderizável através de uma *tag indicativa*
 - Tag do início de um XSL-FO
 - Tag do final de um XSL-FO
- envia porção variável ao FOIterator
- escreve no arquivo de saída

FOIterator

- pré-formata o XSL-FO
 - formata novamente em caso de overflow
- envia para o FOP renderizar
- recebe o resultado da renderização e envia para o FOExtractor

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Bueder | Thiago Tasso Nunes

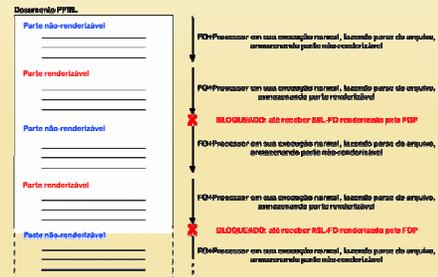
FO+Processor Sequencial



- FOExtractor armazena porção não-renderizável
- FOExtractor escreve porção não-renderizável no documento final
- FOExtractor armazena porção renderizável (XSL-FO)
- FOExtractor encaminha o XSL-FO
- FOIterator recebe XSL-FO renderizado no documento final
- FOIterator formata o XSL-FO
- FOIterator envia o XSL-FO para o FOP renderizar
- FOP retorna o XSL-FO renderizado ao FOIterator

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Bueder | Thiago Tasso Nunes

FO+Processor Paralelo



Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Bueder | Thiago Tasso Nunes

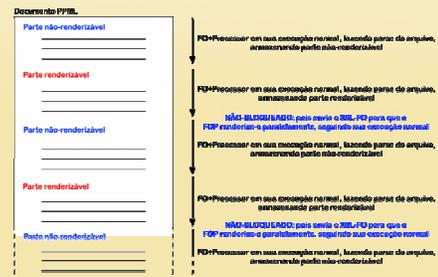
FO+Processor Paralelo

- Medições para validar a hipótese do possível gargalo:

Arquivo	Tempo total	Tempo de bloqueio	Porcentagem
Fire1000	145.8738s	102.0561s	70%
Fire2000	283.6217s	197.8023s	70%
Earth1000	225.5180s	159.4605s	71%
Earth2000	444.2165s	312.8148s	70%
Water1000	264.8311s	178.9853s	68%
Water2000	523.5089s	351.7017s	67%
Wind1000	416.0129s	278.6795s	67%
Wind2000	835.8158s	558.3989s	67%

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Bueder | Thiago Tasso Nunes

FO+Processor Paralelo



Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Bueder | Thiago Tasso Nunes

FO+Processor Paralelo

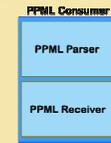
- Foco da paralelização: otimizar o modo de execução do FO+Processor
- Módulo FOExtractor
- Obstáculo:
 - tempo de escrita em arquivo
- Execução e testes:
 - Ganho apenas no processamento dos XSL-FOs
- Cluster Ombrófila (CPAD - Centro de Pesquisa em Alto Desempenho)
- Comunicação entre os processos: troca de mensagens
- Biblioteca MPI (mpiJava)
- Threads
- Modelo mestre-escravo, com uma hierarquia diferente

Paralelização do Processo de Renderização de Documentos XSL-FO
 Marcos Rueder | Trabalho Técnico Números

FO+Processor Paralelo

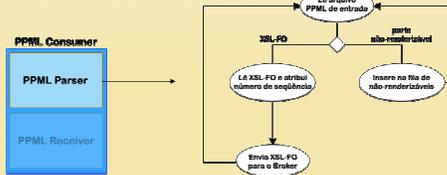
Implementação com 3 Módulos

- PPMLConsumer
- leitura e extração de XSL-FOs do arquivo de entrada
- escrita no arquivo final
- divisão em 2 threads
 - PPMLParser
 - PPMLReceiver



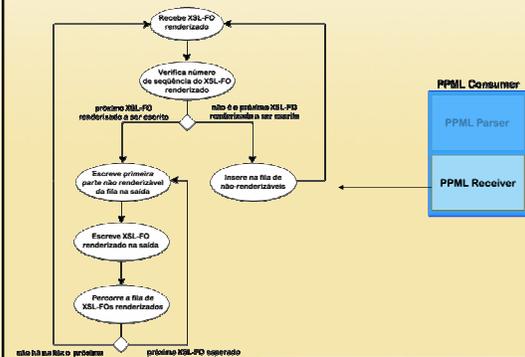
Paralelização do Processo de Renderização de Documentos XSL-FO
 Marcos Rueder | Trabalho Técnico Números

FO+Processor Paralelo



Paralelização do Processo de Renderização de Documentos XSL-FO
 Marcos Rueder | Trabalho Técnico Números

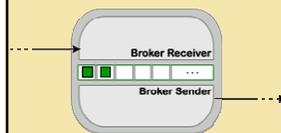
FO+Processor Paralelo



Paralelização do Processo de Renderização de Documentos XSL-FO
 Marcos Rueder | Trabalho Técnico Números

FO+Processor Paralelo

- Broker
- recebe os XSL-FOs do PPMLConsumer
- envia para o FOP

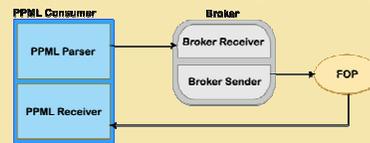


- precisa receber e enviar XSL-FOs
 BrokerReceiver BrokerSender

Paralelização do Processo de Renderização de Documentos XSL-FO
 Marcos Rueder | Trabalho Técnico Números

FO+Processor Paralelo

- FOP
- renderiza os XSL-FOs recebidos do Broker
- envia o resultado para o PPMLConsumer
- avisa o Broker quando estiver livre



- mínimo 3 unidades ativas
- cada unidade adicionada = componente FOP adicional

Paralelização do Processo de Renderização de Documentos XSL-FO
 Marcos Rueder | Trabalho Técnico Números

FO+Processor Paralelo

Implementação com Buffers de XSL-FOs e 4 Módulos

- **Buffer de XSL-FOs**
 - um XSL-FO por vez ⇒ **overhead de comunicação**

Desequilíbrio de carga

□ XSL-FO

- equilibrar o peso das tarefas ⇒ enviar aproximadamente a mesma carga de trabalho para todos

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Rueder | Thiago Tasso Nunes

FO+Processor Paralelo

- agrupamento de XSL-FOs em buffers ⇒ **tamanho em bytes**

Tamanho	Tempo
32Kb	0,25s
64Kb	0,51s
128Kb	0,98s
256Kb	1,94s

- PPMLParser ⇒ **agrupa**
- Broker ⇒ **enfila**
- FOP ⇒ **renderiza**
- PPMLReceiver ⇒ **recebe XSL-FOs renderizados**

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Rueder | Thiago Tasso Nunes

FO+Processor Paralelo

- integridade dos dados ⇒ **XSL-FOs pela metade** ⇒ **Maior quantidade POSSÍVEL de XSL-FOs**
- sobrecarga de comunicação e sub-utilização dos recursos

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Rueder | Thiago Tasso Nunes

FO+Processor Paralelo

• 4 Módulos

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Rueder | Thiago Tasso Nunes

Resultados Obtidos

Casos de teste

Fire1000 e Fire2000

Página 1: 1 XSL-FO (aproximadamente 1,53Kb);
Página 2: 3 XSL-FOs (aproximadamente 1,17Kb, 1,23Kb e 1,4Kb respectivamente).

Tamanho total aproximado:
Fire1000 – 5330Kb
Fire2000 – 10660Kb

Earth1000 e Earth2000

Página 1: 3 XSL-FOs (aproximadamente 1,22Kb, 1,29Kb e 1,18Kb respectivamente);
Página 2: 3 XSL-FOs (aproximadamente 1,29Kb, 1,17Kb e 2,02Kb respectivamente).

Tamanho total aproximado:
Earth1000 – 8174Kb
Earth2000 – 16340Kb

Water1000 e Water2000

Página 1: 2 XSL-FOs (aproximadamente 1,49Kb e 2,24Kb respectivamente);
Página 2: 2 XSL-FOs (aproximadamente 3,33Kb e 1,29Kb respectivamente).

Tamanho total aproximado:
Water1000 – 8350Kb
Water2000 – 16700Kb

Wind1000 e Wind2000

Página 1: 3 XSL-FOs (aproximadamente 1,32Kb, 1,4Kb e 1,51Kb respectivamente);
Página 2: 5 XSL-FOs (aproximadamente 3,69Kb, 1,4Kb, 1,99Kb, 1,36Kb e 1,29Kb respectivamente).

Tamanho total aproximado:
Wind1000 – 13960Kb
Wind2000 – 27920Kb

Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Rueder | Thiago Tasso Nunes

Resultados Obtidos

Análise dos resultados

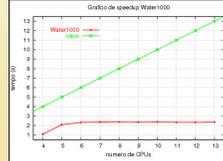
- **Tempos da versão sequencial**

Documentos	Fire		Earth		Water		Wind	
	1000	2000	1000	2000	1000	2000	1000	2000
Tempo (s)	145,8738	283,6217	225,5100	444,2165	264,8311	523,5089	416,0129	835,6158

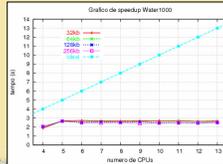
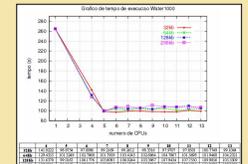
Paralelização do Processo de Renderização de Documentos XSL-FO
Márcio Rueder | Thiago Tasso Nunes

Resultados Obtidos

• Versão com 3 Módulos



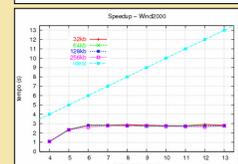
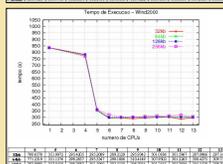
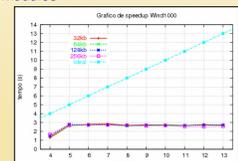
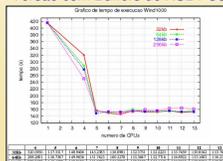
• Versão com Buffers de XSL-FOs e 4 Módulos



Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Resultados Obtidos

• Versão com Buffers de XSL-FOs e 4 Módulos



Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Resultados Obtidos

• Diferença entre os diferentes tamanhos de buffers

- Broker pode ser um gargalo → **Replicação do Broker**
- trabalho dos FOPs mais veloz que os outros componentes
↓
Cálculo do tempo de parsing (PPMLParser)

Estimativa da quantidade de trabalho

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Considerações Finais

- Diversas técnicas e metodologias foram aplicadas para a criação da versão paralela
- O ganho obtido comprova que é possível atingir melhores resultados na renderização de documentos
- Futuramente:

Paralelização do componente FOIterator para atingir as mesmas funcionalidades da versão sequencial

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes

Trabalhos Publicados

- A solução implementada e os resultados obtidos através deste estudo, foram assuntos de um artigo publicado no presente ano, em cooperação com o **Centro de Pesquisa e Desenvolvimento de Aplicações Paralelas (CAP)**, que é um projeto do convênio **PUCRS / FACIN – HP Brasil**.

FERNANDES, L. G.; GIANNETTI, F.; TIMMERS, R.; NUNES, T.; RAEDER, M.; CASTRO, M. **High Performance XSL-FO Rendering for Variable Data Printing**. In: ACM SAC - ACM Symposium on Applied Computing - Document Engineering Track, 2006, Dijon, França. Proceedings of the 2006 ACM Symposium on Applied Computing, 2006.

Paralelização do Processo de Renderização de Documentos XSL-FO
Mateus Raeder | Thiago Tasca Nunes



Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Bacharelado em Ciência da Computação



Paralelização do Processo de Renderização de Documentos XSL-FO

– Tópicos em PPD I –

Mateus Raeder
Thiago Tasca Nunes

Professor: Prof. Dr. Luiz Gustavo Leão Fernandes

Porto Alegre, outubro de 2006.