

Overlay Weaver: An overlay construction toolkit

Kazuyuki Shudo *, Yoshio Tanaka, Satoshi Sekiguchi

*Grid Technology Research Center, National Institute of Advanced Industrial Science and Technology, AIST Tsukuba Central 2,
Tsukuba, Ibaraki 305-8568, Japan*

Available online 14 August 2007

Abstract

A layered model of structured overlays has been proposed and it enabled development of a routing layer independently of higher-level services such as DHT and multicast. The routing layer has to include other part than a routing algorithm, which is essential for routing. It is routing process, which is common to various routing algorithms and can be decoupled from a routing algorithm.

We demonstrated the decomposition by implementing an overlay construction toolkit *Overlay Weaver*. It facilitates implementation of routing algorithms and we could multiple well-known algorithms just in hundreds of lines of code with the toolkit. The decomposition also enables multiple implementations of the common routing process. Two implementations the toolkit provides perform iterative and recursive routing, respectively. Additionally, to our knowledge, the toolkit is the first feasibility proof of the layered model by supporting multiple algorithms and the higher-level services.

Such modular design contributes to our goal, which is facilitation of rapid development of realistic routing algorithms and their application. We demonstrates that *Overlay Weaver* supports the goal by conducting large-scale tests and comparisons of algorithms on a single computer. The resulting algorithm implementations work on a real TCP/IP network as it is.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Overlay network; Structured overlay; Distributed hash table; Emulation

1. Introduction

It is essential for large-scale Internet-wide applications to construct an application-level network in an autonomous and decentralized way. Such a network is called an overlay network because it is based on the underlying physical network, but its structure is different from the underlying network. An overlay provides elemental functions to build distributed applications such as lookup and multicast, and maintains performance and fault tolerance with more than millions of computers.

It requires a fair amount of work to incorporate an overlay algorithm into application software. For example, distributed hash table (DHT), a representative application of structured overlays, itself has thousands of lines of code at least, because it needs various jobs like communication

and storage management other than the overlay algorithm. Its application must be larger and more complicated than it.

Dabek et al. proposed layered abstractions of structured overlays [1]. They separated common services such as DHT, multicast, and anycast from an underlying routing layer. The model suggests that it enables separated design and implementation of a routing layer named key-based routing (KBR) from the common higher-level services.

The routing layer is monolithic even with the layered abstractions. Routing algorithm is not the only part of the routing layer, which has to perform other jobs including communication. Furthermore, the routing layer has to include other part than a routing algorithm, which is the essential for routing. It is routing process, which is common to various known routing algorithms. It can be decoupled from a routing algorithm.

We demonstrated the decomposition by implementing an overlay construction toolkit *Overlay Weaver* [2]. We decoupled a routing algorithm from the common routing process by designing a programming interface between

* Corresponding author. Present address: Utageo Inc., Tokyo. Tel.: +81 3 3461 1118; fax: +81 3 3461 1119.

E-mail address: shudo@computer.org (K. Shudo).

them. The interface has been able to support well-known algorithms, Chord, Pastry, Tapestry, and Kademlia. Additionally, to our knowledge, the toolkit is the first feasibility proof of the layered abstractions proposed by Dabek et al. [1], because the toolkit provides the multiple routing algorithms and multiple higher-level services such as DHT and multicast.

The decomposition facilitates development of routing algorithms. Implementations of the above-mentioned well-known algorithms are just in hundreds of lines of code. The decomposition also allows multiple implementations of the common routing process to be combined with routing algorithms. The toolkit provides two implementations which perform iterative and recursive routing, respectively.

Such modular design contributes to our goal. It is facilitation of rapid development of realistic routing algorithms and their application. Overlay Weaver provides an emulator of distributed environment to support the goal. It enables algorithm developers to test and improve their algorithms with large number of nodes on a single computer. The resulting algorithm implementations work in a real environment.

The following section discusses related work. We show a programming interface, which separates the routing process from the algorithms in Section 3. Section 4 describes the components and tools the toolkit provides and how they support overlay design, implementation, evaluation, and comparison. In Section 5, we demonstrate that we could implement various well-known algorithms with little code. We also present the results of tests and comparisons of the algorithms on a large-scale.

2. Related work

MACEDON [3,4] and Mace [5] are overlay construction software which support multiple routing algorithms as does the Overlay Weaver.

A user describes an algorithm in MACEDON language, which is like C/C++ but specific to the overlay description. MACEDON translates the description into executable C++ code. The generated code communicates using TCP or UDP, and MACEDON can generate code for a network simulator ns even though this is “partial support” [3]. MACEDON provides distributed hash table (DHT) implementations, i.e., Chord and Pastry.

The amount of code MACEDON requires to describe an algorithm is comparable to Overlay Weaver. The difference is only up to 50% of the amount of code for Overlay Weaver (Section 5.1). MACEDON reduced the amount by introducing a domain-specific language and Overlay Weaver achieved similar results by the separation of routing process and algorithms. Possible problems each approach introduces are as follows. MACEDON’s approach involves a higher learning cost for dedicated language. On the other hand, routing driver (Section 3) of Overlay Weaver provides no specific support to implementation of unstructured overlays, which require a large amount of code.

However, it is possible to implement unstructured overlays directly using the toolkit’s messaging services (Section 4).

Experiments with MACEDON have been performed on an Internet emulator ModelNet [6]. The number of underlying computers ranged from 2 to 50 in the emulation. The maximum number of emulated nodes was only 1000 and only experiments on overlay construction (routing table stabilization) was conducted on such a scale. Compared with this, we demonstrate the emulation of 4000 nodes, which is discussed in Section 5.2.

The original length of IDs in Chord is 160 bits and 128 bits in Pastry, but both are 32 bits in MACEDON. The integer type `int` the dedicated language provides is 32 bits and the shortened ID length might be a natural consequence of this. We describe an algorithm in a general-purpose language (Java) for Overlay Weaver and the expressive power of the language is sufficient to represent any algorithm.

Mace [5] is a successive project following MACEDON. The ID length is 160 bits in Pastry, and this is not shortened.

An algorithm description for MACEDON or Mace is dedicated to one of routing styles, iterative, and recursive. In opposition to them, a single description for Overlay Weaver can work in the both styles.

There are libraries that implement structured overlays, such as Bamboo [7,8], Chimera and Tapestry [9,10], and Khashmir [11], FreePastry [12], SharkyPy [13], and OPeN [14] that are all available. However, all the libraries support a single algorithm and we do not have a choice.

P2psim [15] is a simulator for peer-to-peer protocols. It provides multiple algorithm implementations such as Chord, Accordion, Koorde, Kelips, Tapestry, and Kademlia. They are all DHT, which are structured overlays. In the Accordion proposal [16], Li et al. emulated 3000 nodes with p2psim to compare the proposed algorithm with Chord.

Simulation has the advantage being reproducible. Emulation of a concurrent system, which is what Overlay Weaver does, is non-deterministic and its results are not constant. Algorithm implementations for p2psim on the other hand do not work on a real network. They require a relatively large amount of code because they include low-level processes such as remote procedure calls (RPC). The amount of code is several times the length of code for Overlay Weaver and MACEDON (Section 5.1).

3. Decomposition of routing layer

Dabek et al. called routing process on structured overlays key-based routing (KBR) [1]. Following their model, higher-level services are constructed on the KBR layer, which is tier 0. Such services include DHT, multicast, anycast, and message delivery.

Overlay Weaver also followed this model and the routing layer is separated from the DHT service and multicasting service. The model enables choices of multiple components for each layer and applications are combined with various routing algorithms without any modifications.

Download English Version:

<https://daneshyari.com/en/article/449362>

Download Persian Version:

<https://daneshyari.com/article/449362>

[Daneshyari.com](https://daneshyari.com)