1
Introdução

A integração de aplicações, em especial a de aplicações do domínio corporativo (*enterprise application integration* [1]), tem recebido bastante atenção por parte de diferentes áreas de pesquisa em computação e da indústria de *software*. As corporações possuem dezenas e, até mesmo centenas, de aplicações e a indústria de *software* tem conseguido endereçar a necessidade de integrá-las, oferecendo diversos tipos de solução. Dentre essas soluções podemos destacar os vários tipos de *middleware*, em especial os que servem de apoio ao compartilhamento de dados e os barramentos de serviços que servem de suporte à implementação da popular arquitetura orientada a serviços (*service oriented architecture* [2]). Além dessas soluções prontas oferecidas pela indústria, existem também os padrões de projeto para a construção de soluções de integração [3], que servem de guia para o desenvolvimento deste tipo de solução própria por parte das corporações.

O foco principal dessas soluções é, tipicamente, o domínio corporativo e, por causa disso, a integração de aplicações de outros domínios pode não ocorrer de maneira satisfatória ou, até mesmo, ser inviável. Em particular, a integração de aplicações científicas ou de alto desempenho possui requisitos distintos em relação à integração de aplicações corporativas, o que inviabiliza em certos casos a adoção das soluções citadas anteriormente. Nestas situações, a criação de soluções próprias acaba sendo a única opção.

As aplicações científicas têm como característica a elevada complexidade de seus dados e dos algoritmos usados para processar esses dados [4, 5, 6]. Além da representação complexa do dado, é bastante comum que as aplicações científicas trabalhem com grandes volumes de dados, tornando o processamento uma tarefa muito demorada. Essas características dificultam bastante o compartilhamento de dados entre essas aplicações.

Este trabalho apresenta uma arquitetura para servidores de dados estruturados científicos, através da qual as aplicações científicas podem compartilhar seus dados de maneira eficiente. A arquitetura foi concebida para permitir que um dado seja oferecido através de mais de uma visão estruturada, possibilitando assim que as aplicações usem o dado da maneira mais adequada ao seu

domínio; uma aplicação obtém a visão mais compatível com os algoritmos que usará para processar o dado. Além da arquitetura proposta, este trabalho apresenta uma avaliação de diferentes mecanismos para a representação de dados de maneira estruturada e também de diferentes tecnologias para o empacotamento e o desempacotamento destes dados que permitam uma transferência eficiente dos mesmos.

A arquitetura aqui apresentada oferece às aplicações uma navegação hierárquica pelos dados científicos ofertados pelos servidores de dados. Esta forma de representar os dados científicos é bastante comum, como será visto adiante no capítulo sobre trabalhos relacionados. O HDF (*Hierarchical Data Format*) [7], por exemplo, oferece um tipo de dado que seria a raiz da hierarquia e a partir deste dado é que são obtidos os seus elementos descendentes. A forma de navegação na arquitetura dos servidores de dados é análoga: inicia-se o processo obtendo-se as raízes da hierarquia e, posteriormente, são obtidos os elementos descendentes dos dados até que se chegue ao dado de interesse. Este modelo de navegação é oferecido para a obtenção tanto dos dados científicos quanto de seus metadados.

Os dados oferecidos pelos servidores podem ser representados por mais de uma visão, o que permite que cada aplicação cliente utilize a visão do dado mais adequada ao seu uso. A capacidade de utilizar múltiplas visões de um mesmo dado também pode ser utilizada para criar versões diferentes de uma mesma visão. A arquitetura permite ainda a obtenção parcial dos atributos de um dado, tanto através do mecanismo de múltiplas visões como através da própria navegação hierárquica.

As visões utilizadas pelos servidores de dados não são apenas estruturas com um conjunto de atributos, como uma *struct* na linguagem C. Estas visões são, na verdade, objetos que possuem não somente atributos, mas também oferecem operações que atuam sobre esses atributos, permitindo um uso mais sofisticado do dado.

A transferência dos dados, propriamente dita, precisa ser realizada da maneira mais eficiente possível. Por esse motivo, foram investigadas diferentes maneiras de se representar o dado em uma forma binária própria para a transferência através de uma rede. Além disso, também foram investigadas diferentes formas de se realizar o empacotamento e o desempacotamento dos dados, convertendo-os de seu formato estruturado para este formato binário e vice-versa.

Está fora do escopo deste trabalho a atualização e a sincronização dos dados. Não são oferecidas operações para se alterar um dado e nem para a notificação de mudanças em um dado, caso este seja alterado de alguma

forma. Além disso, a arquitetura proposta não oferece nenhuma interface de consulta para obtenção dos dados, apenas oferece a navegação hierárquica, citada anteriormente. Esta funcionalidades, quando necessárias, ficarão a cargo das implementações dos servidores de dados.

Como o foco principal da arquitetura de serviços de dados é o suporte à integração, em um dos estudos de caso esta arquitetura será avaliada em conjunto com um barramento para integração de aplicações científicas denominado OpenBus. O OpenBus foi construído pelo Tecgraf/PUC-Rio em um convênio com a Petrobras para dar suporte à integração de aplicações científicas através de uma arquitetura orientada a serviços (SOA). A integração das aplicações científicas que utilizam o barramento consiste, na maioria dos casos, na transferência de dados entre as mesmas e é nesse contexto que a arquitetura de serviços de dados proposta neste trabalho será utilizada.

A contribuição principal deste trabalho é a arquitetura propriamente dita, baseada em componentes de *software*, extensível e flexível, possibilitando o seu uso em diferentes domínios de aplicação. Para a criação desta arquitetura foram pesquisados e avaliados diferentes mecanismos para a utilização de objetos que representam dados sobre o *middleware* CORBA [8] e também diferentes tecnologias para o empacotamento e o desempacotamento desses dados. Além da arquitetura, uma outra contribuição desse trabalho é a análise do desempenho dos mecanismos e tecnologias pesquisados.

No próximo capítulo serão apresentados os trabalhos relacionados ao tema deste trabalho. Em seguida, no capítulo 3, a arquitetura desenvolvida é apresentada junto com a sua implementação. No capítulo 4 são apresentados os estudos de caso, que incluem uma análise de desempenho. Por fim, no capítulo 5, são apresentadas as conclusões do trabalho.