CAESAR: Middleware for Complex Service-Oriented Peer-to-Peer Applications

Lipo Chan, Shanika Karunasekera, Aaron Harwood and Egemen Tanin

P2P Group/NICTA Victoria Research Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
Background and Motivation

• Background
  – P2P Massively Multi-Player Online game (MMOG)
  – Middleware for MMOGs - currently being commercialized

• Experiences and Motivation
  – Tight coupling to P2P protocols leads to tedious development activity, especially for complex applications.
  – Limited to sharing data storage, hence minimal ability to delegate processing amongst peers; not fully utilizing the computing power of P2P paradigm.
  – Complex development leads to low consideration for software quality attributes, e.g. reliability, security, interoperability and performance.
Outline

- Design principles of CAESAR (*Complex Application DEvelopment using Service-Oriented Architecture*) middleware.

- Overview of CAESAR.

- Implementation of CAESAR.

- Complex application development with CAESAR.

- Current and future agenda.
Design Principles (1)

- P2P protocols use a variety of data structures and complex algorithms to provide efficient lookup within large-scale peer-based network.

- Addition of non-functional requirements increases protocol complexity.

- Desirable to reduce direct exposure to these complexities in order to simplify application development.

- Abstraction through interfaces to resolve issue of tight coupling, hence leads to interoperability and extensibility.
Design Principles (2)

- Complex applications can be computationally intensive, thus requiring more sharing of storage, but logic as well.

- **Sharing logic** through services, where peers offering (or requiring) the same types of services are able to form communities and delegate processing amongst communal peers.

- Service-oriented paradigm enables flexible functional composition to be formed amongst peers, thus harnessing collective computing power.
Design Principles (3)

- Abstraction of P2P protocols enables open services to be developed, i.e. integration with multiple P2P protocols.

- Flexible mix-and-match *enhances peer functionality*.

- Well-defined interfaces support dynamic binding between services and P2P protocols.
Design Principles (4)

- Dynamic nature of P2P paradigm requires handling of issues such as data consistency and integrity.

- Replication and migration mechanisms are essential, but non-trivial, which lead to cumbersome development technicality.

- Supporting *robust development of P2P applications* through an embedded network management approach that hides complicated network processes.
Middleware Overview (1)

- Component-based architecture.
- 3 levels of abstractions: application, service, protocol.
- Application Director used by application developers to access functions provided by CAESAR.
- Equivalently, Service Façade for service developers and Protocol Façade for protocol developers.
- Service/Protocol Plugin supports addition of services and protocols.
• Network and Object Management works with Service Façade and Protocol Façade to support dynamic binding between services and protocols.
• Also manages peer service communities, or as we term service overlays.
• Service overlays are being managed via a common data structure (ServiceObject) that contains processing data and states.
• Also provides essential network processes, e.g. replication and migration.
• Internet Simulation and Emulation to facilitate large-scale testing that reflect real-world P2P networks.
• Emulation consumes significantly less resources.
• Details of APIs are available in the paper.
Collaboration between core components

Middleware for Service-Oriented Computing (MW4SOC)
Middleware Implementation

- Initial implementation was in C++ on a UNIX platform.

- A number of existing P2P protocols have been implemented as protocol plugins to CAESER.

- Several services that use these protocols have been implemented as service plugins.

- More recently, with the commercialization efforts of the MMOG a .NET version of the software was developed, mainly focused towards the MMOG (customized version).
Applications:

- Event Finder
- Platform for Message Passing Interface (MPI) programs
- Collaborative Intrusion Detection Service
- Massively Multiplayer Online Games
- Integrated Development Environment to support the use of CAESAR – plugin to Eclipse.
Current & Future Agenda

- We are enhancing the middleware to support more management functionalities (migration, replication).
- A java version of the CAESER middleware architecture is currently being developed (SEDA is being evaluated as an event driven architecture to implement the middleware).

Questions?

Further information: http://p2p.csse.unimelb.edu.au