Structured overlay Networks Application Platform

SNAP

Project White Paper

Carles Pairot Gavaldà - <carles.pairot@urv.net>
Pedro García López - <pedro.garcia@urv.net>
Architecture and Telematic Services Research Group
Universitat Rovira i Virgili
Tarragona
Spain
### Table of Contents

- What is the SNAP Platform? 3
- Architecture 4
- Edge Services 5
- Core Services 6
- Virtual Communities on top of SNAP 8
- Connection with the ObjectWeb Consortium 9
- Current Stage 10
Peer-to-peer (p2p) networks are powerfully emerging on the Internet, and their main utilization nowadays involves mainly file sharing communities where users can locate and download / upload files of their interest.

Nevertheless, p2p must not be reduced to such application domain. Alternative scenarios can as well be taken into account, thus allowing users to cooperate, collaborate, and even access personal data. However, very few platforms providing such services exist currently, and if they do, they are usually closed and proprietary. This fact makes it impossible to extend many services found in such networks.

In the p2p world, a revolution has undergone since 2001 with the introduction of Distributed Hash Tables (DHTs), also called Structured Peer-to-Peer Overlay Networks. This form of organizing the p2p networks has been quite revolutionary, and it basically allows location and insertion of data in a deterministic and efficient way. In stark contrast, older p2p algorithms could not guarantee location of a specified resource in the network: it could be found or not, depending on the distance between the requester and the data source.

Following such approach, our research group started investigating possible uses of such technology, focusing on developing a middleware framework for easing the process of developing applications on top of these networks. Such efforts made the appearance of Dermi (http://ants.etse.urv.es/DERMI), a decentralized event-based object middleware possible. However, programming distributed objects may seem sometimes quite inappropriate for the developer, since it has to deal with some low-level details.

A natural evolution of Dermi was p2pCM (http://ants.etse.urv.es/p2pcm), which wrapped these distributed objects into components. p2pCM is a component model built on top of a structured p2p overlay network. It provides the typical component’s services, like a life cycle service, deployment service, passivation on demand, etc. Furthermore, it adds several novel services, which include edge component activation, DHT persistence, and others.

However, we believe we need a framework for structured p2p networks which benefits from the services offered by the underlying routing layer, linking together the above described systems. Such framework would provide a unique access point where developers can sustain their brand-new developed applications.

Such framework is the Structured overlay Networks Application Platform (SNAP). SNAP is a J2EE web application deployment and management infrastructure for structured overlay networks. By using SNAP, developers can easily deploy any kind of J2EE web application onto a worldwide structured peer-to-peer network. Moreover, they can use the services the framework provides, which let developers concentrate on the real aspects of the web application to develop, thus forgetting about persistence, security, load balancing, failover, and others.

Indeed, SNAP’s learning curve is low for J2EE-used programmers. Programming SNAP’s services is very easy, thus allowing worldwide web application deployment.
The SNAP platform allows development and deployment of J2EE compatible web applications over structured p2p networks. Since it follows the p2p philosophy by design, we consider network nodes as dynamic lightweight peers, which can join and leave the network at a rather high rate.

Following such lightweight scheme, all nodes host a modified copy of a light webserver (Jetty), which acts both as a p2p network client and server simultaneously. Therefore, clients can connect through their favourite web browser to any node of the SNAP infrastructure for accessing any deployed web applications.

SNAP’s Architectural Diagram

Each node in the overlay network is conformed basically by a p2p network routing layer, which routes all messages to / from the network. Superior layers are the object (Dermi) and component (p2pCM) layers through which the lightweight webserver interacts with the p2p network. The SNAP Core component provides all of SNAP’s services to web application developers.
Edge Services

SNAP provides a set of edge services which try to uniformize application access and deployment, transparently to the web application developer. In such scenario, SNAP applications are deployed onto the network and instantiated dynamically on demand.

Basically, SNAP’s edge services include

- Web application edge service
- Database persistence edge service

In SNAP’s scenario, every time an application is requested by any client, it is automatically downloaded from the p2p network, deployed and instantiated on the local webserver. All accesses will be local to that lightweight server.

However, this will only happen whenever no available instances of that web application are found already running on the network. In such case, the client will automatically be redirected to the closest webserver which hosts that application. This edge service allows multiple replicas of an application to be currently running on multiple network nodes. All changes will accordingly be replicated all along such replicas.

Moreover, web applications may need database access. This form of persistence is also provided by SNAP (more about this topic can be found on the following section). Database persistence is also replicated among all web application running instances (they form a cluster altogether).

Once an application wishes to access the database, an instance of it will automatically be instantiated on the local server node. All SQL statements will be executed locally, but replicated automatically to other already running database replicas. Database persistence can be considered as well, an edge service, since access will occur on the closest active server instance.

More policies regarding edge services can be taken into account to make them more adaptative to dynamic conditions, providing load balancing. New database and application instances could be activated automatically depending on certain threshold parameters. This way, applications would be more tolerant to high request peaks or node congestion failures.
Core Services

SNAP is built on top of two main blocks: p2pCM (http://ants.etse.urv.es/p2pcm), and Dermi (http://ants.etse.urv.es/DERMI). The former one is a reusable component model based on structured p2p networks, while the latter is a distributed object model also for this kind of networks. Both layers rely on an lower p2p routing substrate: FreePastry, from Rice University / Microsoft Research.

SNAP platform’s core services include those services provided to the web application developer which map transparently into the underlying p2p network substrate.

Right now, a research prototype of SNAP is already on the move. It has been developed as open source code, and implements the following core services:

- **Secure web application deployment.** A J2EE application authentication service is bundled into SNAP. We have opted for centralizing the deployment phase, so as only the network's administrator can install, deploy, and monitor applications on the SNAP network. As a consequence, the existent anarchy generally found in p2p networks is controlled, preventing other users to do whatever they want.

- Obviously, before deploying an application, the administrator's signature must be present. Otherwise it will not be deployed, and therefore, users will be unable to use it. Public/private key and certificates are used to implement this functionality.

- **Decentralized web application deployment.** Deployed web application data is not stored solely on one node. These signed applications are distributedly stored among different network nodes (recall that the p2p network itself is organized in the form of a Distributed Hash Table), to guarantee fault tolerance. Whenever any node requires the activation of any application, the closest instance is accessed, or a new one is activated locally, checking that the administrator's signature is present on the web application archive (.WAR). In case other extra-services are required (eg: database), these are activated on demand.

- **Uniform web application location.** For accessing any SNAP deployed web application, special URI-style addresses have to be used (p2p://deskshot.urv.net). These locators uniformize the address space, as well as the application's access independently of its real location (IP address), and the service provider. SNAP will internally redirect requests to these applications to the real IP addresses (which may change over time). Therefore, these p2p locators do not have real location information embedded on them.
• **Adaptation and load balancing.** SNAP optimizes network resources, and it thus adapts to application’s load increments. This way, specification of a minimum number of nodes where the application can be replicated is considered. A cluster of J2EE web applications is then formed, transparently to both users and developers. All requests are distributed accordingly depending on their physical origin. Bottlenecks are thus efficiently reduced.

• **Persistence mechanisms.** Our infrastructure provides two types of persistence modes, depending on the application’s needs.
  - **Replicated file warehouse.** This kind of persistence mode is used to store objects’ state in the DHT network. Data is automatically replicated among various nodes thus guaranteeing its availability and transparent failover.
  - **Replication and clustering of relational databases.** Whenever our applications require persistence on relational databases, SNAP provides transparent replication between the different cluster members of the web application. The database is activated dynamically and on demand. Additionally, a timeout period can be specified (the database does not receive any requests), after which the database is deactivated so as to free resources in the node. Note that consequently following the p2p philosophy, we have opted for a very lightweight database engine: HSQLDB, slightly modified to fit our needs. If all application’s database instances have been deactivated, its state is stored in the replicated file warehouse, thus guaranteeing that next database activation will recover state from there.

• **p2p Application Programmer’s Interface (p2p-API).** Since SNAP has been built on top of the p2pCM component model and the Dermi object middleware, it provides a natural gateway to these frameworks’ API. Therefore, we can easily create components or objects which use their services of distributed interception, application level multicast, or their p2p invocation abstractions (arycall, manycall, multicall, directcall, hoppedcall), thus enriching the possibilities of the application developer.

• **Easy adaptation of already existent J2EE applications to SNAP.** Any J2EE application can be easily transformed into a SNAP application through an easy and automated process of signing and packaging. The administrator can easily deploy static web application contents in the SNAP network too, without having to change a line of code.

When talking about database applications, we have tried to make as transparent as possible the transition to SNAP. Developers can choose using direct JDBC connections (making slight changes in the way the JDBC connection is obtained), or using DataSource (where they only have to change the web.xml file), without touching a line of code.
Virtual Communities on top of SNAP

SNAP by itself does provide the required infrastructure for the development of wide-area web applications. Additionally it is planned to add a virtual community layer on top of SNAP which will allow the inclusion of collaborative services on top of the infrastructure by using plug-ins.

Moreover, we have been developing DeskShot, an application which offers users access to their p2p desktop. Once authenticated, users can access (independent from their location) applications like calendars, bookmark and document management, file sharing, or instant messaging. Additionally, Deskshot provides a profile matching service to help users find each other by seeking their common interests, helping them create virtual online communities.

Furthermore, we are involved in a joint project proposal with Atos Origin, where we are studying the interaction ways between their enterprise platform targeted to worldwide events, and our p2p infrastructure. We will be studying secure deployment and utilization of the p2p network so as to reduce economic expenses. Since 1989, the department of worldwide events at Atos Origin has been exclusively working on providing IT services to international worldwide events. Specially, Atos Origin is involved in the most important international sport events, and has gathered a lot of experience. This experience began with the Olympic Games at Barcelona (1992), and has been continuing all along Salt Lake City (2002), FIFA World Cup Korea Japan (2002), and many others, as well as the upcoming Olympic Games, which will be celebrated on Beijing, China (2008).

The worldwide events platform covers a wide variety of systems and subsystems specially designed for matching the level of demand required for such events. The services provided by such platform include an Event Management System, which manages the accreditation, plans journeys, manages material resources, protocols, athlete hosting, Olympic Village, etc; an Information Diffusion System, which distributed news, and results to the mass media; and a Multimedia Content Manager, which manages different multimedia contents (audio, video, statistics, etc).
Connection with the ObjectWeb Consortium

We believe there exist several connection spots between the SNAP project and the ObjectWeb Consortium.

- SNAP relies on a structured p2p network substrate and provides services which target a different profile not found in any of the existent ObjectWeb projects. Indeed, it is possible to adapt SNAP to provide enterprise-oriented services, which could interconnect with an already existent network of JOnAS servers, providing interesting edge computing abstractions.

- Although SNAP’s clustering and persistence mechanisms perform basically similar functions to those found on ObjectWeb’s C-JDBC (database clustering and replication), they target different domains, which in the case of C-JDBC involves relatively static clusters of fixed nodes. SNAP’s database replication follows the p2p philosophy, and is more dynamic in the sense that it has been designed to provide support for highly dynamic joins and leaves of nodes in the network.

- Another project SNAP compares to is ProActive. The ProActive framework is mainly targeted to grid computing applications, and it thus provides a very well defined API for such matters. Nevertheless, SNAP is designed as a web application deployment infrastructure for wide-area p2p environments. In such context, SNAP’s provided services match the needs of these kind of applications and makes good use of the underlying network substrate to implement p2p-oriented services (p2p call abstractions, decentralized location, decentralized persistence, etc).

We believe SNAP can be an interesting application deployment framework for the ObjectWeb consortium, because it targets an application domain which does not completely fit in any of the already available projects.
Current Stage

A research prototype of SNAP is currently available for download via [http://ants.etse.urv.es/snap](http://ants.etse.urv.es/snap) [COMING SOON].

Many work remains to be done in the future:

- **Extensive testing of SNAP.** Our infrastructure has been designed to be scalable, with a very high number of users in mind. However, we need to completely validate our approach at a worldwide scale. We are now performing extensive tests by deploying SNAP on the worldwide testbed network Planet-Lab ([http://www.planet-lab.org](http://www.planet-lab.org)). This network permits to perform intensive tests in a continuously dynamic and changing environment where latencies, node availability, and other factors change rapidly.

- **Interaction with mobile devices.** Mobile devices have very little resource capacity. We plan to add a new layer to SNAP so as to allow these devices to interact with SNAP web applications.

- **Improve security.** Apart from the security provided at deployment time, we need to strengthen this aspect to avoid undesired intrusions on the network.

- **Improve persistence.** Although we consider all peers to be equal, this assumption may not be true when requiring stronger persistence mechanisms (eg: heavier database engines). In such case, we could consider some powerful and rather static nodes which would form the SNAP network’s backbone.