ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

Information-Centric Networks

Section # 10.2: Publish/Subscribe

Instructor: George Xylomenos

Department: Informatics

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

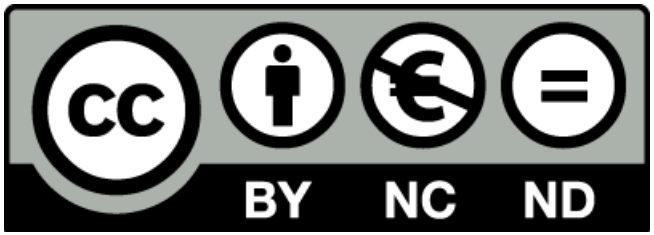
Co- financed by Greece and the European Union

Funding

- These educational materials have been developed as part of the instructors educational tasks.
- The **“Athens University of Economics and Business Open Courses”** project only funded the reformatting of these educational materials.
- The project is being implemented as part of the Operational Program “Instruction and Lifelong Learning” and is co-financed by the European Union (European Social Fund) and national funds.

Licencing

- These educational materials are subject to a Creative Commons License.



Week 10 / Paper 2

- Hermes: a distributed event-based middleware architecture
  - P.R. Pietzuch, J.M. Bacon
  - ICDCS 2002 Workshops
- Main point
  - Pub/sub has limitations compared to invocation models
  - Hermes is a scalable pub/sub middleware
    - Adopts a type- and attribute-based model
    - Type hierarchies are supported
    - Scalable routing using an overlay network
    - Built-in fault-tolerance mechanisms

Introduction

- CORBA and Java RMI have proved very useful
  - Higher-level interface that hides underlying complexity
  - Invocation model: request/reply client/server
  - Good for LANs but does not scale
- Event-based pub/sub is a viable alternative
  - Subscribers express interest in events
  - Publishers create events
  - The middleware delivers events to subscribers
  - Decoupled, many-to-many communication
  - Hermes is a pub/sub middleware
    - Type-checking, reliability, access control, transactions
    - Integration with object-oriented languages
    - General purpose facilities for different applications

Requirements

- Scalability
  - Distributed implementation, local state, low resource consumption
- Interoperability
  - Language, platform and network-independent
  - Allow devices to join and leave the system
- Reliability
  - Support a range of client requirements
  - Fault-tolerance must be built-in
- Expressiveness
  - Content filters for events and composite event expressions
- Usability
  - Intuitive mapping between events and language

Design of Hermes

- Hermes middleware model
  - Application using Hermes consist of event clients and brokers
  - Event brokers represent the actual middleware
  - Event clients are event publishers or event subscribers
    - Communicate via events using Hermes
    - Relatively lightweight as they use the brokers
  - Brokers connected in arbitrary topology
    - Accept subscriptions from subscribers
    - Convert publications to messages
    - Route messages towards subscribers
  - Hermes pushes subscriptions towards publishers
    - Filter events as early as possible
  - A dissemination tree is created to forward events
    - The tree uses overlay multicast

Overlay routing networks

- Overlays are built on top of other networks
  - Suboptimal but more sophisticated routing
  - Event brokers in Hermes form an overlay
  - The overlay is similar to Pastry
    - Routes messages with ID's to the node with the closest ID
  - The overlay takes care of failures and node churn
- Rendezvous between publishers and subscribers
  - Previous systems flooded other subscriptions or publications
  - Hermes uses rendezvous nodes for each event type
    - Their ID is a hash of the event type name
    - Simply send publication or subscription to that ID
    - Replication used to make rendezvous fault-tolerant
    - This is very similar to the Scribe multicast service

Types and attributes

- Event handling by programming languages
  - Usually events are collections of attribute-value pairs
  - Hermes associates each event with a type
    - The event type contains a number of data fields (i.e. attributes)
- Topic and content-based pub/sub
  - Topic-based is scalable but not flexible
  - Content-based is flexible but not scalable
  - Type- and attribute-based pub/sub is in between
    - The event type is the topic
    - Filters operate over type attributes
    - Each event type is served by an event broker
  - Event type hierarchies are possible
    - A typed subscription will match a type and its descendants

Architecture

- Hermes uses a layered approach
  - Assumes an IP unicast network on the bottom
  - Overlay routing network layer handles routing
  - Type-based pub/sub layer sets up rendezvous nodes
    - Provides topic-based pub/sub
  - Type- and attribute-based pub/sub layer handles filtering
    - Distributes filters towards the source
  - Event-based middleware layer
    - Provides API to applications
    - Advertise, subscribe and publish events
    - Add and remove event types
    - Perform type-checking of events and subscriptions
    - Fault-tolerance, reliable delivery, type discovery, security, etc…

Event routing algorithms

- Four message types are used
  - Type messages: add new event types to the system
    - Contain definition of types which are used for type-checking
  - Advertisement messages: publisher announces event types
    - Set up dissemination paths towards rendezvous nodes
  - Subscription messages: subscriber expresses interest on events
    - Also routed towards rendezvous nodes
    - Can continue towards publishers for filtering
  - Publication messages: carry published events
    - Follow paths created by advertisements and subscriptions
- Two levels of routing
  - Type-based (lower layer)
  - Type- and attribute-based (higher layer)

Type-based routing

- Setting a rendezvous node
  - A type message is sent to the rendezvous node
    - The ID is determined by hashing the event type name
- Sending advertisements and subscriptions
  - Each node on the path maintains state about messages
    - Brokers that send or were sent the message
  - Advertisements and subscriptions are merged if possible
    - A broker does not forward a message for the same type
  - A tree of brokers is formed by these messages
- Publishing events
  - Publications are flooded over the dissemination tree
    - They do not need to reach the rendezvous point first

Type- and attribute-based routing

- Distributes filter expressions through broker network
  - Filters examine event attribute fields
  - Filters also propagate towards advertising brokers
    - They do not stop at the rendezvous point
    - They are still merged however if possible
    - More general filters are not merged
  - Publications are filtered as early as possible
    - At the first broker where a filter exists
- Hierarchies of event types
  - A new type can have a parent
  - The rendezvous node of the parent maintains pointers to children
  - Subscriptions for the parent are sent to its children

Fault-tolerance

- Hermes exploits the overlay network for fault-tolerance
  - Brokers only maintain soft-state that needs to be refreshed
  - A heartbeat protocol detects failed neighbors
  - Link failure: overlay routing bypasses the failed link
  - Broker failure: the broker is bypassed by overlay routing
  - Client failure: state created by client expires in due course
  - Rendezvous failure: each rendezvous node is replicated
    - A salt value is used to determine replica IDs
    - Four replication techniques are possible
    - Clients subscribe to all rendezvous nodes
    - Clients advertise to all rendezvous nodes
    - Rendezvous nodes exchange subscriptions
    - Rendezvous nodes exchange advertisements

Implementation

- Java-based implementation
  - Classes for brokers, sources and sinks
  - Communication via XML-defined messages
  - XML Schema is used to define formats and types
  - XML mapped to Java to hide it from clients
  - Events appear as Java objects to clients

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

End of Section # 10.2

Course: Information-Centric Networks, **Section # 10.2: Publish/Subscribe**

Instructor: George Xylomenos, **Department:** Informatics

European Union
European Social Fund
OPERATIONAL PROGRAMME EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION, LIFELONG LEARNING AND RELIGIOUS AFFAIRS
MANAGING AUTHORITY
NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND
Co- financed by Greece and the European Union