

## 5 Conclusão

Neste trabalho, foi apresentada uma arquitetura para servidores de dados estruturados em aplicações científicas. O objetivo dessa arquitetura é permitir que uma aplicação científica possa compartilhar os seus dados de forma estruturada e de maneira eficiente e, assim, permitir que estas aplicações sejam integradas com outras aplicações desta natureza.

A arquitetura apresentada foi desenvolvida através de uma abordagem de componentes de *software* utilizando o modelo de componentes distribuídos SCS. Uma aplicação oferece os seus dados através de um servidor que é um componente de *software*. Esse servidor pode oferecer uma ou mais facetas que representam serviços de dados.

É através de um serviço de dados que uma aplicação pode navegar pelos dados oferecidos, obtê-los, removê-los ou solicitar a criação de um novo dado. A arquitetura oferece uma navegação hierárquica sobre os metadados dos dados oferecidos por um serviço. Quando o dado desejado é alcançado, a aplicação solicita a transferência de uma das visões disponíveis para esse dado. Esse mecanismo de múltiplas visões pode ser utilizado, ainda, para o versionamento das visões, visto que uma visão pode ser uma nova versão de uma outra. As visões, assim como a navegação hierárquica, podem ser utilizadas também para permitir que as aplicações realizem a obtenção parcial de um dado.

A arquitetura foi implementada em dois estudos de caso utilizando aplicações científicas desenvolvidas pelo Tecgraf/PUC-Rio em convênio com a Petrobras. Em um dos estudos de caso foi apresentado o uso de múltiplas visões sobre um mesmo dado, de acordo com os usuários e aplicações que fazem uso desse dado. No outro, havia a necessidade da transferência dos dados de maneira não-estruturada e a flexibilidade da arquitetura permitiu que se criasse uma visão para uma transferência dessa natureza.

Os experimentos realizados demonstraram que a arquitetura pode ser utilizada com diversas tecnologias visando uma transferência eficiente das visões dos dados. O uso de *valuetypes* foi avaliado e serviu como base para a comparação de desempenho com os demais experimentos. Em seguida, o impacto das interfaces abstratas foi medido e, como esse impacto foi praticamente im-

perceptível, esse mecanismo foi utilizado na arquitetura, dando mais flexibilidade para os servidores representarem as suas visões dos dados. O *custom marshal* dos *valuetypes* também foi avaliado através de experimentos e concluiu-se que esse mecanismo deve ser evitado.

Houve ainda uma avaliação de uma tecnologia utilizada pela indústria, o *Protocol Buffers* do Google. A expectativa era verificar se o seu uso seria mais indicado do que *valuetypes* de CORBA, visto que essa tecnologia se intitulava bem mais rápida do que XML, por exemplo. Entretanto, os *valuetypes* apresentaram um desempenho bastante superior à tecnologia do Google. O *Protocol Buffers* foi experimentado também em conjunto com os *valuetypes*, sendo usado para gerar uma representação compacta das visões transferidas através do mecanismo de *custom marshal*.

## 5.1

### Trabalhos Futuros

A arquitetura apresentada neste trabalho já está sendo utilizada em aplicações científicas desenvolvidas pelo Tecgraf/PUC-Rio em convênio com a Petrobras, como foi salientado nos estudos de caso. A implementação da arquitetura em tais aplicações serviu como um sinalizador de que possíveis adições seriam necessárias para um melhor uso da arquitetura. Há ainda o planejamento para a criação de alguns novos servidores de dados e estes também já seriam melhor implementados se algumas características novas fossem incluídas na arquitetura.

Para algumas aplicações a navegação hierárquica não é natural. Uma possibilidade seria a criação de uma navegação através de uma interface de consultas (*queries*). Com uma interface dessa natureza, as aplicações descreveriam uma consulta em alguma linguagem específica, como por exemplo SQL, e o serviço de dados informaria os metadados de todos os dados que se encaixassem no perfil solicitado.

Os estudos de caso e os experimentos realizados com a arquitetura foram todos implementados na linguagem de programação Java. Existem aplicações científicas candidatas a utilizarem essa arquitetura que estão escritas em linguagens de programação diferentes, como C++ e Lua. Portanto, existe a necessidade de se fazer uma avaliação da arquitetura em implementações com essas linguagens de programação.

Há ainda a necessidade de se avaliar novas tecnologias para a transferência eficiente das visões. A criação de novos tipos de *stream* é uma outra possibilidade, ao invés de se utilizar os *streams* padrão de CORBA.