Popularity Estimation in a Popularity-Based Hybrid Peer-to-Peer Network

Marzieh Molaei Kamran*, Siavash Khorsandi**

*Islamic Azad University of Qazvin, Daneshgah Street, Nokhbegan Boulevard, Qazvin, Iran

**Amirkabir University of Technology, 424 Hafez Ave., Tehran, Iran

molaei@qiau.ac.ir, khorsandi@aut.ac.ir

Abstract— **The globally structured hybrid peer-to-peer systems combine structured and unstructured peer-to-peer networks where a structured ring based network forms the backbone of the system and multiple unstructured peer-to-peer networks are attached to this backbone. We show that if items are inserted into the system according to their popularity, search efficiency will be considerably improved. The key challenge in this scheme is how to identify rare items that should be inserted in the DHT. We analyze several approaches of identifying popularity of items in hybrid systems and propose an optimum approach for popularity estimation.**

Keywords— **peer-to-peer network; popularity; structured network; unstructured network; hybrid system**

I. Introduction

In recent years Peer-to-Peer (P2P) file sharing systems have received great attention and popularity. The organization of peers in P2P networks is divided in two different categories, namely structured and unstructured. Unstructured networks like Gnutella [1] do not have inherent restrictions on the network topology and locating resources. In structured networks like Chord [6] and CAN [7], topology is highly organized and items are placed at specified locations based on Distributed Hash Tables (DHTs). Hybrid peer-to-peer system combines these two types of peer-to-peer networks to benefit advantages of both of them. Two ways for combining the unstructured P2Ps with a structured DHT-based network are discussed bellow.

Globally Unstructured Hybrid Systems. Since Gnutella-like P2P is effective for locating highly replicated items but performs poorly to find rare ones. If all the ultrapeers are organized into the DHT overlay, this hybrid infrastructure can be implemented easily. Each ultrapeer would be responsible to identify and publish rare items from its leaf nodes. Flooding is used to find popular items, while rare items are found via DHT. SimpleHybrid [8,9] is a hybrid search infrastructure of this type that only uses local or neighbor information to determine items rarity. Instead, GAB [4] uses gossip to collect global statistics about popularity of items to make the search selection decision. However, they do not successfully deal with the dynamic nature of P2P systems. PASH [5] employs a network dynamic sensing infrastructure and uses the results to make accurate search selection decision.

Globally Structured Hybrid Systems. like [2,3] that a portion of nodes organized in a structured ring based network forms the backbone of the system and another nodes are locally organized in multiple unstructured peer-to-peer networks attached to the backbone and communicate with each other through the backbone. The core structured network provides an efficient and accurate lookup mechanism to forward the queried item within a certain unstructured network, while local unstructured networks provide a flexible and robust mechanism for peers to join or leave the system freely.

In this paper we focus on globally structured hybrid peer-to-peer systems. It is composed of a core structured network and many unstructured networks. The structured network, called *S-Net*, is a ring based peer-to-peer network similar to a chord ring. Each peer in the S-Net namely *SN-peer* joins the ring, based on its ID. Each locally unstructured network is called *U-Net*. Peers in a U-Net namely *UN-peers* are inserted to U-Nets according to their IDs. Figure 1 shows the overview of the system. For load balancing when the degree of a peer reaches a threshold δ, it selects one of its neighbors randomly, and passes the join request to it. The SN-peer sends join request of new UN-peer along a random branch until it reaches a peer that its degree is less than δ to which it will connect. This peer is called the *connect point (cp)* of the new UN-peer. Each UN-peer maintains the address of the SN-peer of the U-Net and its *cp,* besides the neighbor list. The UN-peer whose *cp* has leaved the system, should send a join request to the SN-peer again to rejoin the U-Net. The leaving UN-peer should transfer the load to one of its neighbors.

Each data item is assigned an ID namely *d-id* by hashing the data key. Each U-Net is responsible for the items whose *d-id*s lie in a specified range of ID space. Both data insertion and lookup requests try the local U-Net first to reach the desired item. If the item is not served by the local U-Net, requests will be forwarded along the S-Net until it reaches the U-Net serving the queried item.

We show that if items be inserted in the system based on their popularity, search efficiency improves considerably compared to Simple system introduced in [2]. The key challenge in this scheme is correctly identifying the rare items that should be inserted in the DHT. So, in this work we analyze approaches of identifying popularity of items in this

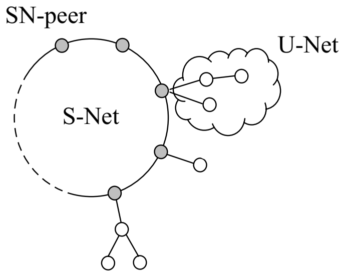


Figure 1. A globally structured hybrid P2P system

type of hybrid systems and propose an optimum approach for popularity estimation.

The rest of the paper is organized as follows. In Section 2 we describe popularity-based system. In Section 3 we present approaches of popularity estimation and propose a new approach. Section 4 shows experimental study and finally we conclude this work in Section 5.

II. Popularity-Based Hybrid P2P System

Popular items that are highly replicated and easy to found could be inserted into unstructured networks, and rare items into core structured network to be found based on their IDs. In this system when a SN-peer receives a data item insertion request, if the popularity of the item exceeds a threshold defined as T_p, it will be sent to the U-Net. Otherwise, if the popularity of the item is equal or less than T_p and SN-peer does not have the item, then it will be inserted into the database of the SN-peer, but if the SN-peer already has the item, the new item will be sent to the U-Net. If the number of rare items that SN-peer retains exceeds a limit, the SN-peer can reduce T_p and send items that have higher popularity to its neighbors. For inserting rare items into the S-Net, each SN-peer needs a mechanism to identify the popularity of new shared items.

III. Popularity Estimation

In hybrid peer-to-peer systems, different heuristics have been used for popularity estimation, such as query results size, term frequency, term pair frequency, sampling of neighboring nodes, and historical statistics on item replicas [8]. As mentioned before, in SimpleHybrid [8, 9] first a query is flooded with a limited depth, and if no result is returned, the query is forwarded to the DHT. Such a design is simple but inefficient because it incurs extra overhead. Some most recent works like GAB [4] and PASH [5] have estimated popularity according to historical statistics on item replicas. In GAB when an ultrapeer receives a document title it has not indexed before, it tosses a coin up to k times until the first tail appears. It stores the number of heads as a value namely CT. The ultrapeers gossip their CT values for all titles with each other. Then each ultrapeer, computes the maximum value of CT for each title. The maximum value of CT represents the popularity of document. Aiming at improving the hybrid search in dynamic environments, PASH employs a network dynamic sensing infrastructure and uses the results to make the search decision.

In this work, we use three ways to estimate popularity, based on historical statistics on item replicas: *Centralized*, *Decentralized* and *Hybrid* approaches. In Centralized approach, when a SN-peer receives data insertion requests, it saves the frequency of meeting each item that indicates the popularity of that item. In Decentralized approach, the information of items stored in the U-Net is gathered at the beginning of each time period denoted by θ. Hybrid approach combines Centralized and Decentralized manner, as, at the beginning of θ, each SN-peer obtains the popularity of the items of its U-Net using Decentralized approach, and during θ, SN-peer updates this popularity information using Centralized approach. In the rest of this section we describe each approach in more detail.

A. Centralized Approach

All of new items must be served by the SN-peer before entering into the U-Net, so the frequency of meeting each item can be computed by the SN-peer over each time period T_i. Then, SN-peer makes decision for the items that join the U-Net over the T_{i+1}. The items that have been inserted into the system before, with high probability exit the system after some time, because of the nodes failure. So T should not be too long to contain only new items. On the other hand, it should be enough long to show plenty of items. Moreover, updating time period can be shortened by dividing T into k parts: $\theta_1, \theta_2, \theta_3, \ldots, \theta_k$. We define d_i as new items entering the system during θ_i, and D as new items entering the system during T, so

$$D = \sum_{i=1}^{k} d_i$$

In the end of θ_{k+1}, the new items d_{k+1} have been inserted into the system, so they must be added to D. On the other hand, the oldest items d_1 must be eliminated from D. Thus, each SN-peer always has history of new items entering the U-Net at recent k periods of time. So

$$D_{new} = D + d_{k+1} - d_1$$

Figure 2 shows above operation. Hence, T is long enough to show plenty of items, and also has new items because of the shorter updating time period θ. If a peer shares an item whose *d-id* belongs to the current U-Net, it stores the item itself and informs the SN-peer about this new shared item. In the Centralized method, SN-peers make decision only according to the recent items entering the system, while some items entering during T, exit the system before T terminates, due to the nodes failure. On the other hand, some items that have entered before T are still in the system.

B. Decentralized Approach

In Decentralized method, the number of replicas of the items stored in the U-Net is gathered at the beginning of each time period θ. Each peer informs its cp about the items that has stored. The cp adds the information of its items, and sends

d_1

A	B	C
1	3	2

d_2

A	D
2	1

D

A	B	C	D
3	3	2	1

d_3

B	D
1	2

$D_{new} = D + d_3 - d_1$

A	B	D
2	1	3

Figure 2. Computing D_{new} in Centralized approach

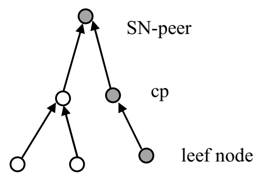


Figure 3. Gathering information using Decentralized approach

total information to its own *cp*. Finally, all the information of the U-Net is gathered in the SN-peer as in figure 3. Using long θ leads to inaccurate estimation due to the frequently joining and leaving of the peers, and short θ generates a large amount of traffic.

C. Hybrid Approach

Each SN-peer is aware of the new items entering to the U-Net, but does not know which items have left the system because of the nodes failure. For each peer, there are three occurrences:

- *Join.* When a peer joins the network, each shared item is forwarded along the S-Net until it arrives at a proper SN-peer to be inserted into the U-Net. So, the SN-peer is aware of the new items and can update information of new items using Centralized approach. If a peer shares an item whose *d-id* lies in the current U-Net, it informs the SN-peer about the new item.
- *Leave*. Since each peer, leaving the system, transfers the load to a neighbor, its items remain in the U-Net, and there is no need to update.
- *Fail.* When a node fails, its items must be eliminated from the information of the SN-peer. If failed node is a SN-peer, the information of the U-Net is missed too. In this case, new SN-peer can use the Decentralized approach to gather the information of the U-Net.

In Hybrid approach, at the beginning of each long time period θ, the information of the U-Net is gathered using Decentralized approach, and at the beginning of each short time period τ, SN-peer makes updates comprised of (i) adding the information of new items entering the U-Net, (ii) eliminating the information of the items of failed nodes, (iii) adding the information of new shared items whose *d-id*s belong to the current U-Net.

We need the following definitions

- $p(t)$: The number of replicas of item i at time t.
- p_{new_s}: The number of times that SN-peer forwards new replicas of item i to the U-Net.
- p_{new_u}: The number of new replicas of item i entering the system during τ that belongs to the current U-Net.
- P_{exit}: The number of replicas of item i that exits the system during τ because of the nodes failure.

In each U-Net we will have

$$p\,(t+\tau) = p\,(t) + \Delta\, p$$

$$\Delta\, p = p_{new\_s} - p_{exit} + p_{new\_u}$$

The SN-peer needs p_{new_s}, p_{exit} and p_{new_u} to update the number of replicas of item i. The SN-peer itself is aware of p_{new_s}. If *d-id* of the new shared item lies in the current U-Net, the peer informs the SN-peer about the new item. In this way, p_{new_u} could be computed. For accessing p_{exit}, each *cp* retains replica counts of items stored in neighboring nodes, and if some neighbors fail, *cp* informs the SN-peer about the items of the failing nodes directly, but if the node failure ratio is high, another node can collect the information for SN-peer. This node, namely *collector*, is selected by SN-peer among its neighboring nodes. To reduce the traffic cost, when the items of a node change, updating messages are exchanged between that node and its *cp* instead of PING/PONG messages.

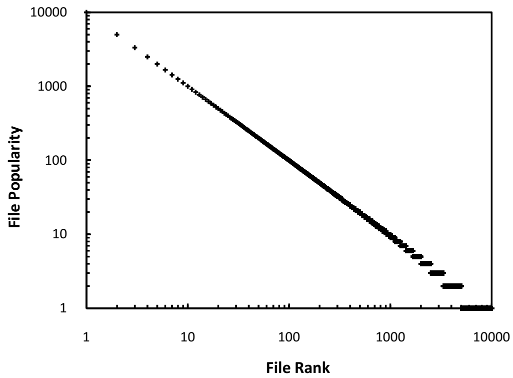
IV. SIMULATION

We extended PlanetSim [10] to simulate the popularity-based globally structured hybrid system. Networks used in the simulations are composed of 1000 nodes. The insertion of the items is assumed to have Zipfian popularity distribution with two Zipf parameters of 1.0 and 0.6, as shown in figure 4. In each round of running simulation, one peer fails and one peer leaves the network. The size of the networks remains fixed with joining new nodes, each bringing n items into the system, randomly chosen from a dataset of 10,000 unique data items that have a Zipf popularity distribution. The value of TTL in U-Nets is set at 1 and δ is set at 2. A system parameter namely P_u is defined as the proportion of UN-peers in the system.

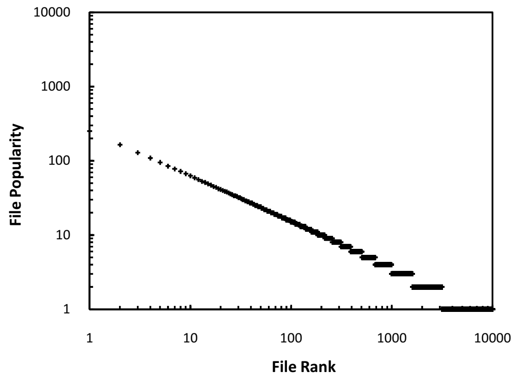
A. Evaluation of Popularity Estimation

When the value of P_u is large, the average number of nodes in each U-Net increases, so, in this case, using an efficient popularity estimation approach has higher impact on the performance of the system. Here, results are obtained from the networks with P_u=0.9. Results obtained in this section do not depend on the Zipf parameter, so we show only results of Zipf parameter of 0.6. We use the following metrics to evaluate the popularity estimation methods.

Estimation error. Here we compare Centralized, Decentralized and Hybrid approaches in the estimating accuracy of item popularity. We define the estimation error ε by $\varepsilon = (E - P) / P$, where E is estimation value and P is the exact popularity value of an item. In the Centralized approach,



(a)



(b)

Figure 4. Distribution of 10000 distinct file (a) Zipf 1.0 (b) Zipf 0.6

T must present plenty of items, and it must be constant for different values of θ (here, the value of *T* is 1000 rounds).

According to figure 5, in all three approaches for small values of θ, estimation error is small, and for large values of θ, estimation error is large. The estimation error of the Centralized approach is larger than other approaches because updating popularity values is only based on the latest new items inserted to the system, without considering items that exit the system. Some items inserted to the system earlier, are still in the system, but some others that are inserted later, exit the system due to the node failures. In Decentralized approach when θ is small, estimation error is also small but the traffic overhead increases. In Hybrid approach, estimation error is totally low, because during θ, information is updated $\frac{\theta}{\tau}$ times (if θ ≥ τ). If a node and its *cp* fail during the same τ, the information of that node will be lost, so some errors occur in estimating the popularity of items of that failed node and these errors remain until the end of θ that SN-peer updates its popularity values of its U-Net using Decentralized approach, or until when the SN-peer fails and then new SN-peer uses Decentralized approach. For larger values of τ, this kind of error occurs more, because in that case, probability of failing a node and its *cp* in a same τ is higher. For smaller values of θ and τ, estimation error is lower than larger values. In Hybrid approach, when θ is less than τ, results are similar to Decentralized approach, because Decentralized approach is practically done instead of Hybrid approach. In this case, Hybrid approach acts a little better than Decentralized approach, because when a SN-peer fails, popularity of items is gathered for new SN-peer of the U-Net.

Average number of queries. Figure 6 shows a comparison of the number of queries generated for popularity estimation in 1000 rounds of running simulation. In Centralized approach, number of queries for different values of θ is uniform and is less than the other approaches, because only when a node wants to share an item whose *d-id* lies in the range of the current U-Net, informs the SN-peer. During 1000 rounds, number of items inserted to the system is constant. If there are *N* nodes in the network, Each new item is inserted to the current U-Net with probability $Prob_{in\_u}$ as follow

$$Prob_{in\_u} = \frac{1}{N(1-P_u)}$$

So the number of queries in Centralized approach does not depend on the value of θ. In Decentralized approach, estimation error for large values of θ is higher than small values, but using small values of θ, generates more number of queries. For example when θ is equal to 1 or 10, estimation error is low but traffic overhead is high. In Hybrid approach, for small values of τ, traffic overhead is low, even if θ is large. In this case, estimation error is also acceptable, so with small values of τ and large values of θ, Hybrid approach outperforms Centralized and Decentralized approaches in terms of both query traffic cost and estimation error. When the value of θ is constant, for larger values of τ, traffic overhead is a little lower than smaller values of τ. One reason is that when τ is large, a node and its *cp* may fail during a same τ with higher probability, and so the *cp* of the failed node is not in the system to inform the SN-peer and the number of queries reduces. Another reason is that when τ is large, each *cp* informs the SN-peer about the changes once each long time period, but during a same time, when τ is small, the changes should be sent several times.

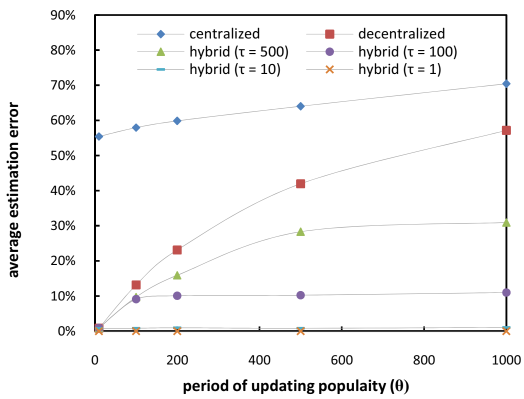


Figure 5. The comparison of the estimation error

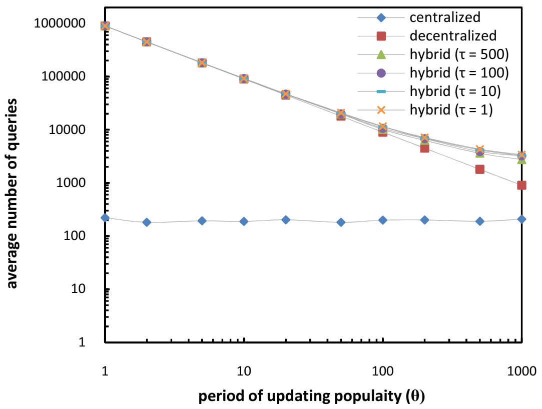


Figure 6. The comparison of the number of queries

B. Lookup Failure Ratio in Dynamic System

According to the results obtained from our previous experiments, here we use Hybrid approach for popularity estimation, and parameters are set as θ=1000, and τ=10. In this section, during 1000 rounds of running simulation, we perform 10^6 random lookups, and when a node leaves or fails, a new node joins the system, so the size of the network stays fixed.

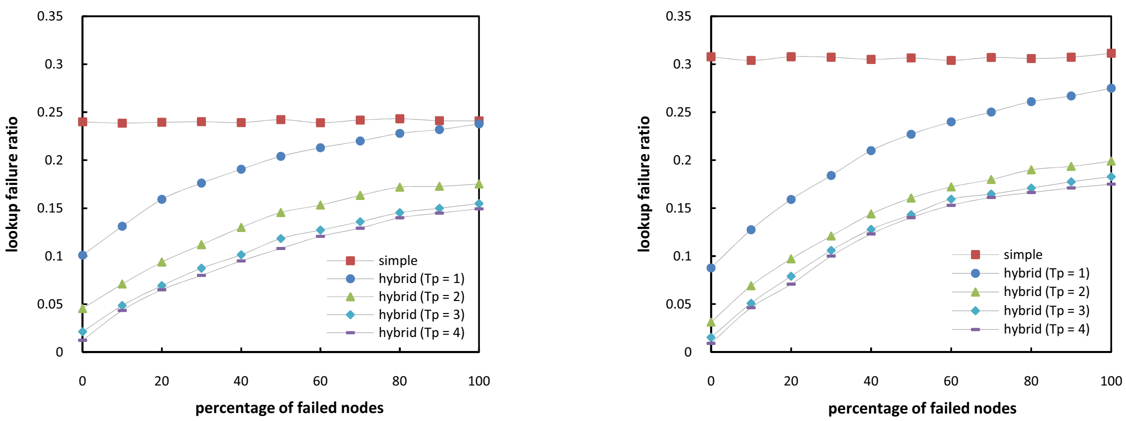
Figure 7 shows lookup failure ratio under different percentage of failed nodes during 1000 rounds. The results show that lookup failure ratio of popularity-based system is lower than Simple system in dynamic network. In both figures 8(a) and 8(b), lookup failure ratio decreases with increasing T_p, because in this case, more rare items are stored in the SN-peers. The difference of the lookup failure ratio, between Simple system and popularity-based system in figure 7(b) is larger than that in figure 7(a), because as shown in figure 4, there are more rare items in zipf 0.6 distribution compared to zipf 1.0, so popularity-based system has reduced lookup failures in figure 7(b) much more than that in figure 7(a). In popularity-based system, when percentage of failed nodes is large, lookup failure ratio increases.

In this case, when number of failed nodes increases, more number of SN-peers also fails, and since rare items are inserted to the SN-peers, so many rare items exit the system with more SN-peer failures, and so lookup failure ratio increases. In this simulation, all nodes fail with the same probability. If SN-peers had longer uptimes, lookup failure ratio was lower. In Simple system, lookup failure ratio is uniform, because rare items are distributed among all nodes, and lookup failure ratio is not influenced by the dynamicity of the system. The difference of the lookup failure ratio between Simple system and popularity-based system in figure 7(b) is higher than figure 7(a), because the number of rare items with Zipf 0.6 distribution is more than Zipf 1.0, so inserting items based on their popularity, has a greater impact on the performance of the system.

V. Conclusion

In this work we have inserted items to a globally structured hybrid peer-to-peer network based on their popularity. Popular items can be found easily in unstructured networks using flooding-based searches, so we insert them to the unstructured networks. On the contrary, rare items are inserted to the structured network because DHTs guarantee perfect recalls and good response time for rare items.

We also analyze some popularity estimation approaches for this kind of hybrid peer-to-peer networks, based on historical statistics on item replicas: Centralized and Decentralized approach. Then we have proposed an optimal approach namely Hybrid approach that combines Centralized and Decentralized manner, as, information of items in the U-Net is gathered using Decentralized approach at the beginning of each time period, and SN-peer updates its popularity information according to the Centralized approach during the time period. Our analytical evaluation has shown that Hybrid approach provides accurate popularity estimation and efficiently saves query traffic cost.



(a) (b)

Figure7. The comparison of the lookup failure ratio in dynamic system (a) Zipf 1.0 (b) Zipf 0.6

REFERENCES

[1] Gnutella, "http://gnutella.wego.com".

[2] Min Yang, Yuanyuan Yang, "An efficient hybrid peer-to-peer system for distributed data sharing," IPDPS 2008: 1-10.

[3] Stefan Zoels, Zoran Despotovic and Wolfgang Kellerer, "On hierarchical DHT systems – An analytical approach for optimal designs", Computer Communications, Volume 31, Issue 3, 25 February 2008.

[4] M. Zaharia and S. Keshav, "Gossip-based Search Selection in Hybrid Peer-to-Peer Networks," In Proceedings of IPTPS, 2006.

[5] X. Shi, J. Han, Y. Liu, L.M. Ni, "Popularity adaptive search in hybrid P2P systems", Journal of Parallel and Distributed Computing (2008).

[6] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Transactions on Networking, vol. 11, no. 1, pp. 17– 32, 2003.

[7] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, "A scalable content addressable network," In Processings of the ACM SIGCOMM, pp. 161–172, 2001.

[8] B. T. Loo, J. M. Hellerstein, R. Huebsch, S. Shenker, and I. Stoica, "Enhancing P2P File-Sharing with an Internet-Scale Query Processor," In Proceedings of VLDB, 2004.

[9] B. T. Loo, R. Huebsch, I. Stoica, and J. M. Hellerstein, "The Case for a Hybrid P2P Search Infrastructure," In Proceedings of IPTPS, 2004.

[10] P. Garc, C. Pairot, R. Mondar, J. Pujol, H. Tejedor, R. Rallo, "PlanetSim: A new overlay network simulation framework," In Software Engineering and Middleware, SEM 2004, in: LNCS, vol. 3437, Springer, 2005, pp. 123_137.