Toroni: Lightweight Alternative To Message Brokers For Many-to-Many Interprocess Communication On A Single Node

Rousko Atanasov, Kalin Tsvetkov

vP6

Challenge

- Crafting an efficient protocol suite for advanced interprocess communication specialized for a single node (no networking)
- Needs to be Brokerless
- Multicast, Publish/Subscribe
- Work on POSIX

Key Objectives

1. Broker-less (1)
2. Many-to-Many (2)
3. Total Order (3)
4. Reliable (4)
5. Termination Safe (5)
6. Support pub/sub for topics

Key Features

1. No dedicated server process
2. Many writers and readers. Runtime dynamic sets
3. Readers agree on message order
4. Readers can detect message loss
5. Process crash not harmful to peers

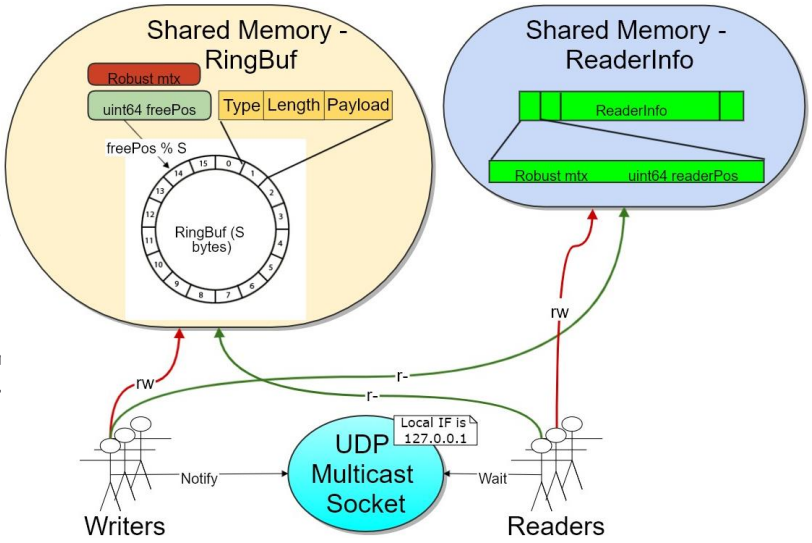
Approach

- Isn't this solved already?
- Toroni is a protocol stack running on top of unreliable multicast (as control plane) and shared memory (as data plane)
- Writers use a robust mutex to append messages to a ring buffer in shared memory. They notify readers through a UDP multicast socket
- Readers consume messages from the ring buffer and report their position in shared memory

	Broker-less	Many-to-Many	Total Order	Reliable	Termination Safety
Message queue	yes	no			
Shm+Disruptor	yes	yes	yes	yes	no
UDP multicast	yes	yes	no	no	yes
UDS	yes	no	yes	yes	yes
PGM	yes	yes	no	yes	yes
SRM			no		yes
0MQ pub/sub	yes	yes	no	no	yes
0MQ Xpub/Xsub	no				yes
0MQ pgm/epgm	yes	yes	no		yes
Aeron	no		no	yes	yes
Toroni	yes	yes	yes	yes	yes

Protocol Stack

- Topic Protocol – Client-based filtering of topics. No membership notion
- Reliable Message Protocol - Robust Futex, Ring Buffer, Message Stream, Backpressure, Notification



Performance

- RMP is 8x faster than TCP
- CPU friendly
- Microseconds latency
- Millions msg/sec throughput

Conclusion

- Unique feature set on single node
- Simpler than a message broker or existing reliable multicast network protocols

Open Source. Contact us for language support

For further info about Poster vP6 contact us at

ratanasov@vmware.com
ktsvetkov@vmware.com