KIPS Trans. Comp. and Comm. Sys.
Vol.8, No.9 pp.209~216 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2019.8.9.209

Hierarchical Message Forwarding Scheme for Efficient Data Distribution in P2P Messaging System

Jin Sun Jung† · Sangyoon Oh††

ABSTRACT

Publish-subscribe communication model is popular for various type of distributed applications because of its loosely coupled style connections. Among the various architecture style for publish-subscribe system, peer-to-peer architecture has been used for the mission critical application domain since it provides high scalability and real-timeness. On the other hand, to utilize the bandwidth of given networks, message filtering is frequently used to reduce the number of messages on the system. Even if P2P provides superior scalability, it is hard to apply filtering to the its messaging system because the filtering process should be done on the peer-side in P2P architecture that are usually done on the broker server in conventional pub/sub architecture. In this paper, we propose a hierarchical subscription management structure as well as message forwarding scheme for efficient data dissemination. Our proposed scheme reduces the number of received messages by filter-out un-wanted messages and offloading the message dissemination work to other subscribers to enhance the messaging throughput.

Keywords : Hierarchical Structure, Publish-Subscribe Model, Message Filtering, Peer-to-Peer(P2P)

P2P 출판-구독 메시징 시스템에서 효율적인 정보 전파를 위한 계층적 메시지 전송 기법

정 진 선† · 오 상 윤††

요 약

출판-구독 모델은 정보 생산자와 사용자를 느슨하게 연결해 주어 다양한 기기들의 연결에 많이 사용되고 있다. 출판-구독 모델에서 네트워크 대역폭을 효과적으로 활용하기 위한 방안으로 조건에 따른 메시지 필터링이 사용되고 있으며, 이에 따른 다양한 연구 결과들이 제안되어 왔다. Peer-to-Peer(p2p)기반의 출판-구독 모델은 높은 확장성을 장점으로 다양한 분야에서 널리 사용되고 있지만, p2p 기반특성에 따라 일반적인 필터링 과정이 구독자단에서 이루어지게 되어 구독자에게 추가적인 성능 부담으로 작용하고 있다. 본 논문에서는 이러한 문제점들을 해결하기 위해 subscription사이의 종속 관계를 가지는 계층적인 subscription 구조와 이를 기반으로 하는 계층적 메시지 전송 방식을 제안한다. 본 제안을 통해 subscription 부모가 존재하는 구독자들은 부모로 부터 필터링 된 메시지를 전송 받게 되어 메시지 수가 증가하더라도 상대적으로 적은 수의 불필요한 메시지를 수신하게 되며, 구독자들 간 트리 구조를 통해 메시지 전송 과정을 출판자에서 구독자로 분산시킴으로써 속도 측면에서의 성능 향상을 얻을 수 있다. 제안 기법의 검증을 위해 제안 기법과 기존 기법들 간의 비교 실험을 진행하였으며, 본 제안 기법에서 메시지 유통량과 전체 throughput의 향상을 확인하였다.

키워드 : 계층 구조, 출판-구독 모델, 메시지 필터링, 피어투피어

1. 서 론

분산 환경에서 프로세스 간 데이터 공유를 위해서 다양한 방법들이 사용되고 있다. 그들 중 정보 생산자와 소비자를 직접 연결하는 직접 통신으로서 클라이언트-서버 패러다임과 간접 통신으로서 message queue와 publish-subscribe (pub-sub) 패러다임이 많이 사용되고 있다. 출판-구독(publish-subscribe) 패러다임은 message queue와 같이 출판자(publisher, 정보 생산자)와 구독자(subscriber, 정보 수신자)간의 시간과 공간에 대한 느슨한 결합을 가지는 장점을 가지며, 구현 구조에 있어서는 다대다 연결이 가능하도록 브로커를 두는 구조와 참여자(publisher와 subscriber)들 간 P2P(Peer-to-Peer) 연결을 제공하는 구조를 사용한다.

브로커 구조에서는 출판자와 구독자 사이에 브로커가 존

※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 기초연구사업임(2018R1D1A1B07043858).
† 비 회 원 : 아주대학교 소프트웨어학과 학사과정
†† 종신회원 : 아주대학교 소프트웨어학과 교수
Manuscript Received : December 17, 2018
First Revision : March 25, 2019
Accepted : May 8, 2019
* Corresponding Author : Sangyoon Oh(syoh@ajou.ac.kr)

재하며, 출판자가 보낸 메시지를 구독자에게 전송하는 라우팅의 기능을 수행한다. 브로커는 구독자의 구독 의사인 subscription 기반으로 라우팅을 수행하며, 구독자가 구독의사를 토픽 기반(topic-based)으로 하는 경우는 구독자의 topic을, 그리고 구독자의 구독 의사가 조건으로 나타난 컨텐츠 기반(contents-based)의 출판-구독에서는 구독자의 조건을 subscription으로 가지고 있어야 한다. Subscription 내용에 따라 구독자는 출판자가 전송하는 메시지들에 대해 필터링을 적용하는 것이 가능하다. 예를 들어, 기기 간 연결을 특징으로 하는 IoT(Internet of Things)기기 중 특정 기기가 온도에 대한 정보만을 수집하고 싶을 경우, 토픽 기반 모델에서 구독하고자 하는 토픽을 '온도'로 설정하여 필터링을 수행한다. 또한 온도 정보 중에서 특별히 10도 이상의 정보만을 수집하고 싶을 경우 컨텐츠 기반 구독 모델에서 subscription의 속성을 10도 이상으로 설정하게 된다.

비 분산 단일 브로커 구조의 경우에 구독자들의 subscription 유지 및 관리가 단순하여 신뢰성 및 데이터 일관성이 경우에 적용되는 구조이지만, 반면에 브로커가 단일 실패점(Single-Point-Of-Failure) 혹은 병목 현상(Bottleneck)과 같은 시스템 장애 요소가 될 수 있다는 단점을 가지고 있어 확장성이 중요한 경우에 적용이 어렵다. 이에 대해 분산 복수 브로커 구조는 브로커들 사이에 네트워크를 형성하여 단일 실패점 및 병목 현상 등에 대해 유연한 대처가 가능하다. 그러나 분산 복수 브로커 구조는 브로커 간에 subscription 데이터의 일관성을 유지해야 하는 어려움을 가진다. 복수 브로커들은 협업을 통해 출판자로부터 전송 받은 메시지를 구독자의 subscription에 기반하여 구독자에게 정확히 전달해야 하는데, 이를 위해 모든 브로커는 항상 동일한 구독자의 subscription에 있어서 데이터 일관성을 유지해야 한다. 하지만, 브로커의 수가 많아질수록 데이터 일관성을 유지하는 것이 어려워지기 때문에 구독자의 출입이 잦거나, subscription의 변동이 잦을 경우 분산 복수 브로커 구조의 출판-구독 메시징 시스템은 오히려 확장성이 나빠질 수 있다.

P2P(Peer-to-Peer)구조의 출판-구독 모델은 브로커를 가지 않으므로, 기존 브로커의 역할을 피어(peer)들이 수행해야 한다. 브로커가 존재하지 않으므로 비 분산 단일 브로커 구조의 중앙 집중형 출판-구독 모델이 가지던 단일 실패점이나 병목현상 등의 단점이 해결되며, 또한 분산 복수 브로커 구조가 가지는 구독자들의 subscription에 대한 데이터 일관성 유지 문제도 해결할 수 있어서 높은 확장성을 가지게 된다. 이러한 장점으로 인해 P2P 기반의 출판-구독 모델이 국방, 교통, 에너지 등 다양한 분야에서 활용되고 있다. 그러나 P2P 구조의 출판-구독 모델은 브로커 기반의 모델들과 다른 문제들이 존재하여 이에 대한 해결이 필요하다. 먼저 브로커가 존재하지 않기 때문에, 참여하는 피어들 간에 P2P 기술을 응용하여 찾기(discovery)가 필요하며 P2P 통신 기법을 사용하여 정보를 전달하게 된다. 이는 다대다 통신에서 부담(overhead)로 작용하게 되며, 참여자 즉 피어의 수가 많아지는 경우 상당한 부담이 되는 것이 확인되었다. 또한 브로커 기반의 출판-구독 모델에서 브로커의 역할을 P2P 모델의 피어들이 수행해야 하므로, 컨텐츠 기반의 subscription 모델의 경우 조건에 맞는 메시지 필터링의 과정을 구독자가 하게 되어 각 피어들의 부담으로 작용한다.

본 연구에서는 이와 같은 P2P 기반의 출판-구독 모델에서 나타나는 문제점을 해결하기 위해서 subscription사이의 종속관계를 가지는 계층적인 subscription 구조를 제안한다. 본 제안 구조를 통해 P2P 기반의 출판-구독 모델에서는 구독자가 가지는 메시지 처리 부담을 해결하며, 출판자만이 담당하던 메시지 전송의 과정을 구독자에게 계층적으로 분산시켜 정보의 전파 성능 향상 효과를 얻도록 하였다. 기존 연구 중 CUPUS[1]와 같이 본 연구제안과 같은 계층적 필터링과 유사한 제안이 있으나, 이는 브로커 기반의 출판-구독 모델에서 설계 및 구현으로서 본 연구가 목표로 하는 P2P 기반의 출판-구독 환경에는 적용이 불가하다.

이후의 논문은 다음과 같은 구조로 구성되어 있다. 2절에서는 본 연구에 대한 구체적 설명을 위해 출판-구독 모델에 대하여 소개하고, 필터링과 관련되어 수행된 관련성이 높은 최신 연구들을 분석한 결과에 대해 기술한다. 본 연구가 제안하는 계층적 subscription 모델과 이에 대한 수행 과정에 대한 자세한 설명은 3절에서 제시되었으며, 제안의 타당성을 검증하기 위해 연구 제안을 구현하고 기존의 방식과 비교한 실험 및 실험 결과가 4절에서 설명된다. 본 연구의 결론과 향후 연구에 대해서 5절에서 설명한다.

2. 연구 배경 및 관련 연구

2.1 Subscription 모델 종류

출판-구독 모델에서, 구독자의 subscription 모델의 종류는 토픽 기반(topic-base), 컨텐츠 기반(contents-base), 타입 기반(type-base)[2]로 나뉜다. 토픽 기반 혹은 채널 기반모델은 구독자가 구독-토픽을 정하게 되며, 해당 토픽으로 메시지가 출판되면 해당 토픽을 구독하는 모든 구독자는 메시지를 받게 된다. 이 경우에 출판-구독 모델에서 출판자와 구독자와의 관계는 느슨하므로 출판자가 구독자의 정보를 알지 못해도 구독자는 메시지를 받을 수 있다. 컨텐츠 기반 모델은 토픽 기반의 모델보다 subscription의 표현에 있어서 정밀하다. 특정 토픽을 정하여 구독 의사를 표현하는 것이 아니라, 출판된 메시지들 중 subscription의 조건에 부합하는 컨텐츠를 담고 있는 메시지만을 전달받는다. 따라서 컨텐츠 기반 모델은 토픽 기반의 모델보다 구독 의사 표현을 더욱 정밀하게 하여 토픽 기반 모델이 가지는 subscription 표현의 한계를 극복할 수 있으나, 이 컨텐츠 기반의 모델을 구현하기 위해서는 구현의 복잡도 또한 증가한다. 컨텐츠 기반의 모델은 주로 사람이 쉽게 이해하고 사용할 수 있는 부등호 등의 기호 혹은 정규표현(regular expression)를 사용하여 subscription을 나타낸다.

토픽 기반 모델에서 토픽들 간 관계가 있어 하위 토픽이 존재할 수 있고, 컨텐츠 기반 모델에서는 subscription은 부등호와 같이 범위를 나타내는 기호로 표시할 수 있으므로 토픽

기반, 컨텐츠 기반 subscription은 모두 포함관계가 존재할 수 있다. 예를 들어, 토픽 기반의 subscription의 경우, '수학'이라는 토픽은 토픽이 '방정식'인 것을 포함할 수 있으므로, 이를 나타내는 subscription간의 포함관계가 생길 수 있다. 또한 컨텐츠 기반의 모델인 경우, subscription이 'Temperature > 10'이 'Temperature > 20'의 범위를 포함하므로 포함 관계로 나타낼 수 있다. 이는 타입 기반의 모델에서도 적용된다.

2.2 출판-구독 아키텍처 종류

브로커를 두는 구조의 출판-구독 모델은 출판-구독 모델의 특징인 느슨한 결합을 통해 얻을 수 있는 장점인 확장성을 향상시키기 위해 다양한 방법이 제안되었다. 그 중 하나로 Kafka에서 사용하는 Zookeeper를 들 수 있다. 사용자들의 검색 기록, 통계 기록 등의 Log기록을 처리하기 위해 설계 및 구현된 Kafka[3-4]는 브로커 클러스터를 형성한 뒤 Zookeeper를 통해 브로커의 추가, 삭제 등 관리를 용이하게 함으로써 확장성의 한계를 해결하려 하였다. 또한, Kafka의 주요한 특징으로는 브로커가 구독자의 상태를 저장하지 않는다는 점이다. 브로커는 구독자의 상태(예: 구독자가 어느 메시지까지 읽었는지)를 유지하지 않고[3-4] 구독자가 이를 스스로 관리함으로써, 브로커가 유지해야하는 정보량을 줄이고 관련된 오버헤드를 줄여 성능 향상을 달성한다.

P2P기반의 출판-구독 모델은 단일 실패점(SPOF)과 병목현상(bottleneck)에 대한 유연한 대처를 할 수 있어서 브로커 구조와 비교하여 확장성이 뛰어나다. P2P 구조의 출판-구독 모델의 대표적 표준 및 구현체로 DDS(Data Distribution Service)가 있다. DDS는 확장성, 실시간성, 안전성 및 고성능 달성을 목표로 설계 되어 금융거래, 항공 교통 관제, 스마트 그리드 관리, 산업 자동화 시스템 등의 미션크리티컬(Mission Critical) 분야에서 많이 적용되고 있다[5].

2.3 출판-구독 모델에서의 메시지 필터링

DDS 표준에서 정의된 출판-구독 모델은 브로커를 가지지 않으므로, 메시지 필터링이나 계층적 선택(hierarchical selection) 기법의 적용이 매우 어렵다. DDS 표준에서 지원하는 필터링 관련 기법으로는 Content-Filtered-Topic이 있다. 이는 구독자가 특정 토픽에 대한 출판자의 출판 메시지를 모두 수신하는 것이 아니라, 특정 토픽에 대해 구독자가 설정한 추가 컨텐츠 subscription 조건을 만족하는 메시지만을 전달받는 방식이다. 필터링은 출판자, 구독자에서 모두 가능하지만, 출판자에서 필터링이 진행되는 경우에는 해당 토픽에 연결되어 있는 모든 구독자의 subscription의 조건을 충족시키지 못하는 메시지만을 필터링하고, 연결된 구독자 data-reader의 단 하나의 subscription을 만족시키는 subscription이 존재하면 해당 메시지를 모든 data-reader 즉 구독자에게 전송한다[6]. 예를 들어, 토픽 'A'에 연결된 구독자가 a1, a2, a3로 존재하고, 이들에 해당하는 subscription이 각각 s1,s2,s3라고 하자. 이때, 해당 토픽 'A'로 출판자가 메시지 'M'을 보냈을 때, M이 s1,s2,s3 중 오직 s1 조건에만 부합하더라도, 출판자는 모든 구독자(a1,a2, a3)에게 보낸다. 따라서 구독자 a1, a2는 subscription 조건에 부합하지 않는 메시지를 구독자 단에서 필터링 해야 한다. 따라서 대부분의 경우 컨텐츠 필터링은 구독자 단에서 이뤄지게 되며, 이는 구독자가 불필요한 메시지를 지속적으로 받게 되어 전체 네트워크 대역폭이 비효율적으로 사용되는 결과를 낳는다.

이에 본 연구에서는 P2P구조의 출판-구독 모델에서, 구독자들의 subscription을 기준으로 종속 관계에 따른 트리 계층 구조를 이룬 뒤 메시지를 교환하는 방식에 기반을 둔 구독자 단의 메시지 필터링을 제안한다. 구독자들의 구독은 계층구조를 이루고 있기 때문에 출판자는 계층의 상위 구독자에게 전송하도록 하여 구독자에 대한 정보 유지 오버헤드가 줄어들게 되는 동시에, 데이터 전송에 대한 과정을 분산시킴으로써 속도 측면에서의 향상을 이룰 수 있다. 또한, 구독자는 계층에서 상위 구독자에서 n차 필터링 된 메시지를 받음으로써 출판자로부터 직접 메시지를 받을 경우보다 필터링할 메시지가 줄어들게 되어 부담이 줄어들게 된다.

P2P구조의 출판-구독 모델에서는 peer들의 생존 여부(liveness)를 통해 전체 망 구조를 지속적으로 관리하며, 이를 위해 피어 탐색 메시지 (peer discovery message)[7]가 주기적으로 전송된다. P2P구조 출판 구독 모델인 DDS에서 사용하는 피어 탐색(peer discovery) 방식인 SDP(Simple Discovery Protocol) 방식에서는 메시지를 같은 도메인 내에 존재하는 모든 피어들에게 보내는 multicast 방식으로 이루어져 있어, 위 메시지를 수신할 필요가 없는 피어들에게도 메시지가 전송되고, 관리할 필요가 없는 피어들의 정보까지 저장하고 있게 된다. 따라서 피어의 수가 증가할수록, 네트워크, 메모리 등 자원의 낭비가 발생하여[7] 출판-구독 모델의 장점인 확장성에 제한점이 발생하게 된다. 이 문제의 해결을 위해 DDS에서 muliticast 방식이 아닌, 해당 탐색 메시지의 수신이 필요한 피어에게만 보내는 CFDP(Content-Filtered Discovery Protocol) 방식[7]이 제안되었다. CFDP에서는 multicast의 사용이 불가하지만 불필요한 메시지의 송신 및 수신을 줄임으로써, 네트워크, 메모리, 저장 공간 등 컴퓨팅 자원의 불필요한 소모 문제를 해결함으로써 확장성의 이슈를 해결하고 성능을 향상시킬 수 있었다.

CUPUS("CloUd-based PUblish/Subscribe")에서는 중앙집중형 구조의 컨텐츠 및 토픽 기반의 출판-구독 메시징 시스템을 제안하였으며, 본 연구의 제안과 유사한 계층적 subscription 관리를 적용하였다. CUPUS에서는 브로커를 크게 클라우드 브로커 및 모바일 브로커로 구분 지은 뒤, subscription간 관계를 계층 구조로 형성하여(subscription-forest) 클라우드 브로커가 출판자가 출판하는 메시지에 맞는 구독자를 매칭해 준다. 계층적 구조를 통해서 많은 수의 subscription에 대해 매칭하는 구독자를 찾아 메시지를 전달하는 것이 용이해 진다.

본 논문에서 제안하는 방안에서는 출판자가 출판하는 메시지를 구독자들 간에 계층적 구조를 형성하는 것을 통해, 출판자가 전송한 메시지를 구독자가 필터링 할 메시지 수를 줄이는 방식을 제안하고 있다.

3. P2P 기반의 출판-구독 메시징 시스템에서 계층적 Subscription 및 메시지 전송 기법

본 논문에서는 P2P 기반의 출판-구독 모델을 기반으로 하는 메시징 시스템에서 계층적으로 subscription을 구성하여 효과적인 메시지 전송을 하는 기법을 제안한다. 제안방법은 attribute 기반 출판 구독 방식과[8] 같이 간단한 토픽 기반 방식과 표현적 성질을 가지는 컨텐츠 기반 방식의 중간적인 성질을 보인다. 구독자들의 subscription의 종속관계에 따라 구독자들 사이의 부모-자식 관계를 가지는 트리 구조를 형성한 뒤, 부모에서 자식으로 메시지를 전송한다.

3.1 계층 구조 형성

구독자 subscription간의 종속관계에 따른 트리 구조를 형성하기 위해, 구독자들의 메타데이터(ip주소, port번호 subscription)를 유지하는 메타 서버를 둔다. 구독자가 추가 될 경우, 구독자는 서버에 자신의 메타 데이터를 서버로 보내며, 이를 받은 서버는 서버 내 유지하고 있는 트리에 깊이 우선 탐색(Depth First Search) 알고리즘을 통해 구독자로부터 받은 메타데이터가 종속관계를 유지하는 노드가 되도록 트리를 재구성한다. Fig. 1A는 서버에서 구독자에게 트리 구조 내 해당 구독자와 자식의 관계로 연결되어 있는 구독자들에 대한 위치 정보를 보내주는 과정을 나타내고 있다. Fig. 1B는 출판자가 서버에 연결하여 모든 구독자에 대한 정보가 아닌 서버 내 형성하고 있는 트리의 루트 노드에 속하는 구독자의 위치 정보를 얻는 것을 나타내고 있다.

예를 들어, 구독자 A,B에 대해, 구독자 A의 subscription attribute들의 집합을 attr(A)라 할 때, 'attr(A)⊆attr(B)' 조건을 만족하는 경우에만 A-B관계는 부모-자식 관계를 가진다. 새로운 구독자가 참여하는 경우, 먼저 깊이 우선 트리 탐색 알고리즘을 적용하여 부모-자식 관계를 형성하는 노드를 찾아 연결한고, 만약 이러한 관계를 만족하는 노드가 트리에 존재하지 않을 경우에는 새로운 트리의 루트 노드로 두게 된다. Fig. 1C에서 이를 나타내고 있다.

또한 트리 구조 내 노드들의 자식 수의 제한을 둠으로써, 트리 높이에 제한을 두었다. 본 제안구조가 subscription 종속관계에 따른 계층 구조이므로 특정 구독자의 필터링 조건에 부합하지 않는 메시지는, 해당 구독자의 모든 자손 구독자의 필터링 조건에도 부합하지 않음을 의미한다.

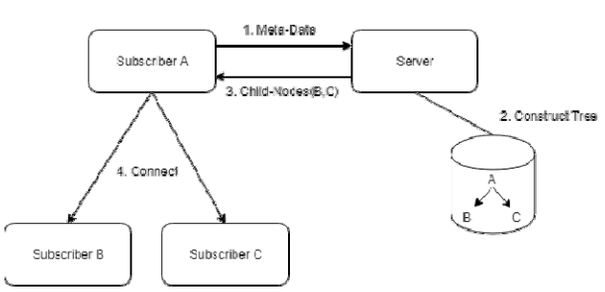

Fig. 1A. Constructing Hierarchical Relationship Between Subscribers

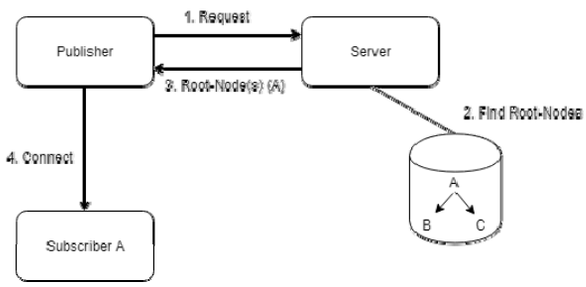

Fig. 1B. Connection Between Publisher and Root Subscriber

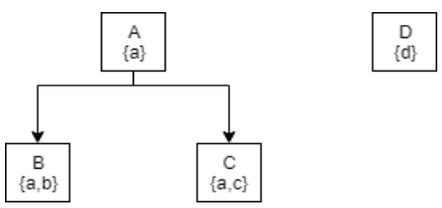

Fig. 1C. Hierarchical Structure Based On Subscription

예로서 attr(A) = {a}, attr(B) = {a,b}, attr(C)={a,c}, attr(D) = {d}

위 그림에서 attr(A)⊆attr(B), attr(A)⊆attr(C) 이어서, 구독자 A는 구독자 B, C 의 부모가 된다. attr(D)는 종속 관계를 만족시키는 노드가 존재하지 않으므로 새로운 서버는 구독자D를 루트로 하는 새로운 트리를 형성한다.

3.2 피어 간 메시지 전송 과정

제안하는 기법에서 메시지 필터링은 구독자단의 필터링을 목표로 하였다. 부모로부터 메시지를 전달 받은 구독자는, 자신의 subscription을 기준으로 메시지 필터링을 진행한다. 필터링의 과정을 거친 뒤에 자신의 subscription 기준에 부합하는 메시지만을 자식 관계를 맺고 있는 구독자에게 전달(forwarding)한다. 트리의 root node에 해당하는 구독자의 경우, 출판자가 보내는 모든 메시지를 수신하게 된다.

자식이 받는 메시지 수를 n(child)라 하고, 부모가 받은 메시지 수를 n(parent)이라 하면 자식 구독자는 부모 구독자의 필터링을 거친 메시지만을 수신하게 되므로 항상 n(child)≤n(parent)를 만족하게 된다. K(s1,s2,….sK)명의 구독자와 출판자가 전송할 N개의 메시지가 있을 경우 순수 Subscriber Side Filtering의 경우 전체 송신되는 메시지 수는 K*N이다. 하지만, 위와 같은 계층 구조를 이루고 있을 경우 임의의 i에 대하여 n(si)≤N을 만족한다(단, 1≤i≤K). 따라서 n(s1)+n(s2)+……+n(sK) ≤ N*K을 만족한다. 또한, 높이가 log(K)인 트리 구조를 통해 메시지 전송 부담을 구독자들에게 분산시켰다. 출판자가 K명의 모든 구독자들에게 메시지를 전송하지 않고 트리 구조에서 Root Node에 해당하는 구독자에게만 보냄으로써 출판자의 메시지 송신 부하 및 구독자로의 메시지 전달 시간을 줄일 수 있다. UDP multicast를 사용하는 경우에 출판자의 부담은 네트워크 레벨로 이전되지만,

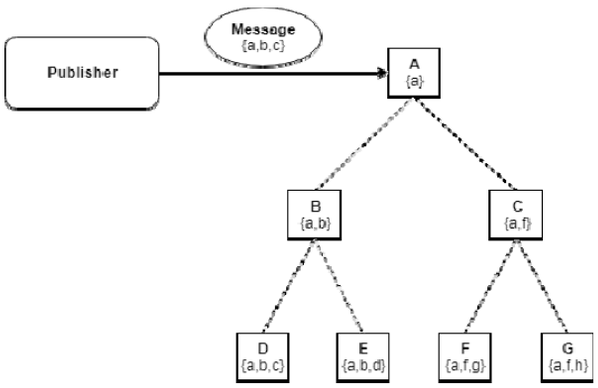

Fig. 2A. Forwarding Messages from Publisher to Root Subscriber

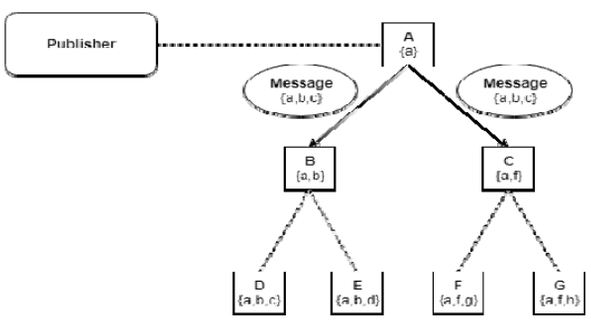

Fig. 2B. Forwarding Messages Between Subscribers

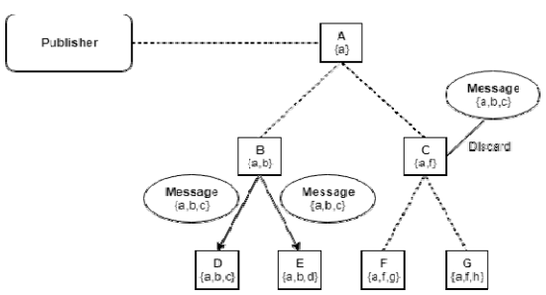

Fig. 2C. Filtering Messages on Subscriber Side

위와 같은 하위 구독자로의 부하 분산은 응용시스템을 설계 구현하는 데에 있어서 효과적인 기법이 될 수 있다. Fig. 2A에서는 attr(Message) = {a,b,c}일 때, 위와 같은 트리구조를 형성하고 있는 경우 A는 트리의 루트에 해당하므로 출판자가 전송한 메시지를 전달받는 것을 나타내고 있다. Fig. 2B에서는 attr(A)⊆attr(Message)를 만족하고, 구독자 단에서 필터링을 하므로 자식의 attr(child)을 고려하지 않고 자식(B, C)에게 전송하는 것을 나타내고 있다. Fig. 2C에서는 A로부터 메시지를 전송받은 구독자 B의 경우도 attr(B)⊆attr (Message)를 만족하므로 자식 (D,E) 들에게 메시지를 보낸다. 하지만, A로부터 메시지를 받은 C의 경우, 구독자 단 필터링을 통해 자신의 subscription에 부합하지 않은 메시지임을 판별하여, 자식 (F,G)에게 메시지를 보내지 않는다. 따라서 위와 같은 경우 모든 구독자가 받은 메시지 수의 총합은 5가 된다. 하지만, 위와 같은 구조를 가지지 않는다면, attr(A,B,D)⊆attr (Message)을 만족하므로, 출판자는 연결되어 있는 모든 구독자에게 메시지를 보내게 된다. 따라서 구독자들이 받은 메시지 수의 총합은 7이 된다.

4. 실험 결과 및 분석

제안 기법의 효과성을 입증하기 위해서 2가지 실험을 진행했다. 트리구조에서 차수(Degree), 깊이(Depth), 일치율(Matching Rate)에 따른 메시지 수 감소율을 matlab을 이용해 시뮬레이션 했고, 본 연구에서 제안하고 있는 구조에서 실제 메시지를 크기별로 보내면서 메시지 전송 시간 측정했다. 이때, 일치율이 K%라는 것은, 어떤 구독자가 받은 메시지 수가 100개일 때, 자식 구독자에게 보내는 메시지의 수가 K개라는 것을 의미한다. 메시지 수 감소율은 트리구조일 때, 트리구조가 아닐 경우 전송되는 메시지 수에서 줄어든 비율을 나타낸다.

Fig. 3에서는 트리 구조에서 부모와 자식의 subscription 일치율이 80%일 때, 차수와 높이에 따른 메시지 수 감소율을 보여주고 있다. 차수가 클수록 깊이가 증가함에 따라 늘어나는 subscriber의 수가 커지므로 필터된 메시지 수가 많아지게 된다. 따라서 메시지 수 감소율은 차수와 깊이에 비례하여 증가하고 이는 전체 구독자 수가 많아질수록 낭비되는 메시지 수의 비율이 감소하는 것을 보여준다.

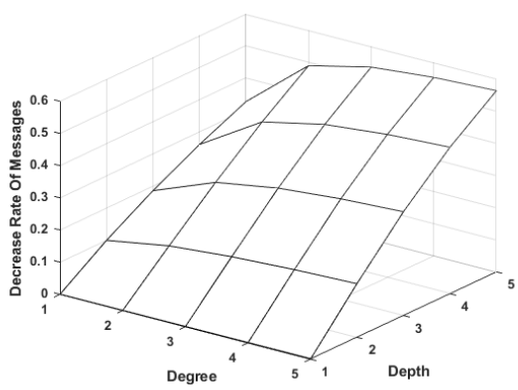

Fig. 3. Decrease Rate of Messages Compared to Degree and Height

Fig. 4에서는 차수가 2인 트리에서, 일치율이 각각 40%, 60%, 80%일 때 메시지 수 감소율을 나타내고 있다. Fig. 3을 통해 트리의 모든 차수에서 Fig. 4와 비슷한 양상을 보이는 결과를 나타낼 것임을 알 수 있다. Fig. 4에서 일치율이 낮은 경우 항상 높은 경우보다 메시지 수 감소율이 높음을 보여주고 있다. 즉 포함 관계를 형성하고 있는 부모와 자식의 subscription의 일치율이 작을수록 본 연구가 제시하는 트리 구조의 효율은 증가하게 되는 것을 보여준다. 또한, 높이가 커질수록 메시지 수 감소율 증가의 폭이 줄어들고 있는 것을 통해 트리 구조 형성 과정에 높이 제한을 두는 것이 효율적임을 알 수 있다.

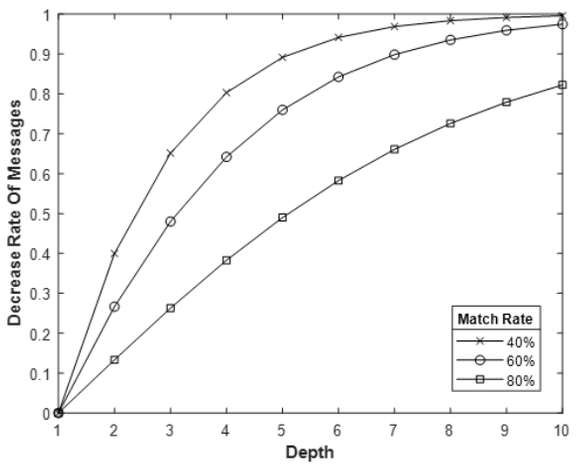

Fig. 4. Decrease Rate of Messages Compared to Match Rate

제안 기법과 기존 토픽 관리 기법을 비교하는 실험을 진행하였다. 실험에서는 7명의 구독자과 구독 정보를 가지는 메타 서버를 준비하였다. 본 실험에서는 트리를 형성하는 시간은 고려하지 않고, 트리가 구성된 이후 정적인 상태에서의 메시지 전송 분산 및 계층화된 필터링 기능으로 인한 성능 변화를 측정하였다.

실험에서는 구독자들의 subscription의 종속관계에 따른 계층 구조에서의 메시지 전송과 계층 구조를 따르지 않고 모든 전송을 출판자가 하는 경우를 비교하기 위해 다음과 같이 3가지 경우를 비교했다.

1) 트리 구조를 형성하지 않고 출판자가 모든 구독자들에게 메시지를 보내는 경우(Fig. 5A)

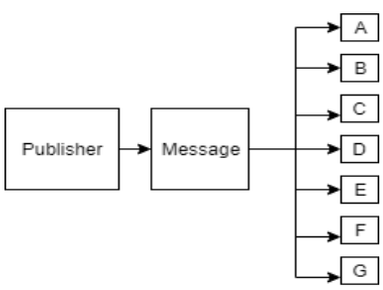

Fig. 5A. Non Hierarchical Relationship Between Subscribers

2) 트리 구조를 형성 하지만 필터링이 적용되지 않은 경우 (Fig. 5B)

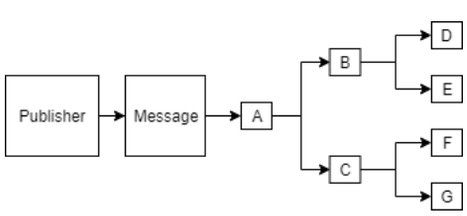

Fig. 5B. Hierarchical Relationship Between Subscribers Without Considering Subscription

3) 트리 구조와 필터링이 적용되어 특정 메시지가 필터링 되는 경우 (Fig. 5C)

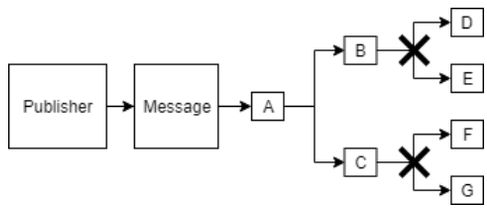

Fig. 5C. Hierarchical Relationship Between Subscribers Considering Subscription

Table 1. Test Environment

OS	Ubuntu 18.04
CPU	Intel® Core(TM) i5-2500 3.30GHz
RAM	8.00GB

Table 1에서 나타내는 환경에서 실험을 진행했다. 출판자가 15개의 메시지를 보낸 뒤에 출판자가 첫 메시지를 보낸 시간으로부터 7명의 구독자 중 15개의 메시지가 가장 늦게 도착한 구독자의 시간의 차이를 End-to-end delay으로 정의하고, 구조들 간의 그리고 메시지 크기 별로 비교한다. 결과 값의 신뢰도를 위해 각 실험값은 10번의 실험의 평균값으로 했다.

트리 구조를 형성하지 않은 경우, 출판자는 모든 구독자에게 메시지를 보내고, 트리는 완전 이진 트리 구조이다. P2P구조의 출판-구독 모델에서는 UDP[9] unicast, UDP multicast 방식, 그리고 TCP 방식 모두 사용 가능하지만, 많은 경우 UDP를 이용한 멀티캐스트 방식을 통해 구독자에게 전달한다. 데이터 손실이 일어날 수 있는 UDP에 Reliability을 제공하기 위해 DDS의 경우 QoS(Quality Of Service)등을 추가적으로 적용시켜 Reliable UDP 방식을 제공한다. 본 실험에서는, QoS 등을 추가적으로 정의하지 않아서 reliable UDP 방식 대신에 TCP 방식으로 피어간 통신을 구성하였다.

Fig. 6과 Table 2의 실험 결과에서는 메시지의 크기가 1mb, 5mb, 10mb일 때 모두 트리 구조를 형성하여, 메시지 전송을 구독자에게 분산시킨 결과를 보여준다. 트리 구조를 형성 하지 않은 채 출판자가 모두 전송하는 경우보다, 메시지 크기가 1mb인 경우 약 27.1%, 5mb의 경우 약 22.3%, 10mb의 경우 약 18.7%의 향상된 속도를 보였다. 또한, subscription이 포함되어 본 연구에서 제안하는 방법인 트리 구조에서는, 구독자 B, C가 자신의 subscription을 통해 메시지를 필터링하여 자식 구독자에게 메시지를 보내는 과정을 생략할 수 있어 subscribe D, E, F, G 는 불필요한 메시지를 받지 않았다. 따라서 기존의 7명의 구독자 중 3명의 구독자만 메시지를 받았고 이를 필터링이 적용되지 않은 트리 구조에서의 메시지 전송 방식과 비교한 결과 1mb의 경우 약33.3%, 5mb의 경우 42.9%, 10mb의 경우 48.8%의 향상된 속도를 보였다. 위 실험을 통해, subscription이 세분화 될수록 부모 구독자에 의해 필터링 되는 메시지의 수가 많아지게 되어 자식 구독자가 받는 불필요한 메시지의 수가 줄어들 가능성이 커지게 되어 성능 향상의 폭이 커짐을

알 수 있다. 또한, 수신하지 않은 총 불필요한 메시지의 크기가 증가할수록 성능 향상의 폭이 커지는 것을 위 메시지 크기에 따른 실험값 비교를 통해 알 수 있다. 즉, 메시지의 크기가 클수록 성능 향상이 폭이 증가한다. 또한, 출판자가 트리 구조의 루트 구독자에 대한 정보만을 유지하게 되어, 오버헤드를 줄일 수 있음을 알 수 있다.

Fig. 6. Message Delivered Time Compared to Message Size and Structure

Table 2. Message Delivered Time Compared to Message Size and Structure

FILE SIZE	1mb	5mb	10mb
WITHOUT TREE	2.636822	13.718241	30.663228
TREE WITHOUT FILTERING	1.921007	10.64703	24.923427
TREE WITH FILTERING	1.279427	6.075668	'12.750575

5. 결론 및 향후 연구

P2P기반의 출판-구독 모델은 비 분산 단일 브로커 구조가 가지던 단일 실패점과 병목 현상에 대한 문제와 분산 복수 브로커 구조가 가지는 구독자들의 subscription 데이터 일관성 유지 문제에 대하여 유연함을 가지고 있어 브로커 구조와 비교하여 확장성이 뛰어나다. 하지만, 메시지 필터링 및 전송의 과정의 경우 기존의 브로커가 담당하던 부분을 피어들이 담당하게 되었으며, 대부분의 경우에 메시지 필터링의 과정은 구독자단에서 이루어진다. 따라서 유통되는 메시지의 수가 증가하고 메시지의 크기가 커질수록 구독자는 불필요한 메시지를 필터링 해야 하는 오버헤드가 증가하게 되었다.

본 논문에서 제안한 방식에서는 P2P 기반 발행 구독 모델에서 구독자 피어들 간의 관계를 subscription을 기준으로 종속관계에 따라 계층적 구조인 트리 구조를 형성함으로써 부모가 존재하는 구독자들은 부모로부터 필터링된 메시지를 전송 받게 되어 메시지 수가 증가하더라도 상대적으로 적은 수의 불필요한 메시지를 수신하게 된다. 또한, 구독자들 사이의 트리 구조를 형성하여 메시지 전송 과정을 출판자에서 구독자로 분산시킴으로써 속도 측면에서의 성능 향상을 얻을 수 있었다. 단, 본 논문에서 제안하고 있는 구조는 트리구조를 형성하고 있으므로 상위 구독자가 네트워크 상에서 연결이 끊어진다면 해당 구독자를 루트로 하는 서브트리에 속한 모든 구독자들은 메시지를 받을 수 없는 상황이 발생할 수 있다. 이에 대비하여 P2P네트워크 상에서 논리적으로 가까운 구독자간 heartbeat를 발생시켜 생존여부를 확인하고, 생존이 확인되지 않는 구독자에 대한 정보를 메타 정보를 유지하고 있는 서버에 알린다면 fault-tolerant한 시스템을 구축할 수 있을 것이다. 추가하여, 메타 정보를 유지하고 있는 서버에서 트리 구조 유지를 위해, 특정 구독자가 끊겼을 때 해당 구독자를 대신할 designator를 유지하고 있다면 보다 효율적인 시스템이 될 것이라 예상할 수 있다. 단, Heartbeat을 확인하는 구독자가 동시에 끊어지는 상황은 확률이 매우 희박하기 때문에 고려하지 않는다. 실제로 Hadoop 파일 시스템[10]에서는 heartbeat을 활용하여 노드 간 생존여부를 확인한다. 이를 통해 본 과정이 구독자의 연결이 끊어지는 상황에 대비한 fault-tolerant한 시스템을 구축하기에 적합한 것을 알 수 있다.

실험 결과를 통해 메시지의 크기가 커질수록, 성능 향상도가 커짐을 확인하였고, 제안 방법이 메시지의 크기가 크고 주기적으로 발생하는 환경에서 효과적임을 알 수 있다.

향후 연구로서 메시지 병합을 통한 성능 향상 기법의 추가가 고려된다. Kafka에서 메시지 전송에서 성능을 높이기 위해 여러 개의 메시지를 단일 메시지로 병합 후 일괄 전송하여 TCP/IP 과정에서 생기는 roundtrip을 줄이는 방식을 이용하고 있다. 본 구조에서 여러 개의 메시지를 하나의 메시지로 병합하여 보낸다면 roundtrip을 줄임으로써 생기는 성능 향상과 동시에 메시지의 크기가 커짐에 따른 성능 향상을 함께 기대할 수 있을 것이라 예상된다.

또한 제안 방식에서 subscription사이의 종속 관계를 파악하고 계층 구조를 형성하기 위해, 구독자들의 메타데이터를 관리하는 서버를 두었다. 이와 같은 서버 의존성(단일 부모로 인한 단일 실패점 및 병목 현상 발생 가능성)을 해결하기 위해, subscription간의 계층구조를 서버의 도움 없이 피어들 간 subscription 정보 교환으로 동적으로 형성 할 수 있다면 구독자들의 출입에 보다 유연해짐으로써 확장성 있는 P2P구조의 출판-구독 모델이 될 것이다.

References

[1] A. Antonić, M. Marjanović, K. Pripužić, and I. P. Žarko, "A mobile crowd sensing ecosystem enabled by CUPUS: Cloud-based publish/subscribe middleware for the Internet of Things," *2014 International Conference on Future Internet of Things and Cloud*, Dec. 2014.

[2] P. T. Eugster, P. A. Felber, R. Guerraou, and A. M. Kermarrec, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys*, Vol.35, No.2, pp.114-131, Jun. 2003.

[3] Z. Wang, W. Dai, F. Wang, H. Deng, S. Wei, X. Zhang, and B. Liang, "Kafka and its Using in High-throughput and Reliable Message Distribution," Nov. 2015.

[4] J. Kreps, N. Narkhede, and J. Rao, “Kafka: a Distributed Messaging System for Log Processing,” *Proceedings of 6th International Workshop on Networking Meets Databases (NetDB)*, Athens, Greece. 2011.

[5] Pardo-Castellote, Gerardo, “Omg data-distribution service: Architectural overview,” *23rd International Conference on Distributed Computing Systems Workshops,* 2003. Proceedings, IEEE, 2003.

[6] RTI. “Where Filtering is Applied—Publishing vs. Subscribing Side” [Online], https://community.rti.com/static/documentation/connext-dds/5.2.0/doc/manuals/connext_dds/html_files/RTI_ConnextDDS_CoreLibraries_UsersManual/Content/UsersManual/Where_Filtering_is_Applied.htm.

[7] K. An, A. Gokhale, and D. Schmidt, “Content-based Filtering Discovery Protocol (CFDP): Scalable and Efficient OMG DDS Discovery Protocol,” *Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops*, May 2014.

[8] M. Li, F. Ye, M. Kim, H. Chen, and H. Lei, “A Scalable and Elastic Publish/Subscribe Service,” *2011 IEEE International Parallel & Distributed Processing Symposium*, May 2011.

[9] W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Bunchmann, “A Peer-to-peer Approach to Content-Based Publish/Subscribe,” *Proceedings of the 2nd International Workshop on Distributed Event-based Systems*, Jun. 2003.

[10] Robert Chansler, Hairong Kuang, Sanjay Radia, Konstantin Shvachko, and Suresh Srinivas, “The Hadoop Distributed File System,” In *Proceedings of IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 2010.

정 진 선

https://orcid.org/0000-0002-8983-5869
e-mail : js9609@ajou.ac.kr
2015년～현 재 아주대학교
소프트웨어학과 학사과정
관심분야 : 병렬 및 분산 컴퓨팅

오 상 윤

https://orcid.org/0000-0001-5854-149X
e-mail : syoh@ajou.ac.kr
2006년 미국 인디애나대학교 컴퓨터공학과 (박사)
2006년～2007년 SK텔레콤 전략기술부문
2007년～현 재 아주대학교
소프트웨어학과 교수
관심분야 : 분산/병렬 시스템, 고성능컴퓨팅, 클라우드컴퓨팅, Semantic Web