Edge Caching for Enriched Notifications Delivery in Big Active Data

ICDCS 2018, CCF-B
Md Yusuf Sarwar Uddin, Nalini Venkatasubramanian
UCI 加州大学欧文（尔湾）分校

题目说明

Big Active Data System，由许多计算和存储节点组成的集群；
BAD 是一种数据管理模式,可以接收大量的来自异构数据源的数据,如传感器数据、社交网络、web 数据等，并使大量的终端用户通过声明订阅来订阅这些数据项。
Notification 指的是在发布-订阅（publish-subscribe）模型中，用户订阅了感兴趣的内容后，数据集群中有该类内容时，系统对此用户的通知。
Enriched Notification
BAD 能够跨多个发布的内容与订阅相匹配，因此有多种不同的内容来 enrich 通知。

背景

传统的发布-订阅系统将订阅与单个发布流匹配，以生成通知
在传统的发布-订阅系统中，消费者(用户)通过订阅表达对信息的兴趣;发布(可能来自其他用户)与订阅匹配，包含信息的通知发送给感兴趣的订阅者。
BAD 可以匹配多个发布之间的订阅(通过利用后端存储)，因此可以用一组丰富的不同内容丰富通知。

BAD 与 pub-sub 架构、active database 的不同:
传统的 pub-sub 架构一般无 broker，而 BAD 开发了基于 broker 的体系结构，以便将订阅的存储、发布的处理和路由分散到订阅的用户；
主动数据库主要是针对数量有限的触发器或查询，而且只针对有限的查询模式，而 BAD 平台旨在及时向大量用户交付感兴趣的数据。

BDMS 后端负责发布(内容)的存储和处理, BAD 代理网络负责通知管理和向订阅者分发通知，且这是可伸缩且高效的。

引入 Broker 的好处:
1. 订阅者间的订阅可共享，从 broker 中获取缓存结果减少延迟；
2. 允许订阅者在检索频道(channel)结果时可以有一定程度的异步性(当 chanel 结果准备交付时订阅者可以保持脱机状态，当它们联机时订阅者可以接收它们)。

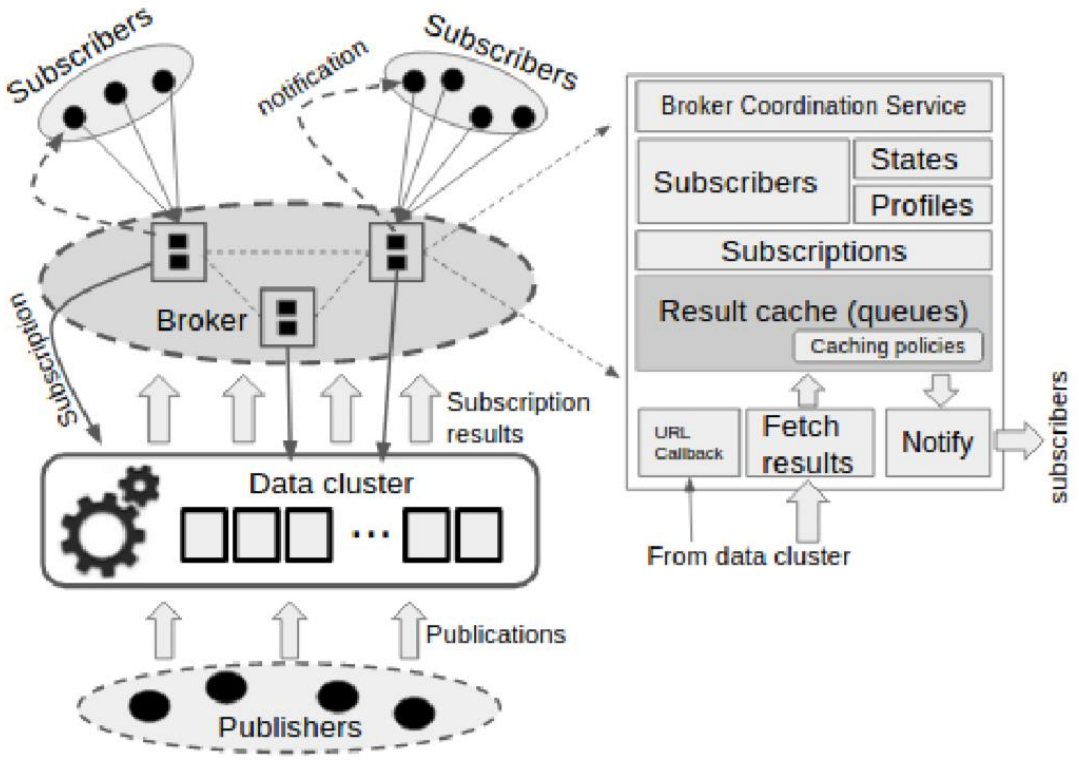


Fig. 1. Components of Big Active Data (BAD) systems.

相关工作

Eviction-based 缓存策略

两种常用的 metrics：recency、frequency；

如 LRU、LFU，以及它们的一些变体（LRU-size、LRU-threshold、LFU-aging、LFU-weighted）；

TTL(time to live)-based 缓存策略：当缓存的对象过期时，缓存对象将被替换出去

应用场景：如 DNS；

挑战

应用场景

当订阅者订阅了一些应急场景如龙卷风、洪灾、枪击等发生时的通知。当

体系架构

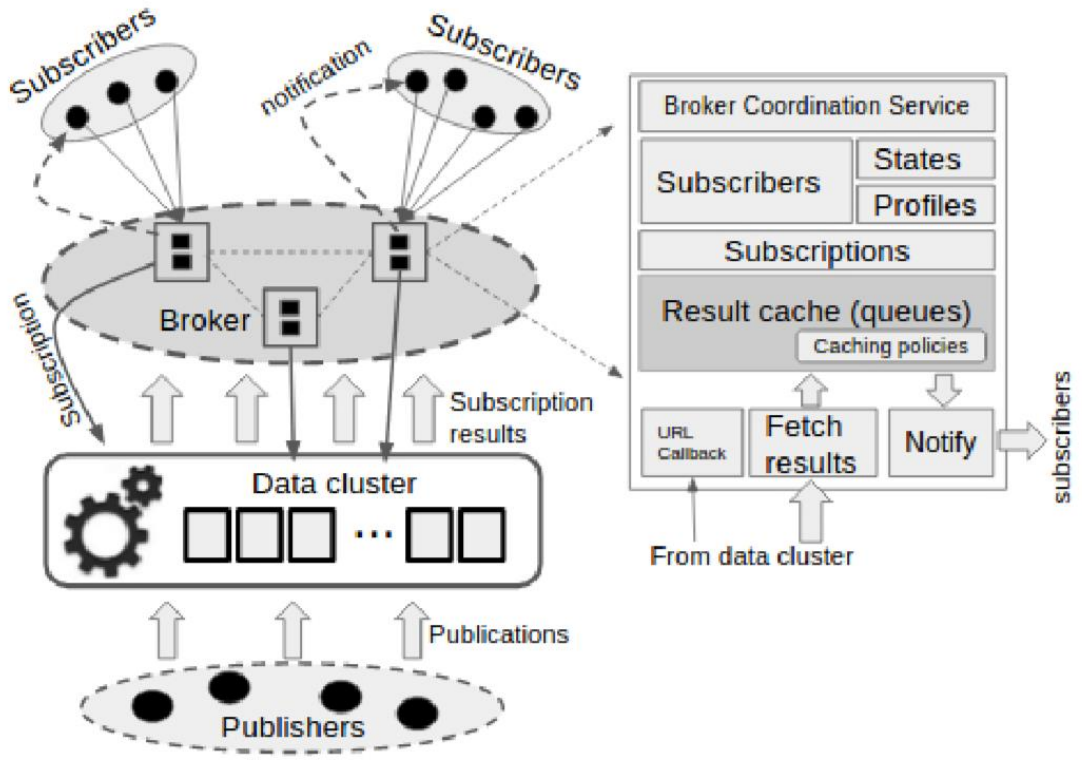
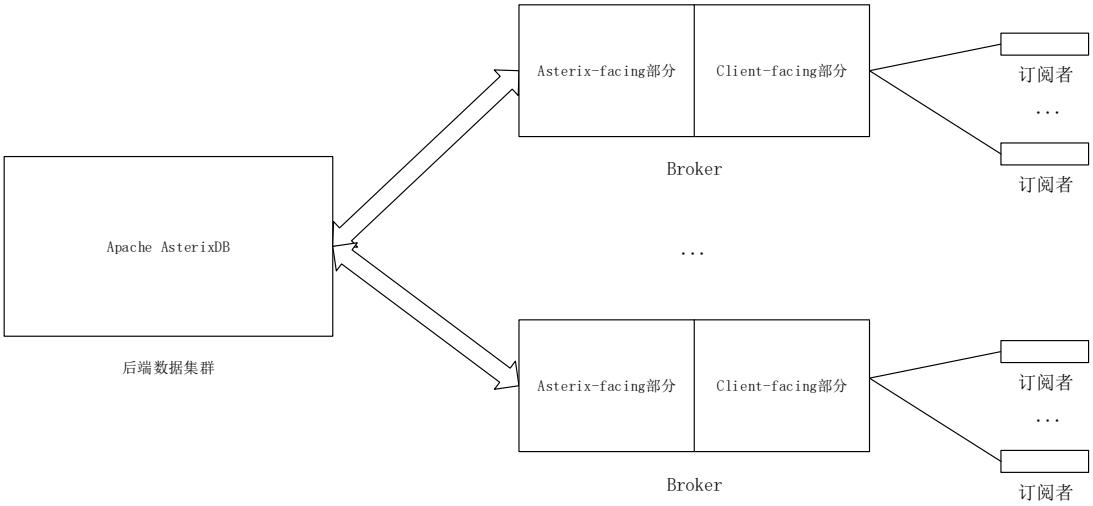


Fig. 1. Components of Big Active Data (BAD) systems.

BAD 组件两大功能：

1. 大数据管理和监视——由 BAD 数据集群处理；
2. 通知管理和分发——由 broker 处理；

BCS 管理 broker 网络：当一个新的 broker 加入到 broker 网络中，需要在 BCS 中注册；



Broker 分为两部分

Asterix-facing 部分负责与后端进行交互，使用 RESTful HTTP

Client-facing 部分负责管理 client，并提供 REST call 给 client 以便其从 broker 中接收服务。

数据集群与 broker 之间的信息交互模型可以分为：推模型和拉模型，本文采用拉模型。当 broker 上传订阅(subscription)到数据集群时，broker 会注册一个回调 URL(callback URL)指向自身(通常称为 WebHook)；当数据集群中有匹配的内容时，数据集群将调用 callback URL 来通知 broker。

BAD 系统执行三个任务：

1. 管理和存储到来的发布(内容)和订阅；data in 阶段
2. 将发布与订阅进行匹配；data processing 阶段
3. 交付结果(匹配好的发布)给订阅者；data out 阶段

本文工作主要是在 data out 阶段。

工作流程

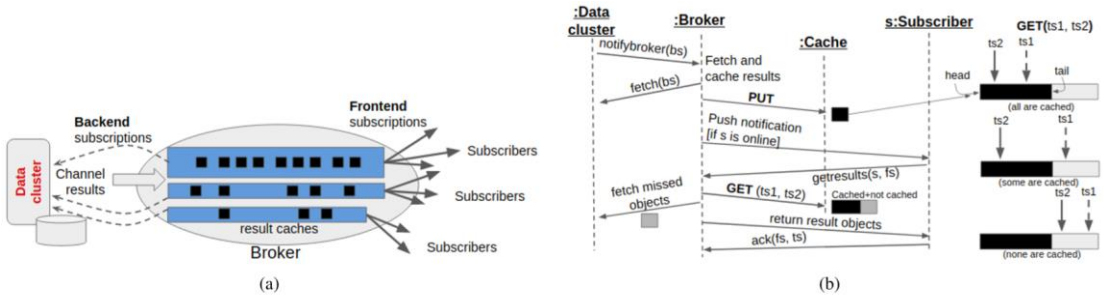


Fig. 2. Sequence diagram of subscriber retrieving channel results.

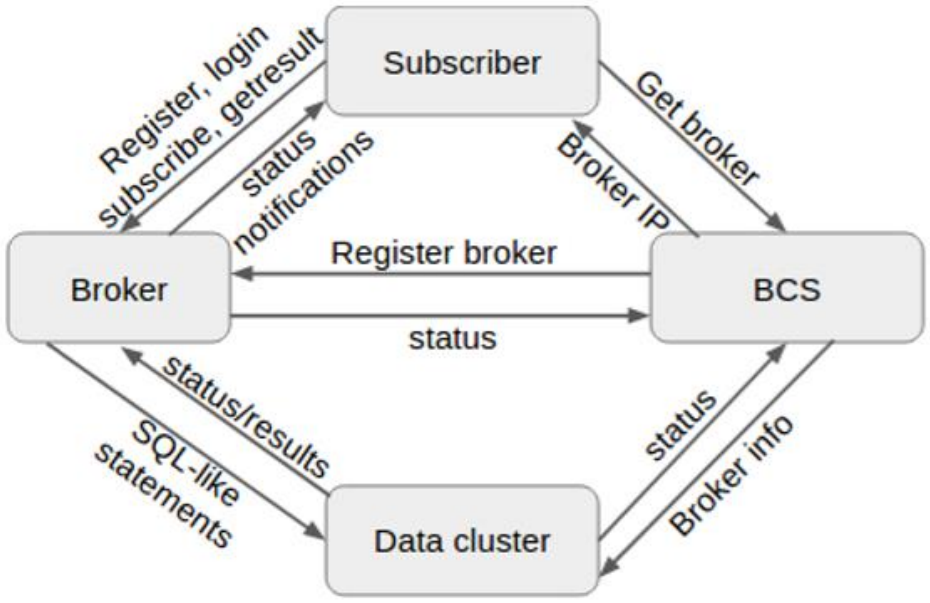


Fig. 6. Interactions among components of Big Active Data (BAD) systems

实验设置

系统原型组成：三个数据节点组成集群，一个 broker 节点
Broker 节点是一个 RESTFul HTTP 服务器，使用 Tornado web 框架，用 Python3 编写。

TABLE III
CHANNELS

Channel (params)	Predicates	P (s)
emrgn (E)	Events of type 'E'	10
emrgn (L)	— near loc 'L'	20
emrgn (E, L)	— of type 'E' near loc 'L'	30
emrgn-ws (E, L)	— of type 'E' near 'L' with shelters	60
emrgn-nm (U)	— near user 'U'	10
emrgn-nm (E, U)	— of type 'E' near user 'U'	10
emrgn-nm (E, U)	— of type 'E' near 'U' with shelters	10

TABLE I
UTILITY-DRIVEN CACHING POLICIES

Utility, $\Delta(i,j,k)$	Caching value, ϕ_{ij}	Dropping criteria	Name
Uniform, 1	$\sum_k z_{ijk} = f_{ij}$	$\min \frac{f_{ij}}{s_{ij}}$	LSCz
Size, s_{ij}	$f_{ij} \times s_{ij}$	$\min f_{ij}$	LSC
Latency, l_{ij}	$f_{ij} \times l_{ij}$	$\min \frac{f_{ij} l_{ij}}{s_{ij}}$	LSD

TABLE II
SIMULATION SETTINGS

Setting	**Value**
No of subscribers	10000
Subscription per subscriber	10
No of unique subscriptions	1000
Subscription duration	Lognormal(1, 2) minutes
Result object size	Uniform(1KB, 500KB)
Allowed cache size	50MB — 500MB
Result object arrival	Poisson, rate 1 per 10–60sec
Subscriber ON duration	Lognormal(1.753, 1.425)
Subscriber OFF duration	Lognormal(2.415, 1.11)
Broker to data cluster bandwidth	10MB/s
Broker to subscriber bandwidth	1MB/s
RTT (broker to data cluster)	500ms
RTT (broker to subscribers)	250ms

Scheme	Dropping criteria
LRU	drop from the least recently accessed cache
LSC	drop object with the fewest subscribers
LSCz	LSC normalized by object size
LSD	drop object with the least delay-size ratio
EXP	earliest object to be expired
TTL	drop objects when TTL expires

实验分析

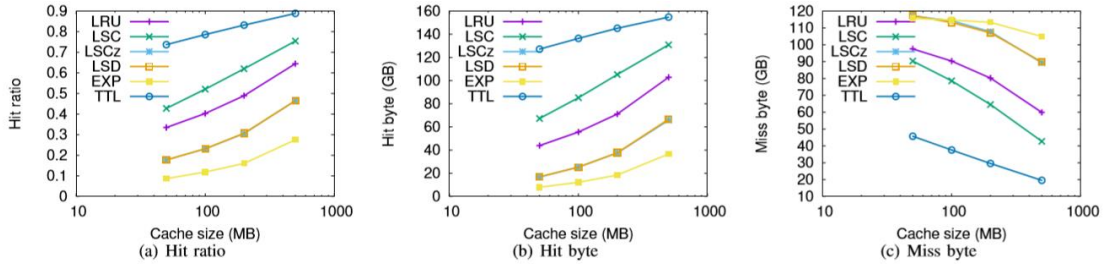


Fig. 3. Hit ratio, hit byte, and miss byte across caching policies.

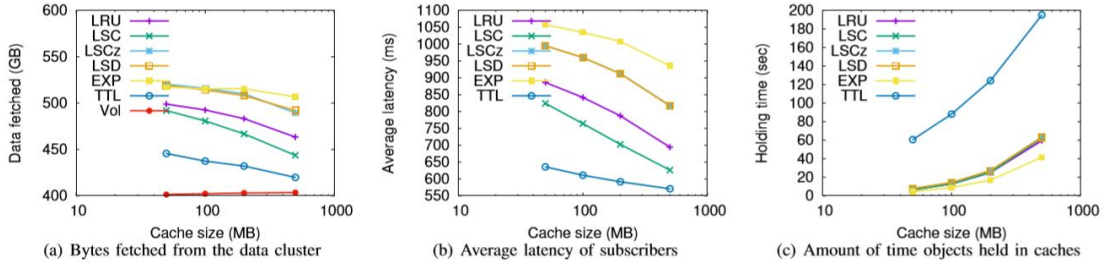


Fig. 4. Average latency of subscribers and the total amount of data fetched from the data cluster

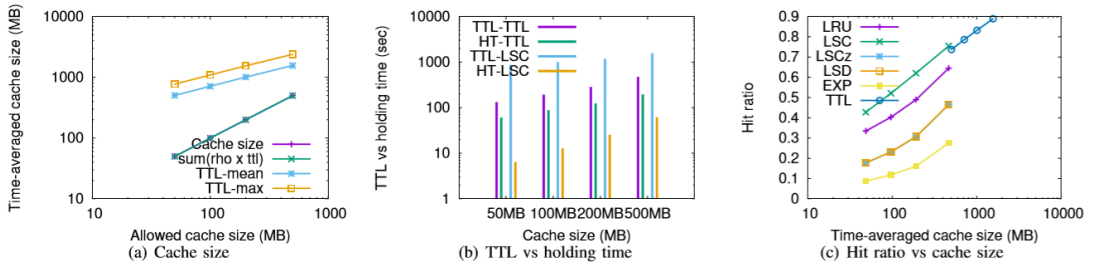


Fig. 5. Comparing eviction-based caching policies with TTL caching.

总结