



Topic based Pub/Sub using a Distributed Database



Group 9
Tanooj Parekh
Sandeep Reddy
Harun Rashid

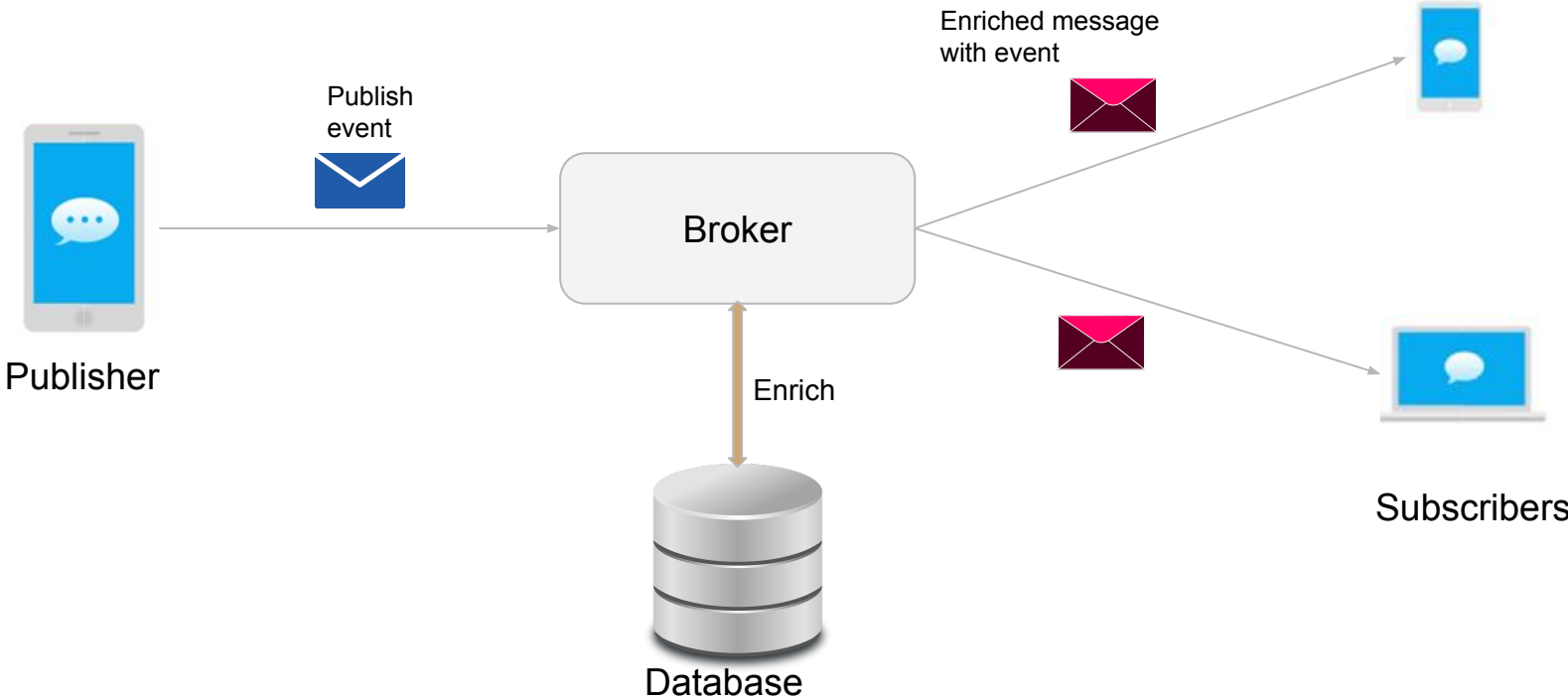
Related Work

- Several topic based Pub/Sub systems have been designed in literature and industry
- Topic based systems ex. Scribe^[2], TIBCO Rendezvous^[3], SpiderCast^[1], PADRES^[4]
- Content based systems ex. Gryphon^[5], Meghdoot^[6]
- Focus is on developing **smarter ways to route messages** and maintain subscriptions
- Routing types explored so far: Tree based, distributed hash tables, probability based etc
- Most of them exploit what the underlying network offers: network overlay^[SCRIBE], network level multicast^[Gryphon], Peer2Peer infrastructure [Meghdoot]

Motivation and Goals

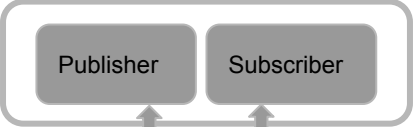
- Not enough research focused on innovative architectures to provide features that extend **beyond simple pub/sub**.
- With big data and better mining techniques, there is scope for a system which is able to **enrich** traditional publish messages with relevant information.
- Our goal is to develop a platform which is **flexible** enough to allow such an architecture, but also leverages the usual advantages of distributed systems such as **scalability** and **fault tolerance**.
- Once such an architecture is created, several applications or systems can be created that are able to leverage these capabilities.

Enriched Topic based Pub/Sub

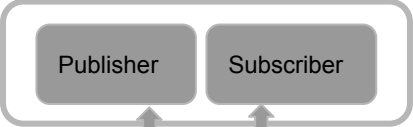


Architecture

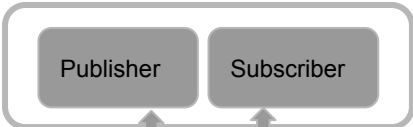
Client



Client



Client



Message Handlers

Publish Subscribe Unsubscribe

Database Access Layer

Message Handlers

Publish Subscribe Unsubscribe

Database Access Layer

Message Handlers

Publish Subscribe Unsubscribe

Database Access Layer

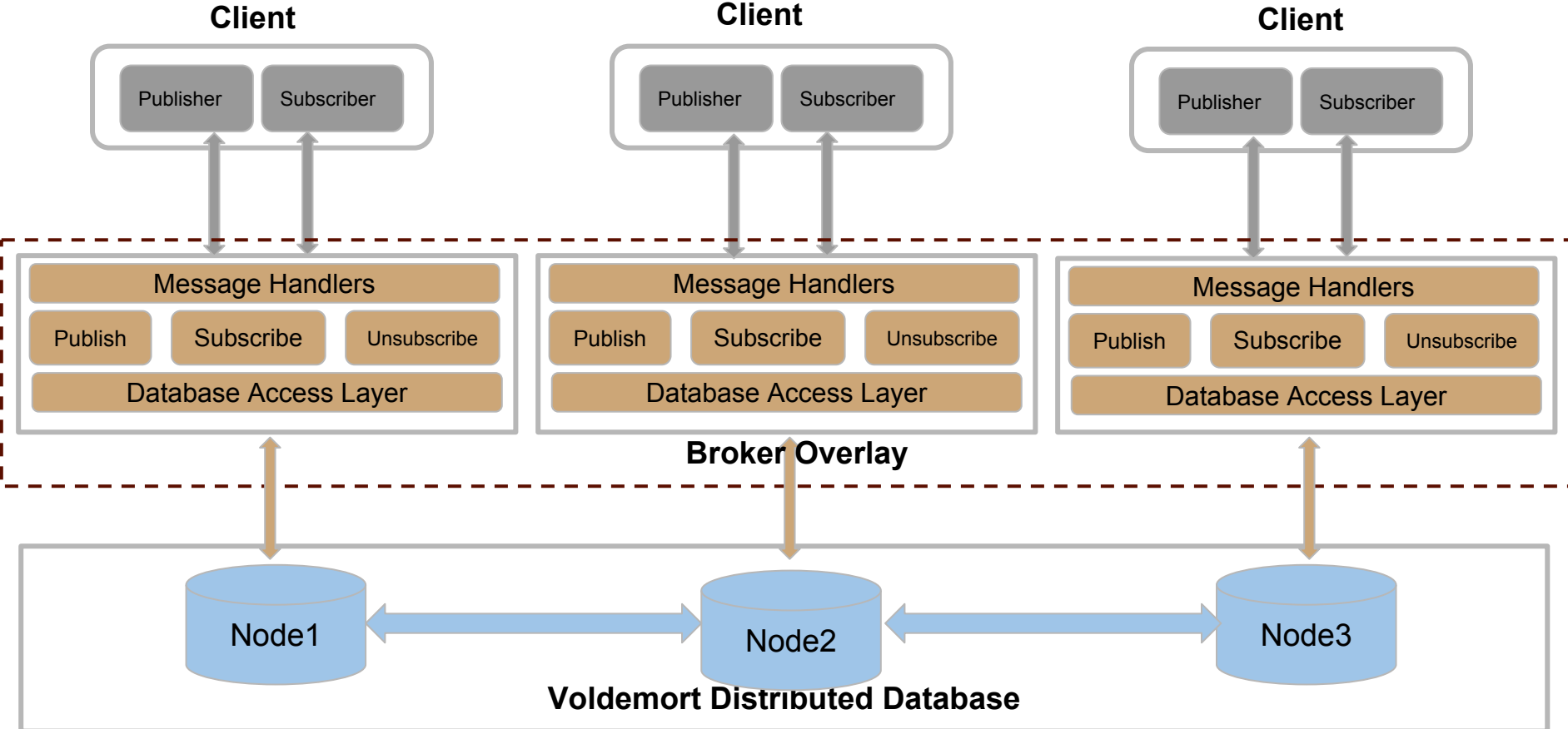
Broker Overlay

Node1

Node2

Node3

Voldemort Distributed Database



An application - Disaster Management

Screencast Video Link

<https://drive.google.com/file/d/0B7ycUNo8ea1HYmNmRFFuM2VKcEk/view>

Future Work

Leverage existing research on creating more **efficient** broker overlays.

Use smarter algorithms to **dynamically enrich** messages with the most relevant information.

Create the broker network and database nodes with **geographical considerations**, and place a load balancer that assigns clients to the nearest broker.

References

- [1] Chockler, G., Melamed, R., Tock, Y., & Vitenberg, R. (2007). SpiderCast. *Proceedings of the 2007 Inaugural International Conference on Distributed Event-based Systems - DEBS '07*. doi:10.1145/1266894.1266899
- [2] Castro, M., Druschel, P., Kermarrec, A., & Rowstron, A. (2002). Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE J. Select. Areas Commun. IEEE Journal on Selected Areas in Communications*, 20(8), 1489-1499. doi:10.1109/jsac.2002.803069
- [3] TIBCO Rendezvous product. Link: <http://www.tibco.com/products/automation/enterprise-messaging/rendezvous>
- [4] Jacobsen, H., Cheung, A., Li, G., Maniyamaran, B., Muthusamy, V., & Kazemzadeh, R. S. (n.d.). The PADRES Publish/Subscribe System. Principles and Applications of Distributed Event-Based Systems. doi:10.4018/9781605666976.ch008
- [5] ArXiv.org cs arXiv:cs/9810019. (n.d.). Retrieved May 31, 2016, from <http://arxiv.org/abs/cs/9810019>
- [6] Gupta, A., Sahin, O. D., Agrawal, D., & Abbadi, A. E. (2004). Meghdoot: Content-Based Publish/Subscribe over P2P Networks. *Middleware 2004 Lecture Notes in Computer Science*, 254-273. doi:10.1007/978-3-540-30229-2_14