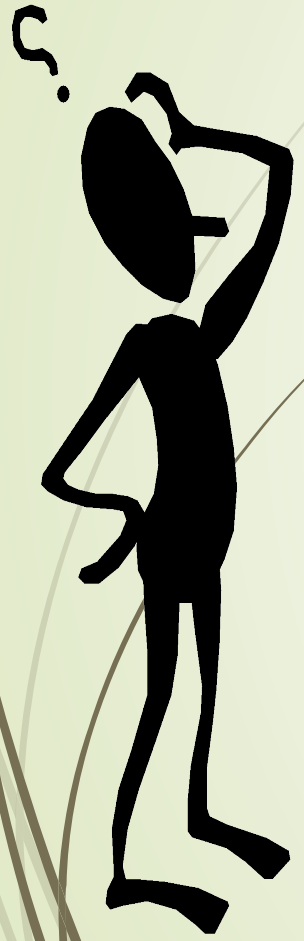
ALFA: Status and plans

Mohammad Al-Turany

ALFA's talks today!



- Transport layer (FairMQ) — Alexey
- Plugins mechanism for integration with external services — Dennis
- Dynamic Deployment System (DDS) — Anar

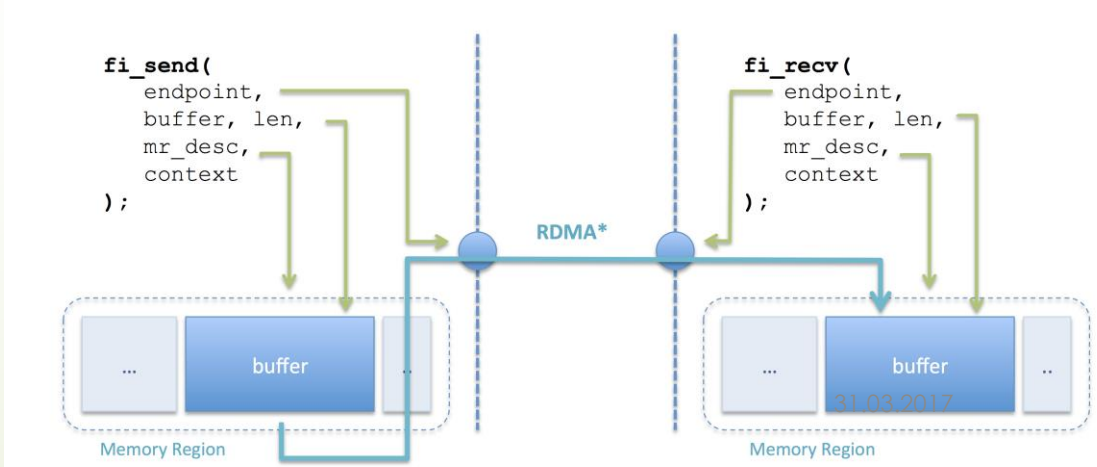
This talk

- RDMA support in ALFA (Plans)
- FairMQ: What could change?


RDMA

Remote Direct Memory Access (RDMA)

- A technology that allows computers in a network to exchange data in main memory without involving the processor, cache, or operating system of either computer
- This permits high-throughput, low-latency networking, which is especially useful in massively parallel computer clusters.



RDMA support: Implementing a new transport protocol in nanomsg.

- Low level bindings to the network fabric using libfabric abstraction.
- The transport translates the POSIX-like API of NanoMsg into an RDMA-like API for libFabric, transparently to the user.

Ioannis Charalampidis

RDMA: Alternative solution

OFA Developer Workshop 2014

Shared Memory Communications over RDMA (SMC-R): Update

Jerry Stevens IBM
sjerry@us.ibm.com

SMC-R for Linux: Overview

- IBM is in the process of developing an SMC-R solution for Linux
- kernel based solution (working towards upstream kernel acceptance)
- No special license requirements (GPL)
- Introduces a new AF_SMC socket family and a preload library to transparently run AF_INET socket applications for SMC-R (no application changes required)

SMC-R Key Attributes - Summary

- ✓ Optimized Network Performance (leveraging RDMA technology)
- ✓ Transparent to (TCP socket based) application software
- ✓ Leverages existing Ethernet infrastructure (RoCE)
- ✓ Preserves existing network security model
- ✓ Resiliency (dynamic failover to redundant hardware)
- ✓ Transparent to Load Balancers
- ✓ Preserves existing IP topology and network administrative and operational model

A Big Networking Update For Linux 4.11

Written by Michael Larabel in Linux Kernel on 22 February 2017 at 06:38 AM EST. 5 Comments

David Miller has mailed out the rather big set of updates to the networking subsystem for the Linux 4.11 kernel.

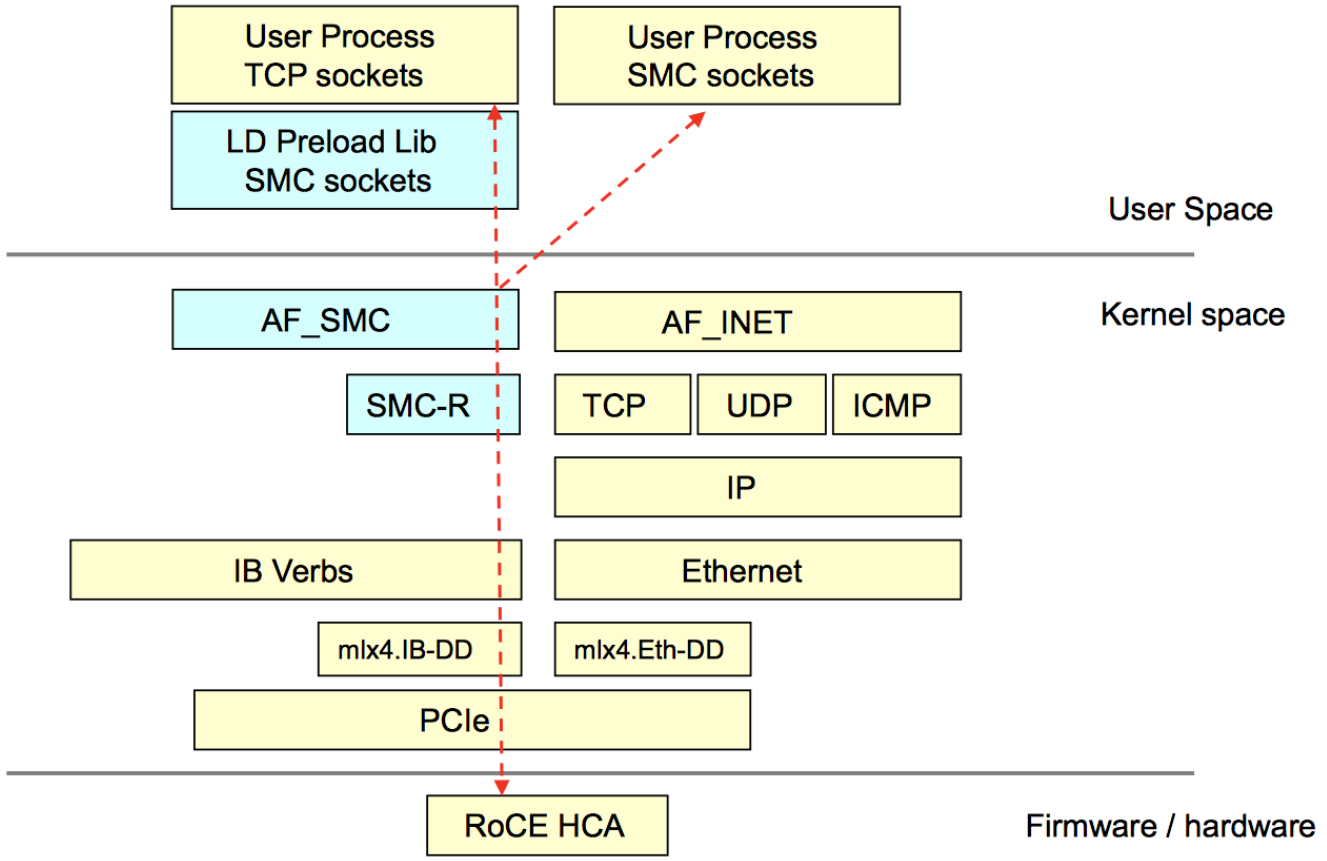
The networking pull request is another significant update. Among the changes that jump out include landing SipHash for greater security in the kernel. The initial SIPHASH usage is for secure sequence numbers and syncookies.

The networking updates for Linux 4.11 also have reduced CPU usage for ICMP replies that are going to be limited or suppressed, a Shared Memory Communications socket layer, improved inet port bind conflict handling, TX batching in vhost_net, continued BPF work, support for port mirroring in the b53 driver, and much more.

For more details on the Shared Memory Communications (SMC) addition, see the patch series. From there, "Shared Memory Communications-RDMA" (SMC-R) protocol as defined in RFC7609. While SMC-R does not aim to replace TCP, it taps a wealth of existing data center TCP socket applications to become more efficient without the need for rewriting them. SMC-R uses RDMA over Converged Ethernet (RoCE) to save CPU consumption. For instance, when running 10 parallel connections with uperf, we measured a decrease of 60% in CPU consumption with SMC-R compared to TCP/IP."

http://www.phoronix.com/scan.php?page=news_item&px=Linux-4.11-Networking

Linux SMC-R Overview



Define responsibilities! And define things properly

- FairMQ Transport
- Data Format

FairMQ Transport, why?

- Hide all the sockets and message transport operation from the user
- Allow non-expert to write messaged based code without going into the details of the transport or the system below
- Offer a **clean** and **maintainable** and **extendable** interface to the existing different data transport (ZMQ, nanomsg, ..etc)
- Allow usage of combinations of transport layers in one device in a transparent way

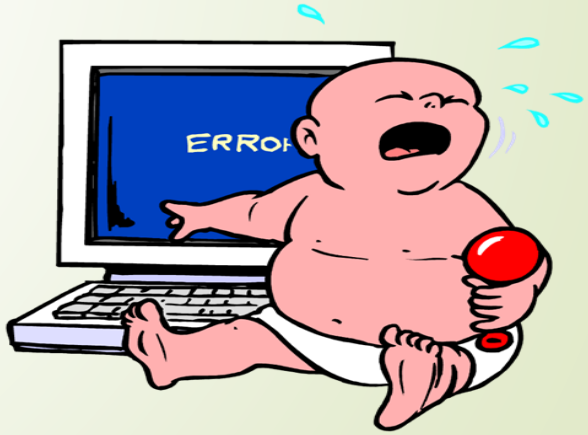
FairMQ Transport layer: Simple ownership model

- Sender device (user code) passes ownership of data to framework with send call,
- Framework transfers to next device, passes ownership to receiver (user code) in onData function call.
- No sharing of ownership between different devices,
- In case of shared memory requires copying with multiple receivers (or read only!).

Data Format

- The FairMQ transport does not know anything about data format
- Does not impose any format on messages.
- Messages are blobs from zero to gigabytes large.

Dropping the multipart (ala ZeroMQ) support in FairMQ?



Multipart in ZeroMQ

- A series of frames with a "more" bit set to one, followed by one with that bit set to zero.
- The ZeroMQ API then lets you write messages with a "more" flag and when you read messages, it lets you check if there's "more".

Multipart in ZeroMQ: lexicon

- A message can be one or more parts.
- These parts are also called "frames".
- Each part is a zmq_msg_t object.
- You send and receive each part separately, in the low-level API.
- Higher-level APIs provide wrappers to send entire multipart messages.

Multipart: ZeroMQ ?

http://hintjens.com/blog:84

- Around the "Free and Open Source Software Developers' European Meeting (FOSDEM 2015)" , a few **core ZeroMQ maintainers** decided to experiment with a simpler messaging API, and decided it'd be profitable to explore **rethinking multipart messages from ZeroMQ**.
- Multipart messages aim to solve a number of problems, and in turn create a number of problems. At the core of ZMQ team shift is the understanding that **the costs now appear to outweigh the benefits.**

http://hintjens.com/blog:84

Problems that multipart solves in ZeroMQ (1)

- It allows zero-copy for high-volume pub-sub data, where the application sends a routing key, followed by a body. With one part, the body must be copied into a buffer. With two parts, the body can be sent directly from an existing buffer (thus, no copying).

Problems that multipart solves in ZeroMQ (2)

http://hintjens.com/blog:84

- It provides a mechanism for multi-hop request-reply, where routing identifiers are pushed to an "envelope", and popped on the return path.

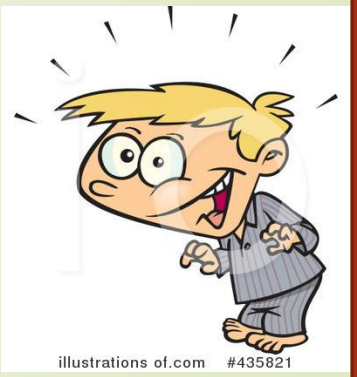
- It provides applications with a simple serialization mechanism, either to avoid using additional libraries, or so that brokers can add and remove information to messages they are forwarding.

Problems that multipart solves in ZeroMQ (2)

http://hintjens.com/blog:84

- It provides a mechanism for multi-hop request-reply, where routing identifiers are pushed to an "envelope", and popped on the return path.

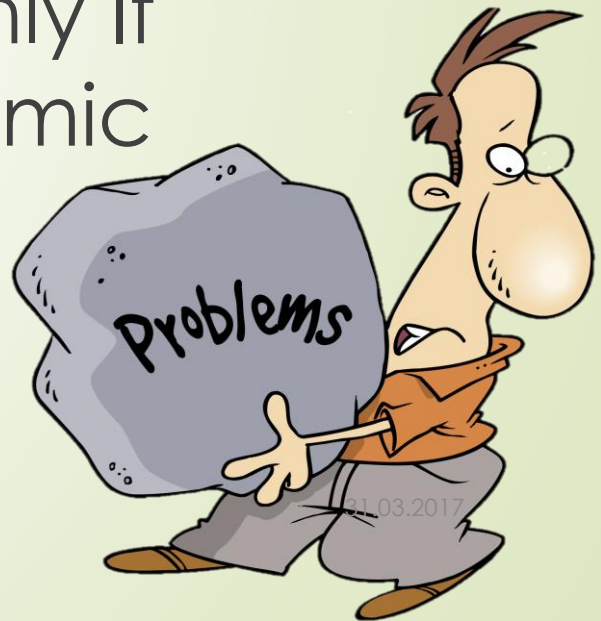
- It provides applications with a simple serialization mechanism, either to avoid using additional libraries, or so that brokers can add and remove information to messages they are forwarding.



Threadsafe Sockets:

http://hintjens.com/blog:84

- With multipart: A socket has to serialize an unknown number of "send" commands from a single caller thread.
- A socket can be thread safe only If receiving and sending is an atomic operation.



Other problems:

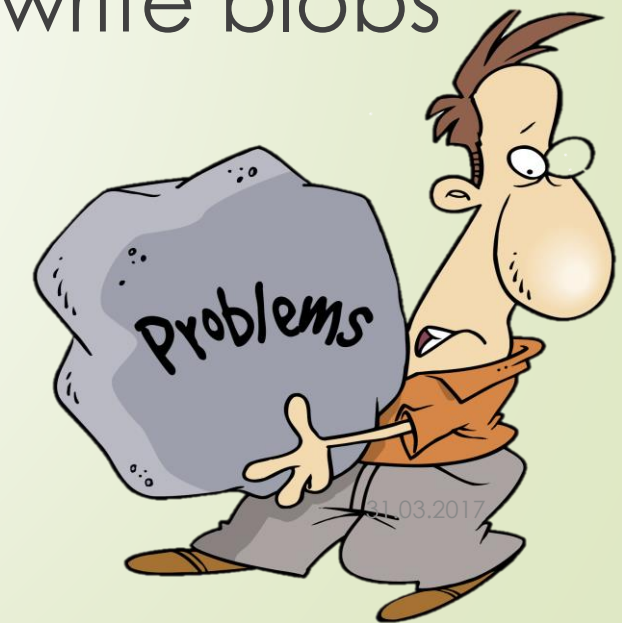
http://hintjens.com/blog:84

- Multipart messages are **counter-intuitive**; the "send" method does not actually send anything; it queues the message until the final part is sent.
- Users often try to use multipart to send large messages. **This does not work**.

More Problems!

http://hintjens.com/blog:84

Multipart data is incompatible with other layers that treat data as a stream of bytes or blobs. For example, the Go I/O interface expects a transport to read and write blobs



Multipart: nanomsg

- Not supported, no plans to have it!
- FairMQ mimic it with MsgPack (copying data around!)

What is next? RDMA

- Work on RDMA transport is ongoing
- Discussions with WP5 (Data distribution and load balancing)

- Different options are under investigation

What is next? Vectored IO

- Vectored IO (scatter-gather) is under investigation
  - A single call reads data from multiple buffers and writes it to a single data stream, or reads data from a data stream and writes it to multiple buffers.
- It will be on the FairMQ level
  - Could re-use the FairMQParts interface but no-multipart in the sense of ZeroMQ

Why vectored (scatter gather) IO

- Cleaner implementation without extra libraries or mixing of data and transport
- FairMQ can offer it on top of all transports without extra costs