

# Estudo de uma aplicação distribuída em grade, caso: GerpavGrid

Mauro Storch, Rafael Garcia  
storch@inf.pucrs.br, rafaelsaraivagarcia@gmail.com

20 de novembro de 2006

## Resumo

A necessidade de estradas em boas condições proporciona melhor movimentação de pessoas e bens. Um software para análises de degradação de pavimento é indispensável para a boa conservação de ruas e rodovias. Para que este possa fornecer respostas rápidas e objetivas, a utilização de uma grade computacional surge como um meio que fornece alto desempenho na execução de aplicações distribuídas.

## 1 Introdução

O presente trabalho tem por objetivo apresentar o artigo "GerpavGrid: using the Grid to maintain the city road system" [2]. O artigo foi escrito por César de Rose e Tiago Ferreto (professores da Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS), Marcelo Farias e Vladimir Dias (Arquitetos de Software da empresa DBServer), Walfredo Cirne e Milena Oliveira (professor e assistente de pesquisa da Universidade Federal de Campina Grande - UFCG), Katia Saikoski (HP Brasil) e Maria Danieleski (SMOV). O artigo citado foi publicado no SBAC-PAD 2006.

O trabalho está focado na idéia de que a cidade de Porto Alegre necessita de boas estradas e ruas, permitindo a movimentação de pessoas e bens. Porém, a cidade vem testemunhando o aumento de tráfego nas suas vias devido ao seu crescimento. Sendo assim, ações rápidas são necessárias visando sempre a melhoria e manutenção da malha viária da cidade.

O órgão responsável pela conservação e avaliação da pavimentação urbana de Porto Alegre é o SMOV (Secretaria Municipal de Obras e Viação). Nos últimos anos, segundo dados da própria secretaria, ela tem sido desafiada a lidar com orçamentos limitados. Assim, é necessário que se tenha ainda mais eficiência na aplicação dos recursos públicos.

## 2 Gerpav

O SMOV decidiu desenvolver um sistema de informação para auxiliar na função de gerenciamento da pavimentação. Este sistema ganhou o nome de Gerpav. A principal função do sistema é auxiliar na tomada de decisão sobre

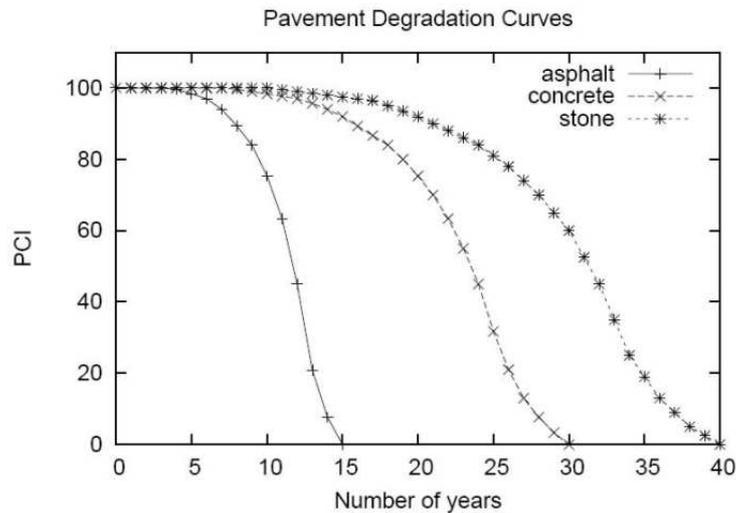


Figura 1: Degradação de pavimentos

a avaliação e manutenção da pavimentação. Este sistema foi concluído, em sua fase inicial, no final de 2004.

O Gerpav auxilia a SMOV na manutenção de informações sobre a malha viária, como tipos de pavimentos, defeitos, pistas e intervenções. Além disto, possui módulos de relatórios e simulação sobre a degradação dos pavimentos.

O sistema divide a malha viária em arcos, sendo que cada um representa um segmento de uma rua e pode conter até quatro pistas. Cada pista, por sua vez, é dividida em um número de faixas. Cada faixa contém informações sobre o tipo de pavimento e levantamentos de campos relatando defeitos. A partir destas informações, o sistema é capaz de gerar o índice de condição do pavimento (PCI). Este valor é calculado para cada faixa e para a pista, de forma global. Seus valores podem variar de 0 a 100, onde 100 representa condição perfeita. Cada tipo de pavimento possui comportamento e performance diferentes, conforme ilustrados pela Figura 1.

A base de dados que o Gerpav trabalha é composta por 26 tabelas e ocupa em torno de 200Mb de espaço em disco. As principais tabelas são Arco (31.252 registros), Pista (32.408 registros) e Faixa (217.141 registros).

O Gerpav pode ser dividido em seis sub-sistemas: - Generic Tables Maintenance Subsystem: responsável pela manutenção de dados em tabelas genéricas (tipos de pavimento, tipos de defeitos, tráfego, etc). - Road System Maintenance Subsystem: responsável pela manutenção das tabelas relacionadas à malha viária (rua, pistas, faixas). - Road System Report Subsystem: responsável pela geração de relatórios relativos às tabelas da malha viária. - PCI Calculation Subsystem: responsável pelo cálculo do PCI. - Data Importing Subsystem: responsável pela importação dos dados de arquivos externos gerados por aplicações externas.

O subsistema de simulação, indicado para ser utilizado inicialmente sobre a plataforma de grid, é responsável pela geração do relatório de comportamento dos pavimentos, simulando a sua degradação ao longo do tempo. A simulação é gerada para cada pista, de acordo com o tipo de pavimento dominante nela. O

algoritmo responsável pela geração desta simulação é ilustrado no Algoritmo 1.

---

**Algoritmo 1: Paviment Behavior Report Algorithm**

---

```
1.1 retrieve segments
1.2 while there are segments to process do
1.3   retrieve tracks
1.4   while there are tracks to process do
1.5     retrieve ICP
1.6     retrieve lanes
1.7     while there are lanes to process do
1.8       identify predominant pavement
1.9     endw
1.10    calculate track degradation
1.11  endw
1.12 endw
1.13 generate report
```

---

Os autores do trabalho realizaram algumas simulações de degradação de pavimento com o sistema. Os tempos de execução foram capturados e analisados. Para um teste envolvendo apenas um bairro de Porto Alegre, o tempo de resposta do sistema foi de 39 minutos. Com isto, chegou-se às seguintes conclusões: é impossível gerar a simulação para a cidade inteira e o sistema está fortemente amarrado ao banco de dados. Esta amarração foi gerada devido à arquitetura da aplicação.

### 3 OurGrid

O OurGrid [1] é um grid grátis de máquinas para execução de aplicações paralelas. Ele está em produção desde Dezembro/2004. Este grid é de fácil adesão e uso.

Para uso do Gerpav no ambiente de grid do OurGrid foi necessário, em primeiro lugar, a criação do Job Description File (JDF) para a submissão de uma *job* para execução. Este componente contém os detalhes de cada tarefa que será executada.

O MyGrid é uma API para facilitar o processo de criação do JDF. Porém, o seu uso não é trivial para aplicações baseadas em ambiente web, como é o caso do Gerpav. Então, para o uso da API MyGrid no Gerpav foi necessário a criação de uma camada de abstração extra, denominada OurGrid Job Abstraction Layer (OJAL). Esta camada é responsável pelo suporte a criação automática dos scripts do OurGrid e abstrai os detalhes de serialização e deserialização dos dados, divisão dos dados e submissão/espera dos *jobs*. A Figura 2 ilustra a arquitetura criada.

### 4 GerpavGrid

A maior motivação da utilização do Gerpav em uma grade computacional é o auxílio no desempenho de geração de relatórios de Degradação de Pavimentos. Visto que a aplicação utiliza muito o banco de dados e precisa fazer muitos cálculos para a geração de tais relatórios a parte da aplicação que foi escolhida para fazer a paralelização foi a de simulação de degradação. A partir

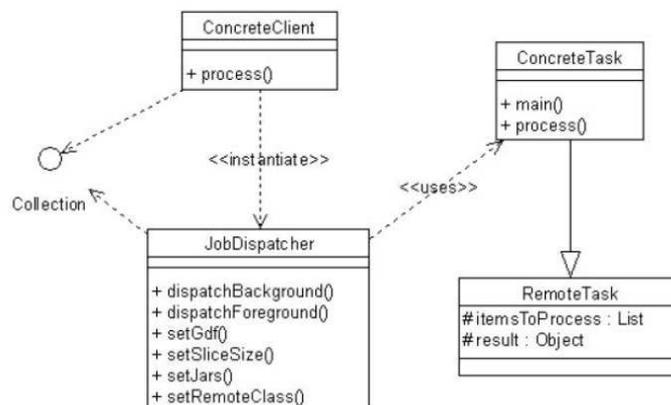


Figura 2: Diagrama de classes do OJAL

dessa paralelização e com resultados que atenderam as necessidades da aplicação anterior, pode-se gerar novos modelos de relatórios com informações mais objetivas. O GerpavGrid foi desenvolvido como uma extensão a aplicação Gerpav adicionando novas funcionalidades e melhores performances.

Algumas partes da aplicação Gerpav tiveram que ser reescritas para que se adequem as modificações propostas e exigidas pela paralelização.

A paralelização do processo de simulação e geração dos relatórios foi dividida em três partes: preparação dos dados, processamento e consolidação dos resultados. A fase de processamento foi dividida em tarefas enquanto as outras duas fases são executadas localmente. Para a fase que foi paralelizada foram implementadas algumas estratégias na otimização.

#### 4.1 Estratégias de Otimização

Foram implementadas quatro estratégias para solucionar os problemas de a aplicação estar amplamente enlaçada ao banco de dados.

- **Filtro no Banco de Dados ou em Memória:** Entre essas duas estratégias a principal diferença é que quando fizemos filtro no banco de dados enviamos consultas complexas e o banco de dados é responsável pelo processamento devolvendo um conjunto pequeno de dados, já no segundo são enviadas consultas simples, ao qual o banco de dados não faz muito processamento, e devolve um conjunto grande de dados que serão selecionados quando já estiverem na memória local.
- **Pequenas Fatias:** Nesta estratégia, o sistema faz requisições ao banco de dados de todas as informações sobre as ruas através de consultas complexas. Esses dados são divididos e distribuídos na grade como tarefas para execução. Depois da execução os dados retornam para a consolidação e geração do relatório.
- **Pipeline:** Neste modelo o sistema faz a requisição dos dados para o banco de dados e divide as ruas por bairros. O conjunto de dados do bairro é dividido em tarefas e enviado para a grade para execução. Ao final do

envio, o sistema já faz a requisição de informações de ruas de um outro bairro para executar os mesmos procedimentos sem aguardar o termino do primeiro bairro. O sistema aguarda todos as tarefas terminarem e faz a consolidação dos dados.

- **Base de Dados Distribuída:** Neste modelo os dados das tarefas enviadas para a grade são copiados para o *peer* responsável por tais tarefas agilizando o processo de acesso ao dados. As tarefas fazem a requisição dos seus dados que estão localizados em um ponto ao qual o acesso da rede é de alta velocidade. Depois de processados os dados o sistema retorna os resultados para a consolidação.

## 4.2 Funcionalidades Adicionais

Com o auxilio das estratégias propostas na paralelização da aplicação, foi possível a implementação de novas funcionalidades que proporcionaram excelentes resultados.

- **Distribuição de Orçamento:** A partir de simulações e informações sobre o índice de degradação dos pavimentos, pode-se desenvolver um módulo para o programa que tem o objetivo de informar, a partir de dados da aplicação, onde e como devem ser aplicado o orçamento que a secretaria dispõem para manutenção de pavimentos.
- **Intervenção Otimizada:** Esta funcionalidade oferece informações de qual o período ao longo dos anos ao qual devem ser feitas as manutenção e em quais vias estas devem acontecer. Com estas informações a secretaria pode distribuir melhor o seu orçamento planejado para um determinado período de forma a aproveitar o máximo do custo/benefício das manutenções da malha viária.

## 5 Avaliações

Nesta seção serão apresentados comparativos de performance que o artigo citado apresenta. Todas as avaliações são sobre a simulação e geração do relatório de degradação de pavimentos para os próximos quinze anos para três diferentes cenários: um bairro de tamanho médio, um conjunto de 50 bairros e um cenário incluindo toda a cidade de Porto Alegre. O número de segmentos e ruas que cada um dos cenários possui são apresentados na Tabela 1.

Cenários	Arco	Pista	Faixa
Aberta dos Morros	897	929	2094
50 Bairros	15070	15816	49195
Cidade Porto Alegre	31252	32408	98223

Tabela 1: Cenários

Os testes foram executados na grade OurGrid e o software GerpavGrid foi instalado em um Dual Xeon 3.0GHz 2GB RAM utilizando SuSE Linux 9.2. Para cada um dos cenários também foram utilizadas as implementações apresentadas no capítulo anterior e identificadas na Tabela 2.

Versão	Estratégia de Otimização	Tipo
I1	-	Seqüencial
I2	Filtro no BD + Fatias Pequenas	Paralela
I3	Filtro no BD + Pipeline	Paralela
I4	Filtro na Memória + Pipeline	Paralela
I5	BD Distribuído + Filtro em Memória	Paralela

Tabela 2: Estratégias de Otimização

Podemos observar que as técnicas de implementação utilizando filtro em memória (I4 e I5) obtiveram melhores desempenhos do que os outros modelos em todos os cenários. Isto ocorre pelo fato de que fazendo consultas complexas ao banco de dados concentra-se muito processamento na máquina que possui o banco e acaba que o processamento não se distribui. Através de consultas simples o processamento das informações se distribui possibilitando que mais máquinas acessem a base e processem mais dados remotamente. A Tabela 3 demonstra um comparativo de tempo dos cenários implementados.

	Aberta dos Morros	50 Bairros	Porto Alegre
I1	200.287	3551.650	7266.923
I2	207.818	3370.382	7430.041
I3	210.408	3489.814	7113.705
I4	32.060	275.864	506.271
I5	12.312	50.327	89.877

Tabela 3: Tempos de Execução em segundos

A tabela nos mostra que obtivemos um ganho de até 80 vezes comparando a versão seqüencial com a paralela. Isso possibilitou o desenvolvimento de novas funcionalidades, como as apresentadas na seção anterior, e mais agilidade para a geração de relatórios.

## 6 Conclusões

O trabalho apresentado neste artigo demonstra uma aplicação seqüencial que foi implementada de forma paralela sobre uma grade computacional. A partir de estudos e técnicas de paralelização pode-se desenvolver não toda a aplicação mas a parte ao qual tinha um perfil para ser paralelizada. O seu êxito foi demonstrado nas execuções dos cenários apresentados neste artigo. Além da performance, pode-se desenvolver novas funcionalidades que seriam inviáveis em uma versão seqüencial pelo nível de complexidade que as execuções exigiam.

## Referências

- [1] W. Cirne, F. Brasileiro, N. Andrade, L. Costa, A. Andrade, R. Novaes, and M. Mowbray. Labs of the world, unite!!! 2006.
- [2] César A. F. De Rose, Tiago C. Ferreto, Marcelo Farias, Vladimir Dias, Walfredo Cirne, Milena Oliveira, Katia Saikoski, and Maria Luiza Danieleski. Gerpavid: using the grid to maintain the city road system. 2006.