

Using FreePastry

Rubén Mondéjar Andreu

<http://www.etse.urv.es/~rmondejar>

ruben.mondejar@urv.net

Departament d'Enginyeria

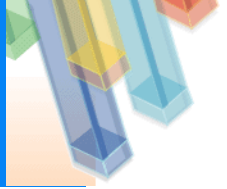
[DΣIM]

Informàtica i
Matemàtiques



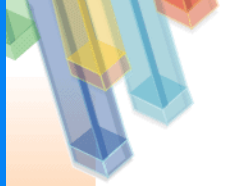
UNIVERSITAT
ROVIRA I VIRGILI

Freepastry : Overview

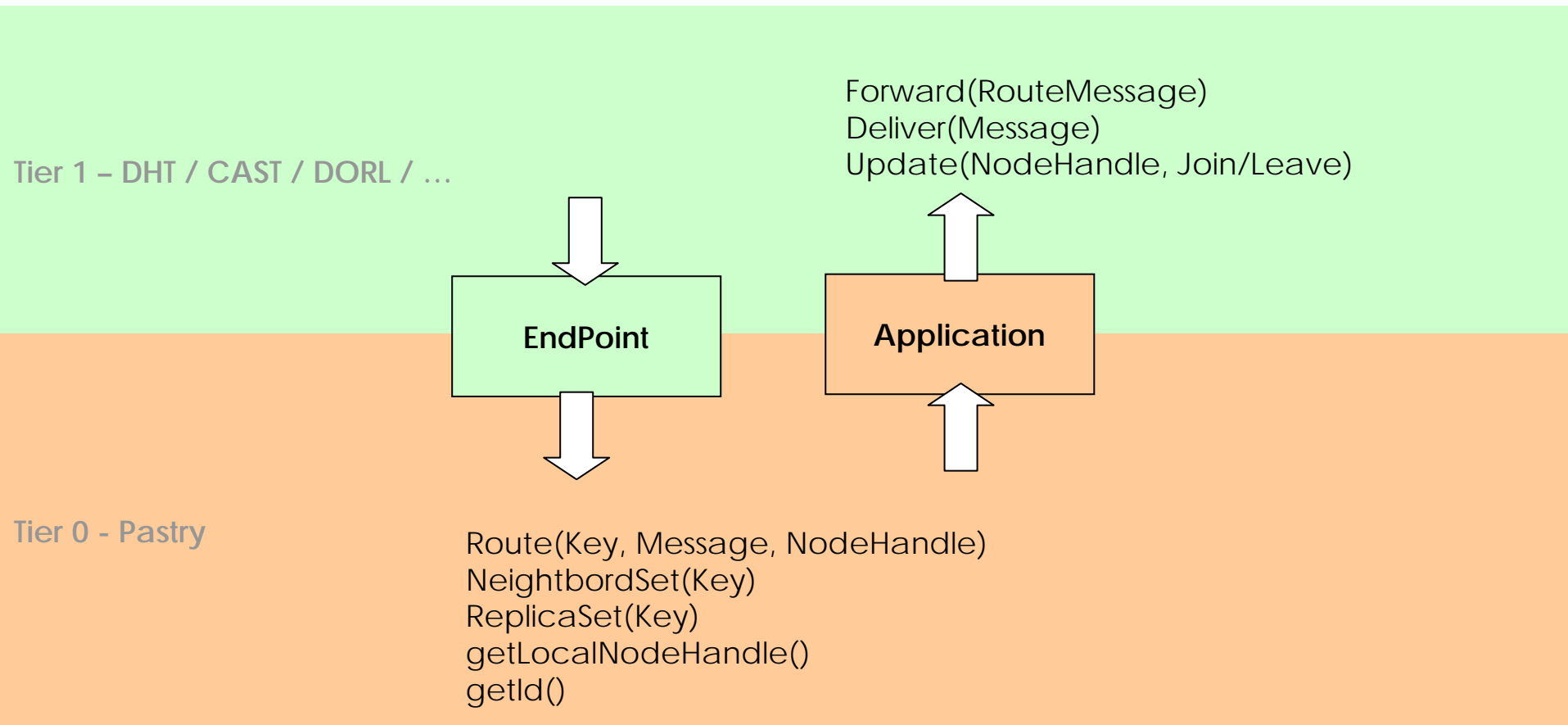


- Implementación en Java del overlay Pastry
- Sigue el contrato de la Common API
- Implementa también aplicaciones propias como Past, Scribe, Splitstream ...
- Última versión : 1.4.2
- <http://freepastry.rice.edu/>

Common API : Overview



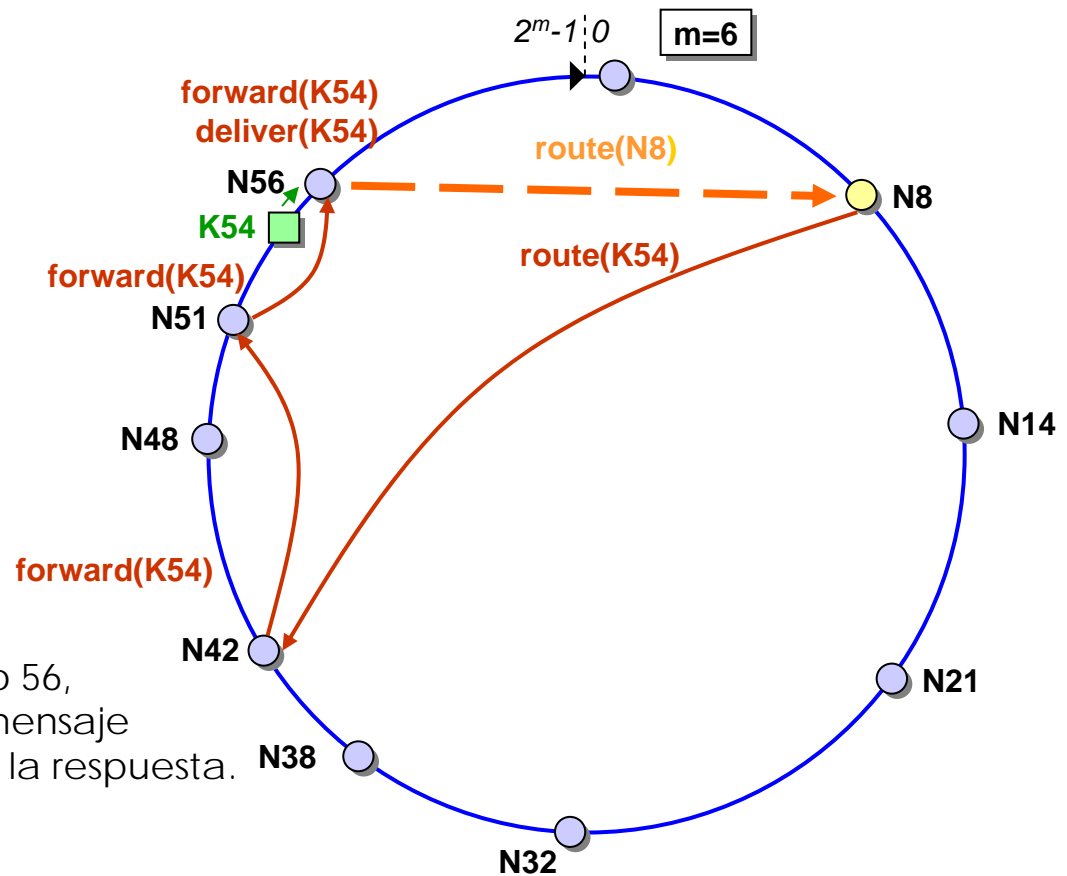
Common API en Freepastry



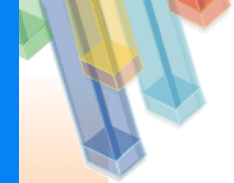
Routing según Common API



- 1) La aplicación del nodo 8 enruta un mensaje de petición sobre la clave 54
- 2) El mensaje llega al nodo 42, haciendo forward sobre él
- 3) El mensaje llega al nodo 51, haciendo forward sobre él
- 4) El mensaje llega al nodo 56, hace forward y como la clave 54 le pertenece también hace deliver
- 5) La aplicación, que corre en el nodo 56, recoge el NodeHandle (nodo 8) del mensaje para poder realizar routing directo de la respuesta.



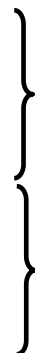
Servicios capa KBR y DHT



- Contrato capa Overlay KBR

- Forward(Message)
- Deliver(Message)
- Route(Key,Message, NodeHandle)

- Update(NodeHandle, Join/Leave)
- NeighbordSet(Key)
- ReplicaSet(Key)
- ...

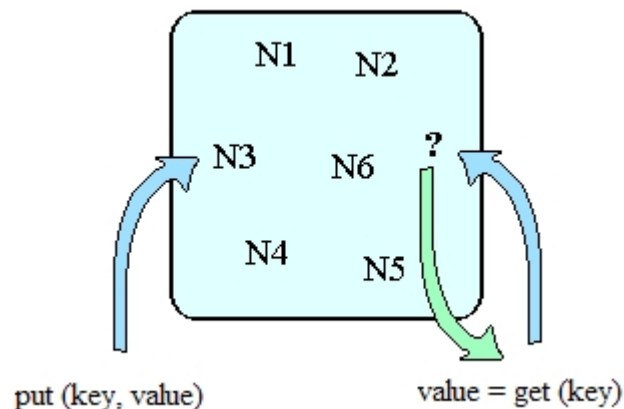


Servicios de routing

Servicios para la obtención de información de la red

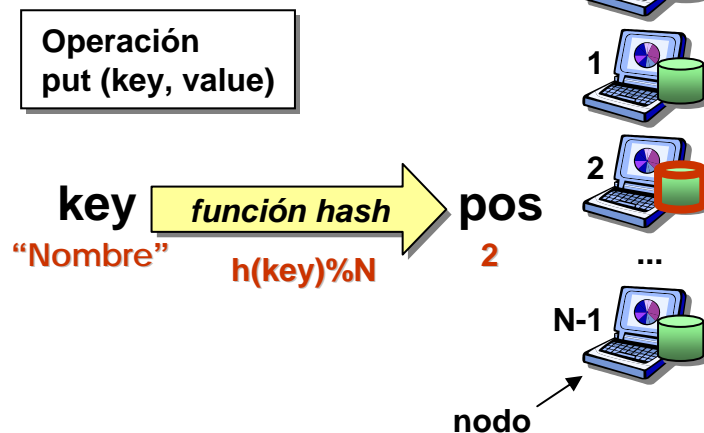
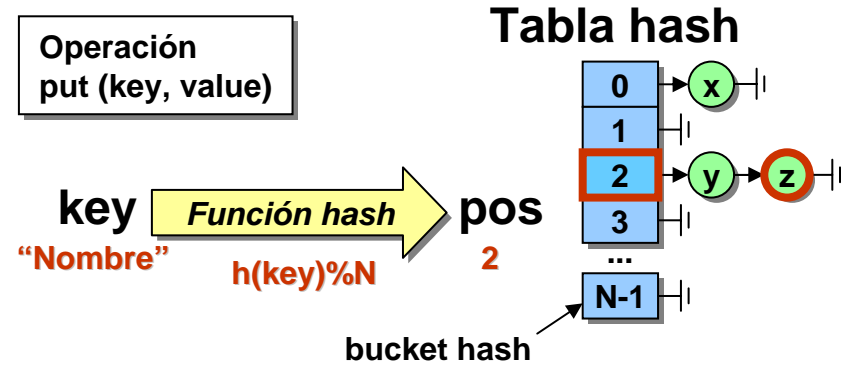
- Contrato capa DHT

- Put(Key,Value)
- Get(Key) => Value
- Remove(Key)



DHT : Distributed Hash Table

- Una tabla de hash asocia datos con claves
 - Se hace hash de la clave para encontrar *bucket* en tabla de hash
 - Cada *bucket* espera contener $\#elems/\#buckets$ elementos
- En una **Tabla de Hash Distribuida (DHT)**, los nodos son los *buckets* de hash
 - Se hace hash de la clave para encontrar el **nodo responsable**
 - Datos y carga se balancean entre los nodos



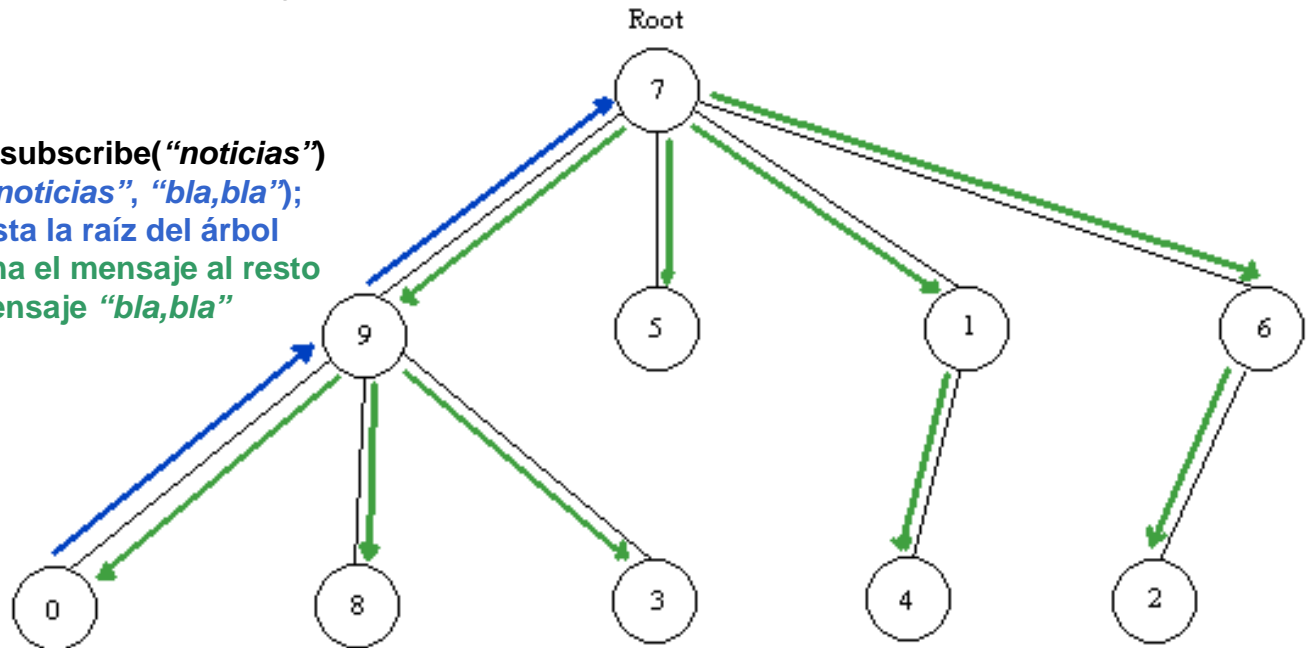
CAST : Scribe



- Canal de propagación de eventos wide-area con un servicio de publicación/suscripción por tópicos
- Construye un árbol de forma eficiente y propaga los eventos a todos sus nodos
- Si un nodo realiza una suscripción, este se enlaza con el nodo más cercano que este ya suscrito.

Ejemplo :

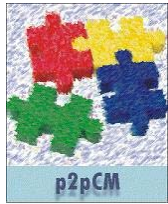
- 1) Todos los nodos -> `subscribe("noticias")`
- 2) `Nodo 0 -> publish("noticias", "bla,bla");`
- 3) El mensaje llega hasta la raíz del árbol
- 4) El nodo raíz disemina el mensaje al resto
- 5) Todos reciben el mensaje `"bla,bla"`





SNAP : Plataforma descentralizada p2p de aplicaciones web.

<http://snap.objectweb.org>



p2pCM : Framework de componentes p2p distribuidos.

<http://planet.urv.es/p2pcm>



DERMI : Middleware de objetos distribuidos sobre un canal de eventos p2p descentralizado, con servicio DOLR.

<http://dermi.sourceforge.net>



Bunshin : DHT con motor de búsqueda por palabras clave y soporte de Bittorrent (plugin Azureus).

<http://planet.urv.es/bunshin>