

# TFD 2004/2005 - Artigo 1

tfd07

Eduardo Sousa 25874

Francisco Temudo 25624

Gonçalo Marques 26531

## Abstract

*O sistema Appia fornece um conjunto de serviços de filiação e comunicação em grupo para a concretização de servidores replicados fiáveis. Para exercitar a sua utilização propõe-se a realização de um servidor replicado contendo um conjunto de registos armezando, cada um, um inteiro. Pretende-se uma implementação que suporte propagação melhor-esforço e propagação atómica de comandos de leitura e escrita nos registos*

## 1. Introdução

O aumento do número de utilizadores de sistemas em rede bem como a disseminação destes sistemas por várias áreas de actividade teve como consequência a necessidade de se criarem mecanismos que permitissem replicar a informação existente neste tipo de sistemas. Esta necessidade advém do facto de ser importante aumentar a tolerância a faltas por forma a colmatar as insuficiências próprias dos sistemas distribuído, muito propensos a falhas. A replicação permite manter um repositório fiável de informação ao diminuir o número de pontos de falha críticos no sistema distribuído.

Em sistemas replicados é importante definir funcionalidades que permitam a comunicação entre os vários repositórios constituíntes dum sis-

tema de forma a garantir a coerência e consistência da informação comum. O sistema Appia resolve este problema fornecendo vários protocolos para implementação de tais funcionalidades.

O sistema Appia é uma extensão de protocolos de comunicação conhecidos (TCP, UDP, etc.) que permite a utilização destes num contexto de informação replicada. O Appia tem um design modular o que permite que a plataforma de comunicação seja independente de qualquer componente aplicacional que se pretenda desenvolver sobre o Appia. Mas o carácter inovador do Appia não se relaciona com a sua modularidade, visto existirem sistemas semelhantes que possuem esta qualidade. A inovação consiste na possibilidade de se coordenarem múltiplos canais estabelecendo-se regras para as suas interacções.

Para se testarem essas capacidades, pretende-se desenvolver um sistema Appia para a gestão de acessos de leitura e escrita a registos num sistema replicado, de duas formas diferentes: melhor-esforço e atómica.

O acesso melhor-esforço consistem em escritas e leituras de registos simultaneamente à actualização das réplicas o que permite acessos a réplicas desactualizadas.

O acesso atómico distingue-se do anterior por obrigar a que cada leitura e escrita se faça sobre réplicas actualizadas, não permitindo, consequentemente, que se faça a actualização das réplicas simultâneamente a leituras ou escritas.

Este artigo contém a especificação da arquitectura Appia que se achou adequada para a implementação acima descrita e encontra-se organizado da seguinte forma: a primeira secção é a introdução, a segunda secção inclui um resumo estendido da aplicação e a terceira secção contém uma conclusão e algumas considerações à cerca do trabalho futuro.

## 2. Resumo Estendido

Como já foi referido nas secções anteriores, o uso de servidores replicados para tratamento e armazenamento de informação está em crescimento, quer por motivos de redução de carga na rede ou por motivos de aumento da disponibilidade de serviços. Desta forma, será também nessa vertente que se irá desenvolver este projecto, cuja descrição se apresenta nos capítulos seguintes. Pretende-se a realização de um servidor replicado, que tenha a capacidade de armazenar um conjunto de registos. Por sua vez cada um desses registos armazena um inteiro. Os clientes que terão acesso a este servidor, poderão realizar dois tipos de operações: Uma de escrita e uma de leitura. A opção de leitura consiste no acesso ao valor inteiro que o servidor contém nesse momento, num dado registo. Para a opção de escrita, o cliente indica qual o registo que quer alterar, e o valor a inserir no mesmo. O servidor ficará encarregado de transmitir a alteração pelas varias réplicas que o constituem. É de salientar que a replicação do servidor é totalmente transparente para o cliente, isto é, o cliente apenas visualiza o servidor como uma unica entidade, independentemente do número de réplicas que possam existir, ou das suas diferentes localizações geográficas. Para desenvolver este modelo, são contudo necessários alguns cuidados, como por exemplo o tratamento de situações como a falha ou reintegração de um dos servidores. Num breve exemplo, podemos observar que após a falha de um dos servidores, o servidor replicado tem que se manter activo, e ao mesmo tempo manter os

dados consistentes. Ou por outro lado, no caso da reintegração de um servidor no grupo, há que actualizar a informação do mesmo com a que consta no grupo. No que se refere aos acessos à informação contida no servidor, vão ser desenvolvidos dois métodos. O primeiro baseia-se num acesso livre aos registos do servidor. Qualquer cliente pode aceder aos dados do servidor, quer em modo de leitura ou em modo de escrita, sem qualquer tipo de restrição. Isto é um cliente pode escrever um valor num registo num dado instante, e por sua vez outro cliente pode ler esse mesmo registo e obter ainda o valor antigo (o cliente contactou um servidor cujo registo ainda não tinha sido actualizado). No segundo método tem que haver atomicidade nas acções efectuadas pelos clientes no servidor. Quando um cliente escreve um valor num registo do servidor, esse valor tem que ser propagado pelas diferentes réplicas, de forma a que qualquer cliente que aceda a esse registo obtenha sempre o valor já actualizado.

Depois desta introdução aos cuidados a ter com um servidor replicado (falha/reintegração de réplicas e controle do acesso concorrente aos registos de inteiros) segue-se um esboço da arquitectura do servidor a desenvolver:

ApplSupportLayer
TotalAbcastLayer
VSynchLayer
StableUniformLayer
HealLayer
InterLayer
IntraLayer
SuspectLayer
MergeOutLayer
GossipOutLayer
GroupBottomLayer
TCPCompleteLayer

Decidiu-se elaborar este desenho com base nos relatórios de anos anteriores e na documentação disponível, o que nos levou a adoptar as camadas GroupBottomLayer, GossipOutLayer, MergeOutLayer, SuspectLayer, HealLayer, InterLayer, Sta-

bleUniformLayer, IntraLayer e VSyncLayer, comuns a todas as pilhas que vimos por serem imprescindíveis para a implementação da pilha de protocolos e consequentemente do sistema Appia. Quanto às camadas restantes: optou-se pela TCPCompleteLayer por ser preferível basear a comunicação no protocolo TCP por ser mais fiável que o UDP o que é coerente com o propósito do projecto; optou-se pela camada TotalAbcastLayer por ser uma implementação já feita de uma camada que garante ordem total na entrega das mensagens, imprescindível para um acesso atómico; optou-se finalmente pela camada ApplSupportLayer para separar as funcionalidades de cada servidor réplica da pilha de protocolos Appia subjacente a cada um.

### 3. Conclusão e Trabalho Futuro

Nesta fase do desenvolvimento pretendeu-se aprofundar os conceitos e os modelos associados ao sistema Appia, em particular, e à gestão de acesso e de informação em servidores replicados. Foi dada particular relevância à escolha da organização da pilha protocolar do Appia tendo-se tentado desenvolvê-la de forma a que a aplicação dela resultante não requeira a utilização de funcionalidades que não estejam já contempladas no pacote Appia actual. Dada esta premissa, decidiu-se usar a base comum a quase todas as pilhas Appia com comunicação persistente - TCPCompleteLayer, GroupBottomLayer, GossipOutLayer, MergeOutLayer, SuspectLayer, IntraLayer, InterLayer, HealLayer, StableUniformLayer, VSyncLayer - acrescentando-lhe a TotalAbcastLayer para permitir o acesso atómico e a ApplSupportLayer para fazer a ligação entre as mensagens provenientes das camadas inferiores e os métodos próprios da aplicação (ler registo, escrever registo). Pretende-se futuramente, aproveitar o código disponível referente a cada uma destas camadas, modificando-o de forma a servir o propósito da aplicação a desenvolver. Nesse sentido e numa primeira fase an-

terior à implementação, proceder-se-á ao estudo exaustivo das classes e funcionalidades disponibilizadas na API do Appia, recorrendo, sempre que possível a implementações que contenham funcionalidades semelhantes às de que iremos necessitar na implementação do sistema. A esta fase, seguir-se-á uma fase de desenho onde se especificará mais pormenorizadamente a arquitectura geral do sistema, fazendo, quando necessário, ajustes ao especificado neste primeiro artigo. Finda essa fase proceder-se-á finalmente a implementação da arquitectura definida na fase anterior contemplando ambas as formas de acesso - melhor esforço e atómica.

### References

<http://appia.di.fc.ul.pt>