Uma proposta para distribuição e execução de *wrappers* em um ambiente de Grid para o CoDIMS

Cristiano Biancardi , Leonardo Jose Silvestre , Alvaro Cesar Pereira Barbosa

[1]Laboratório de Pesquisa em Redes e Multimídia - LPRM
Departamento de Informática
Universidade Federal do Espírito Santo
Campus de Goiabeiras
Av. Fernando Ferrari, S/N 29060-970 Vitória, ES

{cbiancardi,lsilvestre,alvaro}@inf.ufes.br

Abstract. *Grid is a computational environment in which applications can use multiples distributed computational resources in a safe, coordinated, efficient and transparent way. Besides, data integration systems are distributed and they can make use of Grid environments to obtain a better performance and a rational use of available resources. This work describes a proposal for the distribution, allocation and execution of wrappers in CoDIMS data integration system using a Grid environment.*

Resumo. *Grid é um ambiente de computação no qual aplicações podem utilizar múltiplos recursos computacionais distribuídos de forma segura, coordenada, eficiente e transparente. Por sua vez, sistemas de integração de dados são por natureza distribuídos e podem se beneficiar de ambientes Grid para melhor desempenho e uso racional dos recursos disponíveis. Este trabalho descreve uma proposta para a distribuição, alocação e execução dos wrappers do sistema de integração de dados CoDIMS, utilizando um ambiente de Grid.*

1. Introdução

Nos dias de hoje, a quantidade de dados disponível vem aumentando consideravelmente. Com o advento da *Web*, qualquer indivíduo ou organização pode se tornar um fornecedor de informação sem requerer autorização. Uma grande quantidade de informações e serviços, heterogêneos e distribuídos, está prontamente disponível, e os usuários, cada vez mais, necessitam de uma visão integrada dos dados disponíveis a partir dessas fontes. Na tentativa de fornecer uma solução para esse problema, sistemas de integração de dados vêm sendo desenvolvidos [Biancardi et al., 2004]. A integração de dados de múltiplas fontes é um problema que vem demandando pesquisa na comunidade de Banco de Dados por mais de uma década [Sheth and Larson, 1990, Halevy, 2003], tendo recebido, nos últimos anos, um impulso significativo.

Diversas nomenclaturas têm sido usadas para referenciar sistemas de integração, tendo ultimamente convergido para sistemas *middleware* de integração de dados. Tais sistemas visam a fornecer uma única visão, uniforme e homogênea, dos recursos (dados e ambientes) distribuídos e heterogêneos [Barbosa, 2001] (figura 1).

Para ser possível proceder a integração de dados heterogêneos, torna-se necessário transformá-los para um único modelo de dados, chamado modelo global ou canônico. Por exemplo, para se integrar três fontes, sendo a primeira XML, a segunda Relacional e a terceira Orientada a Objetos, assumindo o modelo global com sendo Relacional, deve-se

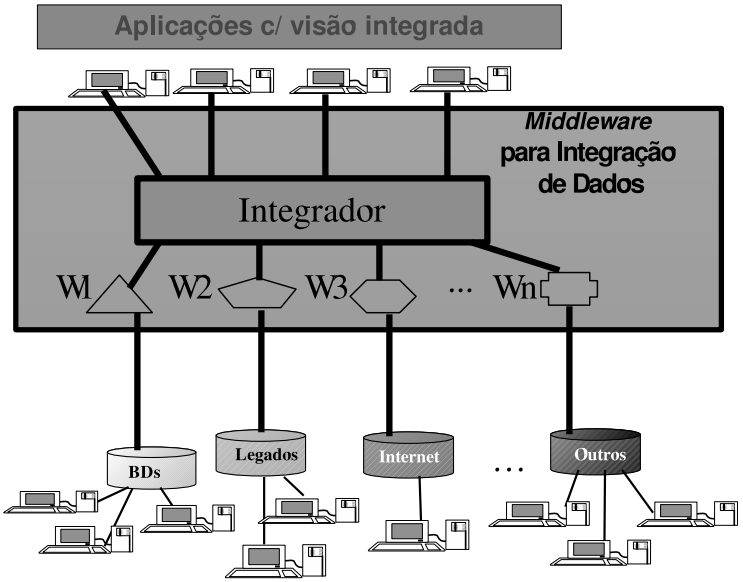

Figura 1: *Middleware* para Integração de Dados.

primeiro converter de XML e OO para Relacional para ser possível realizar a integração (figura 2). Tal conversão é realizada pelos *wrappers*. Além disso, eles são utilizados para prover a comunicação com as fontes de dados. Assim, os *wrapper* são de fundamental importância para os sistemas *middleware* de integração de dados.

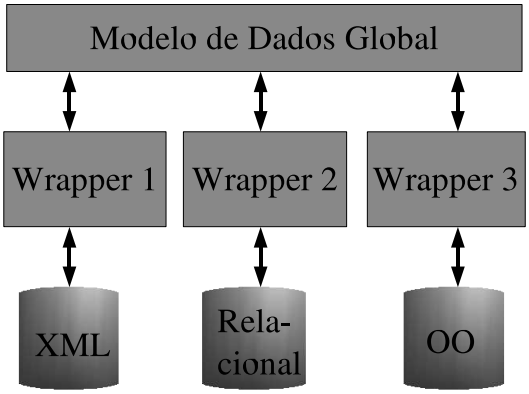

Figura 2: *Wrappers* em um Sistema Integrador.

Como o ambiente de integração é por natureza distribuído, sistemas de integração de dados podem se beneficiar de um ambiente Grid, no que se refere a alocação de *wrappers*, para um melhor desempenho do sistema como um todo, . Grid [Foster and Kesselman, 1999], é um ambiente de computação no qual as aplicações podem utilizar múltiplos recursos computacionais distribuídos geograficamente de forma segura, coordenada, eficaz e transparente. Além disso, Grid tem recebido atenção especial da comunidade científica. No que diz respeito a banco de dados, estão sendo propostos padrões para o desenvolvimento de serviços Grid de banco de dados, focados principalmente em prover acesso consistente a bancos de dados existentes, gerenciados de maneira autônoma [GridForum, 2004]. Segundo [Abiteboul et al., 2003], ”é importante explorar oportunidades para combinar novas tecnologias com banco de dados, de forma a incrementar o uso da informação”.

1.1. CoDIMS

O CoDIMS (*Configurable Data Integration Middleware System*) [Barbosa, 2001, Barbosa et al., 2002, Trevisol, 2004] é um *middleware* para integração de dados baseado

no conceito de *framework*, cuja principal característica é o fato de ser flexível e configurável, podendo ser adaptado para ser utilizado em diversos domínios de aplicação que exigem diferentes tipos de serviços de integração. Um exemplo de adaptação é o CoDIMS-G [Fontes et al., 2004], que é uma instância do *framework* CoDIMS que vem sendo desenvolvida para suportar aplicações visuais (científicas) executando em um ambiente de Grid.

Na implementação original do CoDIMS, todos os seus componentes, inclusive os *wrappers*, estão instalados no mesmo servidor. Isso acarreta sobrecarga do servidor, gerando lentidão na sua execução. Dessa forma, no caso dos *wrappers*, mostra-se importante distribuí-los em outras máquinas para: 1) reduzir a carga do servidor CoDIMS; 2) executar em paralelo as sub-consultas a serem encaminhadas a cada fonte de dados; 3) armazenar o conjunto resultado proveniente da execução de cada sub-consulta, permitindo o uso de transações distribuídas [Özsu and Valduriez, 2001] durante a execução do Plano de Execução de Consultas (PEC) [Pinheiro, 2004]. Assim, como o desenvolvimento de serviços Grid [Foster et al., 2002] fornece uma transparência para as aplicações de usuários no que diz respeito a heterogeneidade do ambiente, nada mais natural do que usufruir de um ambiente Grid para melhor distribuir o processamento dos *wrappers* do CoDIMS pelos recursos disponíveis em tal ambiente.

2. Trabalhos Relacionados

Na literatura são encontradas diversas propostas e sistemas voltados para o problema de integração de dados heterogêneos e distribuídos. Dentre eles, e mais relacionados a este trabalho, destacam-se: MOCHA [Rodriguez-Martinez and Roussopoulos, 2000], OGSA-DAI [Atkinson et al., 2002], OGSA-DQP [Alpdemir et al., 2003], SkyQuery [Malik et al., 2003] e CoDIMS-G [Fontes et al., 2004].

No MOCHA, os *wrappers* (DAP - *Data Access Provider*) são previamente instalados em máquinas próximas às fontes de dados ou nas próprias fontes, não suportando efetuar uma realocação de DAP, em tempo de execução, caso ocorra sobrecarga de alguma máquina onde existe um DAP instalado.

O OGSA-DAI, que pode ser visto como um *wrapper* e não como um sistema integrados de dados completo, não possui a funcionalidade de realocação de instâncias GDS (*Grid Database Service*), em tempo de execução, em outros nós de um Grid onde exitem OGDA-DAIs instalados.

No OGSA-DPQ, nenhuma integração de esquema e resolução de conflitos é suportada durante a importação dos esquemas: os mesmos são simplesmente acumulados e mantidos localmente. Além disso, o plano de execução de consulta gerado pelo mesmo é estático, não permitindo a realocação de seus operadores situados em nós sobrecarregados.

No projeto Skyquery, que fornece uma arquitetura baseada em *wrapper-mediator*, os *Web Services* são implantados em cada banco de dados, sendo a principal interface de comunicação e encapsulamento tanto para o processador de consulta quanto para o tradutor de fontes de dados. Contudo, este projeto adota uma abordagem de processamento de consulta centralizado.

O CoDIMS-G usa o ambiente de Grid para a execução em paralelo de programas sobre uma grande base de dados. Este não tem o objetivo de ser uma extensão do CoDIMS, de forma a adaptá-lo de maneira mais geral, ao ambiente de Grid para efetuar integração de dados heterogêneos, mas sim usar Grid para a solução de um problema específico.

De maneira diferente, esta proposta visa a adicionar uma sistemática ao CoDIMS de forma que as instâncias de seus *wrappers* possam ser alocadas dinamicamente para serem executadas em nós de um Grid, aproveitando os recursos disponíveis e distribuídos em tal ambiente. Com isso, pode-se ter uma execução distribuída e paralela do PEC possibilitando um melhor desempenho do sistema.

3. Contribuições Esperadas

Este trabalho apresenta uma proposta para estender o CoDIMS com um mecanismo de distribuição, alocação e execução de seus wrappers em nós de Grid, através da incorporação de uma nova camada denominada Wrapper-Grid. A partir desta nova abordagem, espera-se os seguintes resultados para o projeto CoDIMS: primeiro, possibilitar a execução paralela dos wrappers; segundo, alocar/realocar instâncias destes wrappers em nós de um Grid visando a um melhor uso dos recursos computacionais disponíveis; terceiro, disponibilizar os conjuntos resultados (dados homogêneos) oriundos das fontes de dados em um ambiente distribuído proporcionando a execução distribuída e PEC.

4. CoDIMS para Grid

O CoDIMS para ambiente de Grid possui uma arquitetura de quatro camadas, que são: Aplicação, Integração, *Wrapper*-Grid e Fontes de Dados (figura 3).

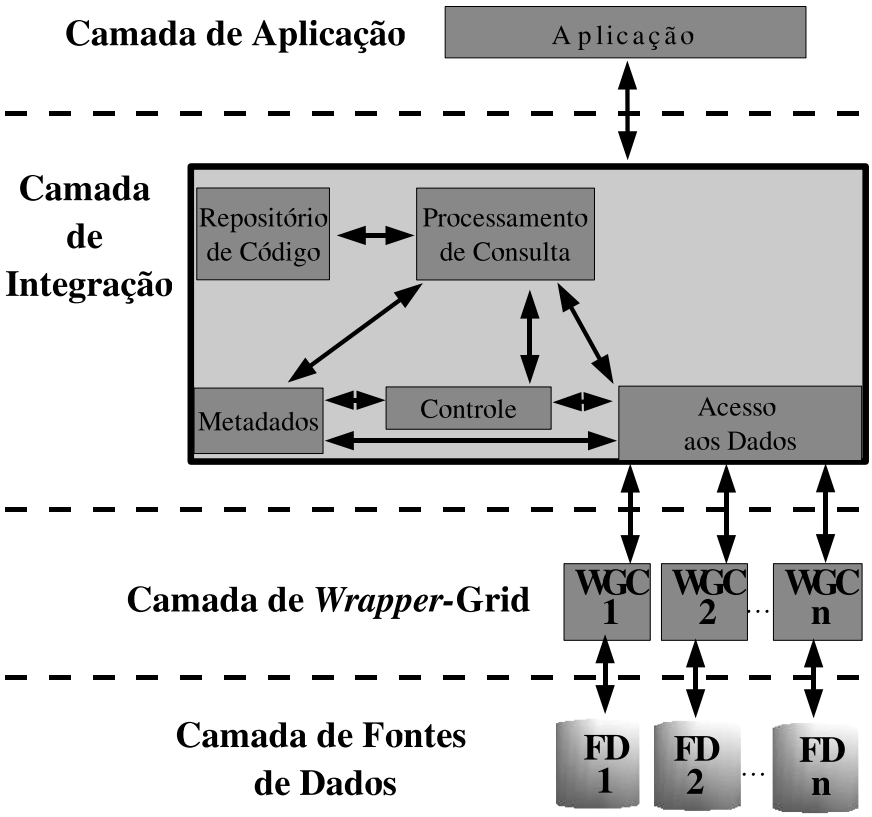

Figura 3: Arquitetura do CoDIMS em um ambiente de Grid.

A camada de *Wrapper*-Grid é formada pelos *Wrappers Grid Components* (WGCs), sendo que cada WGC está instalado em um nó específico do Grid. WGC é baseado em serviços Grid que encapsulam as funcionalidades de *wrappers*, com o intuito de assistir na execução de sub-consultas em cada fonte de dados. Na base deste componente é usado o Globus toolkit [Globus, 2004], que é uma implementação de referência

OGSA [Foster and Gannon, 2003]. Este componente estende os serviços Grid para suportar as seguintes funcionalidades dentro do cenário de integração de dados: a) criar e publicar os serviços de integração (*Create Publish Service* - CPS) oferecidos pelo componente; b) encapsular as funcionalidades básicas de um *wrapper* (comunicação com a fonte de dados e tradução entre modelos de dados) em um "serviço *wrapper*"(*Wrapper Service* - WS); c) enviar uma instância de um serviço (*Send Service* - SS) criado por um componente WGC, mais precisamente um serviço *wrapper*, para um outro componente WGC no Grid; d) receber e executar uma instância de serviço (*Received Execution Service* - RES) enviada por um componente WGC diferente; e) notificar o sistema integrador quando uma consulta for finalizada (*Notify Query End* -NQE). Outros serviços podem ser necessários para aplicações específicas.

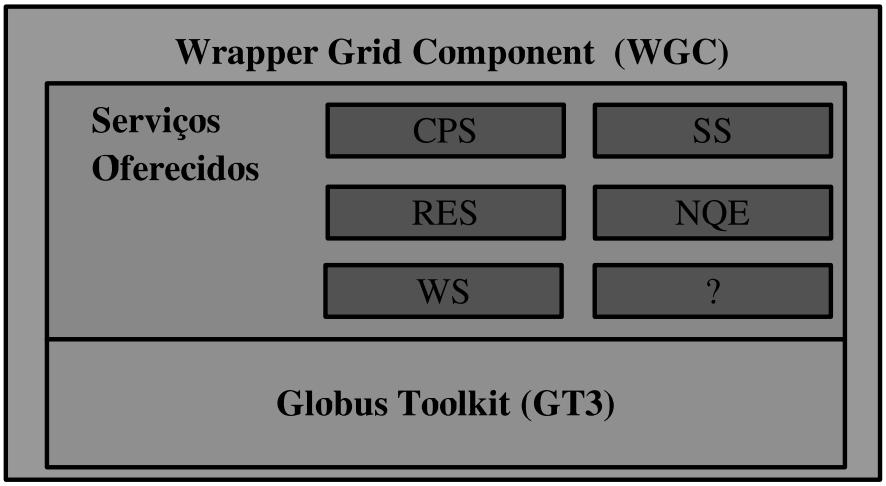

Figura 4: Wrapper Grid Component (WGC).

Na configuração inicial de uma instância do CoDIMS, cada WGC é alocado em um servidor específico (nó do Grid), sendo que para cada fonte de dados distinta existe um WGC associado (figura 3).

Quando do uso do sistema, o usuário submete uma consulta, através da camada de Aplicação, que é passada para a camada de Integração. Nesta, o componente Processamento de Consulta a transforma em um PEC, incluindo as sub-consultas a serem processadas em cada fonte de dados. O PEC é encaminhado para o componente Acesso aos Dados que, para cada sub-consulta, realiza duas atividades: primeiro, faz uma análise da capacidade ociosa dos nós do Grid para decidir em qual nó do Grid uma instância WS a ser criada por um WGC será executada: no próprio nó ou em outro. Segundo, interage com o componente WGC responsável pela fonte de dados a ser consultada informando em que nó tal instância WS deverá ser executada. Isso é possível pois, uma instâcia WS de um WGC pode ser executada por qualquer outro WGC no Grid. Cada instância WS estabelece, então, uma conexão com a fonte de dados respectiva, repassando a sub-consulta que a mesma terá que processar. O resultado desse processamento é retornado para esta instância WS, sendo convertido do modelo nativo da fonte para o modelo de dados global. Este resultado é armazenado no nó onde se situa a instância WS, devendo o componente WGC deste nó, notificar o componente Acesso aos Dados da localização (endereço do nó e diretório) do conjunto resultado da sub-consulta. Esta informação é armazenada em uma estrutura de dados, onde existe uma associação entre consulta e sub-consultas com a respectiva localização do conjunto resultado. Tal sistemática garante que os conjuntos resultado de cada sub-consulta ficam armazenados em diferentes nós do Grid, permitindo assim, a continuidade da execução do PEC de forma distribuída e em paralela das suas operações internas.

5. Conclusões e Trabalhos Futuros

Computação Grid tem se tornado muito popular e muitos projetos tem sido iniciados para suportar, em parte, a visão de um Grid [Foster and Kesselman, 1999]. Com isso, a computação Grid tem emergido como um importante novo campo, distinto da computação distribuída convencional, dado que o seu foco está no compartilhamento de recursos e aplicações inovadoras. Neste contexto, sistemas como Globus toolkit estão sendo desenvolvidos para fornecer um conjunto de serviços grid básicos seguindo o padrão *OGSI*. Para suportar aplicações em integração de dados, cujos dados são essecialmente heterogêneos e distribuídos, outros tipos de serviços, tais como aqueles característicos de um *wrapper*, podem ser adaptados para o ambiente Grid. Neste artigo foi apresentada uma camada de *Wrapper*-Grid na qual componentes WGCs estão instalados em nós de um Grid. Também foram apresentadas as funcionalidades que devem ser desempenhadas pelo componemte Acesso aos Dados para que seja possível utilizar os WGCs presentes na camada de *Wrapper*-Grid.

Esta proposta difere-se dos demais trabalhos relacionados nos seguintes aspectos: apesar de o sistema integrador possuir uma configuração estática inicial com relação aos WGCs instalados na camada de *Wrapper*-Grid, instâncias de serviços *wrapper* (WS) dos mesmos poderão ser alocadas em qualquer nó do Grid onde existe este componente instalado. Existe um componente WGC para cada tipo distinto de *wrapper*, que será capaz de criar, executar e enviar instâncias de *wrappers*, além de receber instâncias criadas por um WGC diferente e ser capaz de executar a mesma. Tudo isso objetivando um melhor desempenho do sistema de integração de dados como um todo, aproveitando-se dos benefícios do Grid.

Esta proposta contribui com o projeto CoDIMS por possibilitar a execução paralela dos wrappers; alocação/realocação de instâncias destes em nós de um Grid; disponibilização dos conjuntos resultados oriundos das fontes de dados em um ambiente distribuído, proporcionando a execução distribuída do PEC, objetivando um melhor desempenho. O seu estágio de implementação ainda é inicial com relação ao WGC, sendo desenvolvido no ambiente Linux e usando a linguagem Java.

Devido à necessidade de realizar uma realocação de instâncias WSs em tempo de execução, torna-se interessante em trabalhos futuros que o componente Acesso aos Dados possua a função de monitorar a vazão de cada sub-consulta. Quando identificada a necessidade de realocação, uma nova instância será criada para ser executada em um outro nó com um WGC diferente. Contudo, a instância antiga poderá continuar executando, pois pode acontecer de, apesar de a nova instância estar executando em um nó menos sobrecarregado, a instância antiga finalizar a sua execução antes da nova. Aqui surge a necessidade de abortar a execução de uma instância caso o resultado já tenha sido gerado por outra. Dessa forma, o componente que gerar primeiro o resultado deverá notificar aqueles componentes que possua instâncias WSs identicas às da sub-consulta em questão.

Um outra funcionalidade a ser adicionada no CoDIMS para Grid seria a implantação automática de código específico de uma aplicação, realizado através do envio de classes Java para componentes WGCs. Para aquelas fontes com capacidade de processamento de consulta, classes Java poderão ser aplicadas nos conjuntos resultado, provenientes destas, com o intúito de efetuar a redução desses conjuntos. Para as fontes com capacidade parcial de execução de consulta, classes Java poderão também ser responsáveis por efeturar o processamento da sub-consulta. Todas as classes Java serão primeiro armazenadas em um Repositório de Código (ver figura 3), a partir do qual são posteriormente recuperados e implantados, de acordo com sua necessidade.

Referências

Abiteboul, S., Agrawal, R., Bernstein, P., and Carey, M. (2003). The lowell database research self assessment.

Alpdemir, N., Mukherjee, A., Paton, N. W., Fernandes, A. A., Gounaris, A., and Smith, J. (2003). Ogsa-dqp: A service-based distributed query processor for the grid. UK e-Science All Hands Meeting Nottingham. EPSRC.

Atkinson, M., Baxter, R., and Hong, N. C. (2002). Grid data access and integration in ogsa. EPCC, University of Edinburgh.

Barbosa, A. C. P. (2001). *Middleware para Integração de Dados Heterogêneos Baseado em Composição de Frameworks*. PhD thesis, PUC-Rio, Brasil.

Barbosa, A. C. P., Porto, F., and Melo, R. N. (2002). Configurable data integration middleware system. *J. Braz. Comp. Soc.*, 8(2):12–19.

Biancardi, C., Silvestre, L. J., and Barbosa, A. C. P. (2004). Integração de dados heterogêneos em ambiente web. ERI2004. Vitória(ES) , Rio da Ostra(RJ).

Fontes, V., Schulze, B., Dutra, M., Porto, F., and Barbosa, A. C. P. (2004). Codims-g: a data and program integration service for the grid. In *Proceedings of the 2nd workshop on Middleware for grid computing*, pages 29–34. ACM Press.

Foster, I. and Gannon, D. (2003). The open grid services architecture platform. Disponível em http://www.globalgridforum.org/Meetings/ggf7/drafts/draft-ggf-ogsa-platform-2.pdf.

Foster, I. and Kesselman, C. (1999). *The Grid: Blueprint for a New Computing Infrastructure*, chapter 11, pages 259–278. MORGAN-KAUFMANN.

Foster, I., Kesselman, C., Nick, J., and Tuecke, S. (2002). The physiology of the grid: An open grid services architecture for distributed system integration. *Open Grid Service Infrastructure WG, Global Grid Forum -*.

Globus (2004). The globus toolkit. Technical report, www.globus.org.

GridForum (2004). Database access and integration services wg - dais-wg. https://forge.gridforum.org/projects/dais-wg.

Halevy, A. Y. (2003). Data integration: A status report. In *Proceedings of 10th Conference on Database Systems for Business Technology and the Web (BTW 2003)*, pages 24–29, Germany.

Malik, T., Szalay, A. S., Budavari, T., and Thakar, A. (2003). Skyquery: A web service approach to federate databases. In *CIDR*.

Pinheiro, F. S. (2004). Incorporando uma máquina de execução de consultas ao codims. Monografia de Conclusão de Curso, Universidade Federal do Espírito Santo - UFES.

Rodriguez-Martinez, M. and Roussopoulos, N. (2000). Mocha: a self-extensible database middleware system for distributed data sources. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 213–224. ACM Press.

Sheth, A. P. and Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236.

Trevisol, G. G. (2004). Codims: Incorporando nova abordagem na comunicação entre seus componentes. Monografia de Conclusão de Curso, Universidade Federal do Espírito Santo - UFES.

Özsu, M. T. and Valduriez, P. (2001). *Princípios de Sistemas de Banco de Dados Distribuídos*. Editora Campus, Rio de Janeiro - RJ.