

HERMES

A Distributed Event-Based Middleware Architecture

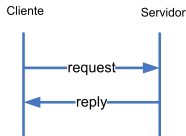
Alexandre Zua Caldeira

Tecnologias de Middleware 2006/07

15 de Dezembro de 2006

- **Introdução**
- Hermes Design
- Type-based Algorithm
- Type- and Attribute-based Algorithm
- Tolerância a Falhas

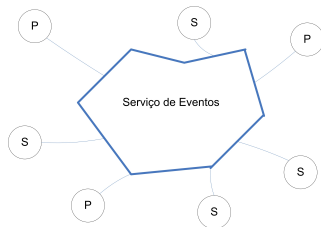
Invocation-based Middleware(CORBA, Java RMI)



Problemas

- One-to-one
- Problemas de escalabilidade na Internet

Event-based Systems: Uma alternativa



Problemas

- Não fornecem serviços normais da camada de middleware
 - *Type checking* das expressões ou filtros, Fialbilidade, Controlo de Acessos, etc...
- Filtros sobre os eventos são feitos através de pares (**atributo, valor**)
 - Mais difícil para especificar filtros sobre os eventos
 - Pouca integração com linguagens orientadas aos objectos

Event-based Middleware: Requisitos

● Escalabilidade

- A implementação tem que ser distribuída
- Estados locais
- Utilização eficiente dos recursos

● Interoperabilidade

- Independente da Linguagem e da Plataforma
- Utilizadores fixos e móveis

● Fiabilidade

- Vários modos de garantias de fiabilidade e QoS: desde melhor esforço até garantia total
- Mecanismos de Tolerância a Falhas

● Expressividade

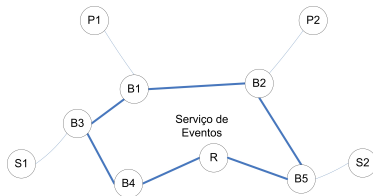
- Mecanismos para a exprimir filtros adequadamente
- Mecanismos para a composição de filtros

● Usabilidade

- Integração com as várias aplicações
- Fácil de usar pelas aplicações
- Mapeamento intuitivo **evento - objectos de Linguagem de programação**
- Type checking estático e dinâmico de subscrições e publicações

- Introdução
- **Hermes Design**
- Type-based Algorithm
- Type- and Attribute-based Algorithm
- Tolerância a Falhas

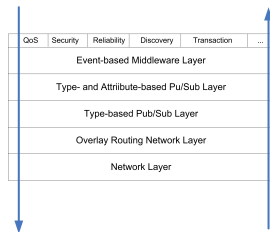
Publishers, Subscribers and Brokers



- **Subscribers:** subscrevem eventos, que lhe serão entregues quando publicados.
- **Publishers:** Publicam eventos
- **Overlay Network of Brokers**
 - Fornece o serviço lógico de encaminhamento na rede de brokers
 - Principal função: route(mensagem, id-destinatario)
 - Mecanismos de tolerância a falhas

Descrição geral da Arquitectura do Sistema

- **Type-based Pub/Sub Layer**
 - Estabelece o Rendez-vous
 - Serviços equivalentes aos Sistemas baseados em tópicos
- **Type- and Attribute-based Pub/Sub Layer**
 - Implementa os filtros
 - Distribui os filtros pela rede brokers
- **Event-based Middleware Layer**
 - Fornece API: Adicionar e remover tipos; *advertise*, *subscribe*, *publish* de eventos
 - Implementam funcionalidades típicas de



- Introdução
- Hermes Design
- Type-based Algorithm
- Type- and Attribute-based Algorithm
- Tolerância a Falhas

Tipos de mensagem e Estados

Na rede de brokers circulam 4 tipos de mensagens:

- Mensagens de tipo
- *Advertisements*
- *Subscriptions*
- Publicações de eventos

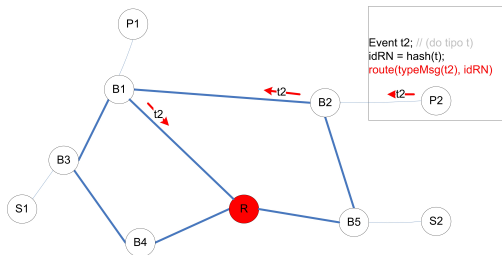
Os estados nos brokers referem apenas informação local. Cada broker guarda estado sobre:

- Tipos de Evento encaminhados por si
- *Advertisements* que recebeu ou enviou (e de quem ou a quem)
- *Subscriptions* que recebeu ou enviou (e de quem ou a quem)

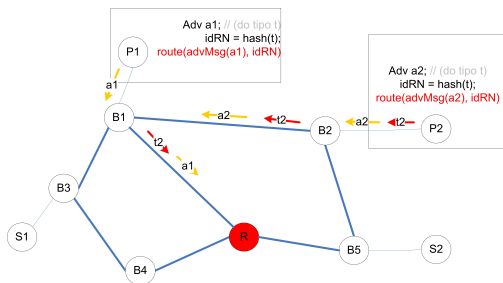
Importante: Advertisements e Subscriptions são propagados pelos brokers apenas se estes não propagaram mensagens desse tipo ou mais gerais.

Setup do ponto Rendez-vous

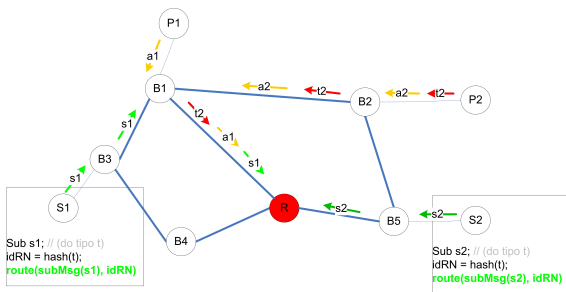
Servem para identificar um nó para representante do novo tipo de evento.



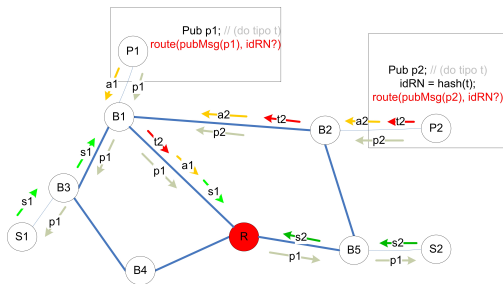
Advertisement



Subscrições



Publicações



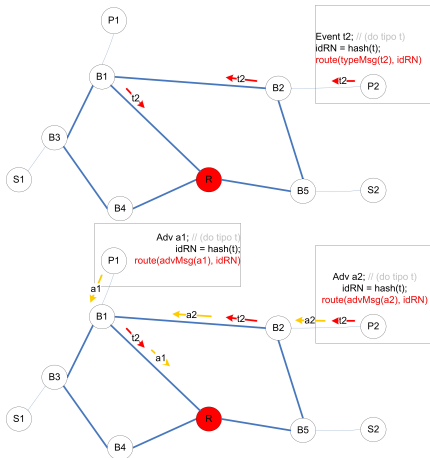
- Introdução
- Hermes Design
- Type-based Algorithm
- **Type- and Attribute-based Algorithm**
- Tolerância a Falhas

Type attributes: O que há de novo?

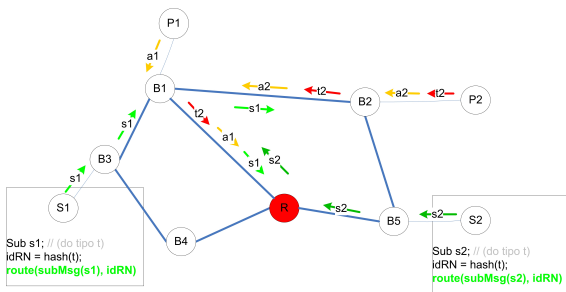
Atributos de Tipo:

- Um tipo de evento pode ter atributos
- Subscribers aplicam filtros sobre os atributos do tipos de eventos em que está interessado
- As publicações serão entregues aos subscribers de acordo com o tipo de evento e com os filtros guardados na rede de brokers
- O algoritmo anterior é estendido com a noção de **atributo de tipo**
- Os broker deverão além de manter estado sobre os tipos de evento, mantém também estado para expressões filtros sobre os atributos que tenham propagado

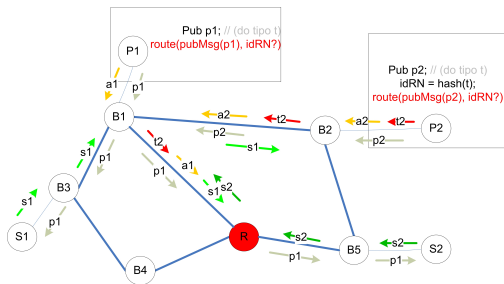
Nada de Novo



Subscrição

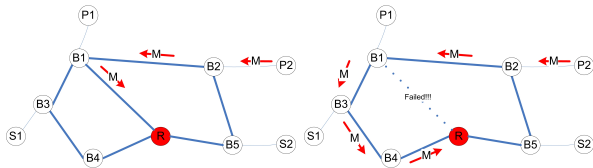


Publicação

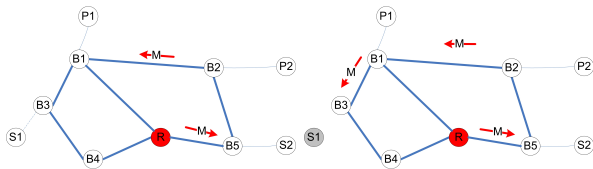


- Introdução
- Hermes Design
- Type-based Algorithm
- Type- and Attribute-based Algorithm
- Tolerância a Falhas

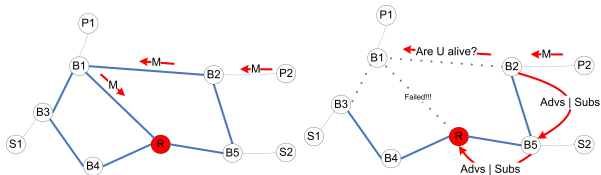
Falha no Link



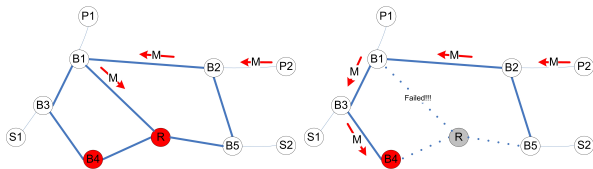
Falha no Cliente



Falha no Broker



Falha no Rendez-vous



Modelo de Consistência

- Clientes subscrevem todos os RNs
- Clientes anunciam (advertise) a todos os RNs
- RNs trocam subscrições
- RNs trocam advertisements

Referências Bibliográficas

[1] Pietzuch, P.R.; Bacon, J.M.

Hermes: A Distributed Event-Based Middleware Architecture

Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on , vol., no.pp. 611- 618, 2002

[2] Miguel Castro, Peter Druschel, Anne-Marie Kermarrec, Antony Rowstron

SCRIBE: A large-scale and decentralized application-level multicast infrastructure

IEEE Journal on Selected Areas in Communications, vol.20, no.8pp. 1489- 1499, Oct 2002