Poster: Persistent Content-based Publish/Subscribe Service On Top Of DHT

Yan Shvartzshnaider[1], Maximilian Ott[2], and David Levy[1]

[1] School of Electrical and Information Engineering
The University of Sydney, Australia
yshv6985@uni.sydney.edu.au
david.levy@sydney.edu.au
[2] National ICT Australia (NICTA)
max.ott@nicta.com.au

Approach

We propose design for a distributed and persistent content-based publish/subscribe service based on the Rete algorithm. Rete [1] has been designed to support pattern matching in production rule systems. In particular, Rete is able to manage, interpret and evaluate a set of rules against a large persistent dataset. The matching process relies on a loosely coupled dataflow matching network which makes the Rete algorithm a suitable candidate for implementation in distributed settings, such as with DHTs. In our approach we make the following key contributions to the previous work: efficient information filtering, persistence of subscriptions and publications, scalability, query expressiveness.

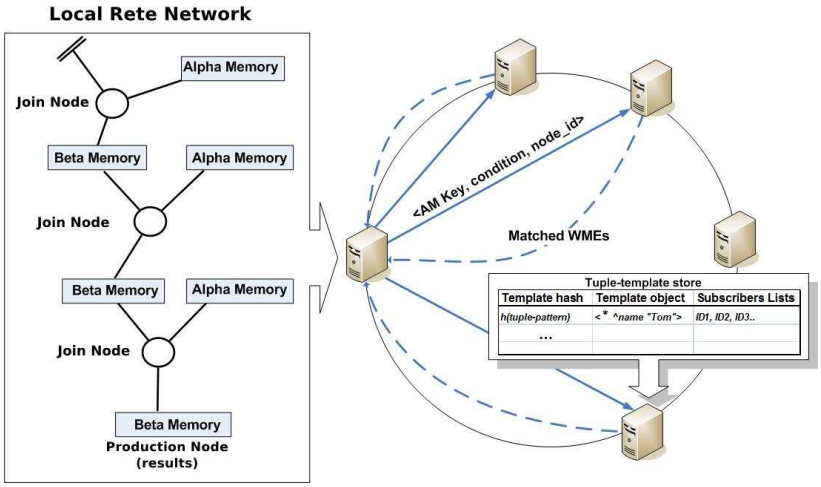

Fig. 1. An extremely simplified illustration of our implementation approach.

Rete generates a dataflow network from a given set of rules. This dataflow network consists of loosely coupled alpha memory (AM), beta memory (BM)

and join nodes. The alpha memories act as predicates on the dataset. With each new matched tuple *right activation* is triggered. The beta memories store the intermediate results from previous join node (note, these results can be shared by different subscriptions at the node). The join nodes perform variable binding tests between sets of tuples in the BM and data tuples in the AM, and outputs binds as variable tuples to the next BM. This process is called *left activation*. In other words, Rete's matching process is a chain of successively triggered right and left activations with the final result stored in the last node in the chained beta-memory, also called *production node*. The activations allow Rete dynamically to adjust its results to any incremental changes in the data.

Our approach. Similarly to Rete, our system treats tuples as primitives. Published events are tuples and subscription is essentially a set of bindings by variables rule tuples. Intuitively, we propose to follow the rendezvous model to distribute Rete's alpha-memories over the DHT nodes to arrange a meeting between each AM and matching to it published tuples. Each node in the network has the ability to subscribe to and publish content. All subscriptions at the subscribing node are managed locally by a single Rete network. As Figure 1 shows, our DHT's nodes store the subscription's rules and forward every matched tuple to appropriate subscriber nodes where on arrival the tuples are distributed among the alpha memories, triggering the matching process.

Contributions and Future work

We have presented a design for a distributed and persistent content-based publish/subscribe system on top of DHT. Our main contributions compared to other related work are: (1) *Persistence* – our system stores both subscriptions and published events. Hence a query can also return previously published, matching events. (2) *Subscription expressiveness* – subscriptions are defined as a set of conditions (rules) bound by variables. (3) Efficient information filtering due to the efficiency in pattern matching of the Rete algorithm. In our future work we plan to improve the current design to mitigate any potential load balancing problems due to non-uniform distribution of published events' attributes. We will analyse system's performance, and in particular, will compare the current push-based design against alternative pull- based approaches. Finally, our ultimate goal is to create a persistent and content-based pub/sub facility that can be used as a building block by distributed Internet application and services.

References

[1] Forgy, C.: Rete: A fast algorithm for the many patterns/many objects match problem. Artificial Intelligence **19**(1) (1982) 17–37