

# Open Source for OSD

Dan Messinger

**Seagate**

We turn on ideas



# The Goal

To make OSD technology available to the public.  
(public == anybody outside the small group of developers working on OSD itself)

Requires that OSD drivers be available for a released kernel.

Which in turn requires that the “sockets” that OSD drivers need to plug into exist in a released kernel.

Currently those sockets don't exist, and it is a significant effort to re-invent them for each release of Linux.

# What We Currently Have

IBM open source support:

- Definitions of OSD device type.
- Support for large CDBs.
- Changes to stack to support device scan.
- Loadable driver and library to build CDBs (SO).
- Support for iSCSI 4.0.2 (but it has bugs) to handle bidirectional data and large CDBs.
- All patches and support for Linux 2.6.10.

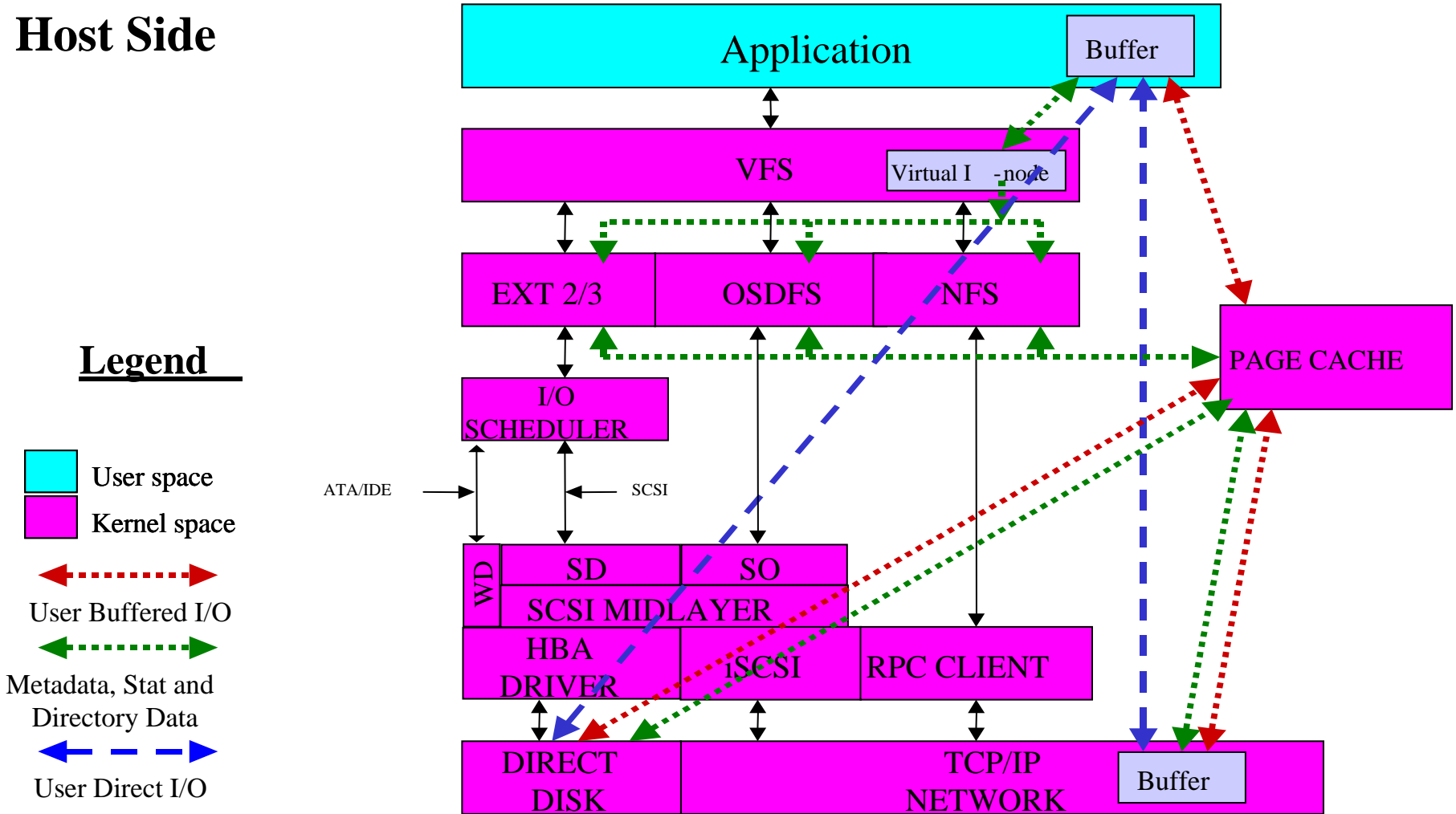
# What We Currently Have (Cont)

## Seagate Open Source Support:

- SG driver support for large CDBs and bidirectional data for debugging (using IBM bidi-buffer structure).
- Linux-iscsi support for multiple R2Ts (above and beyond IBM patches).
- Changes to open-iscsi to support large CDBs, multiple R2Ts and bidirectional data. For use with Linux 2.6.14 and beyond (open-iscsi-1.0-485 release).
- A basic VFS file system that can talk to an OSD via the IBM SO driver (only up to Linux 2.6.12). (Has memory issues without scheduler).

# Software Stack

## Host Side

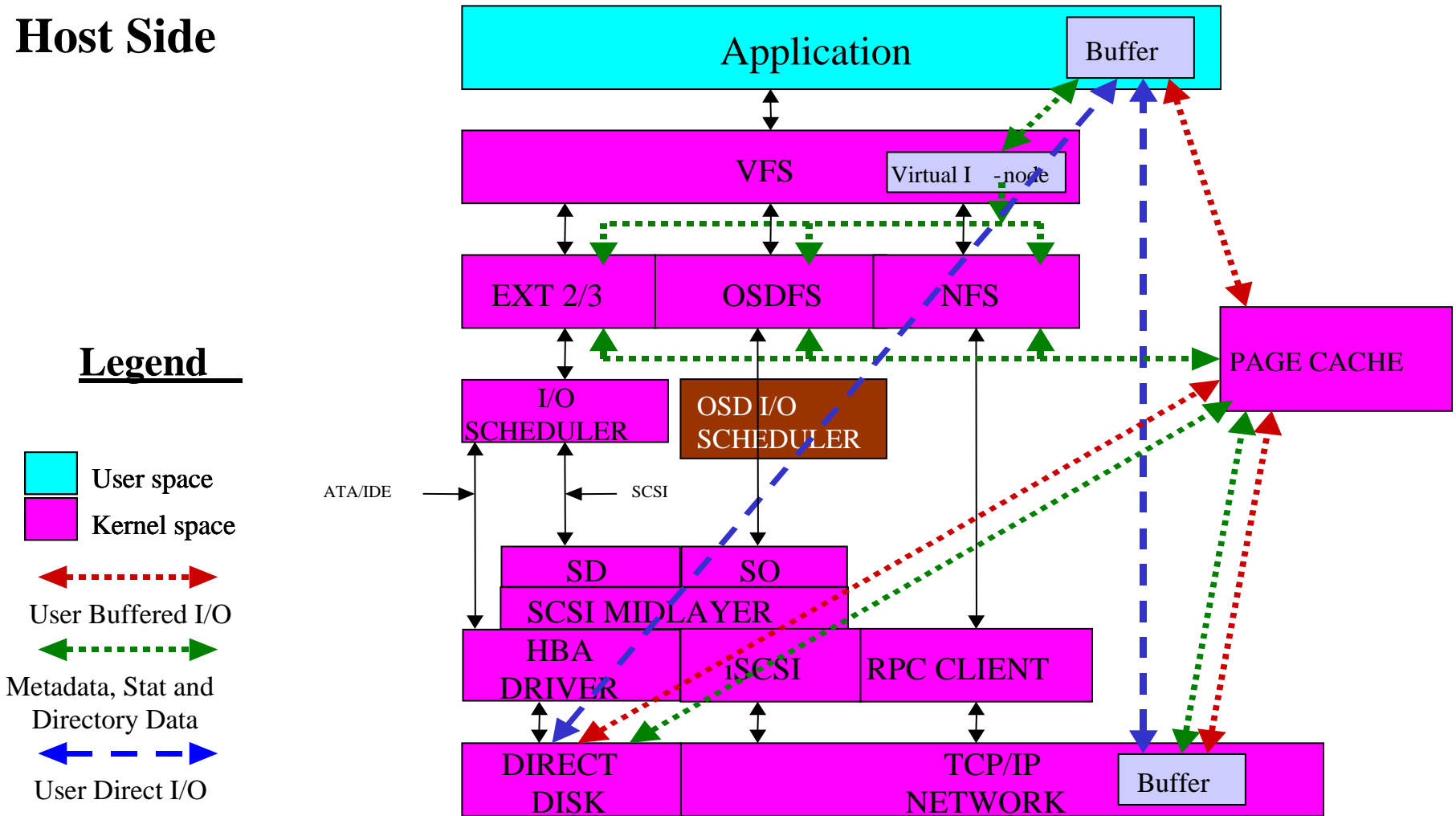


# What We Now Need

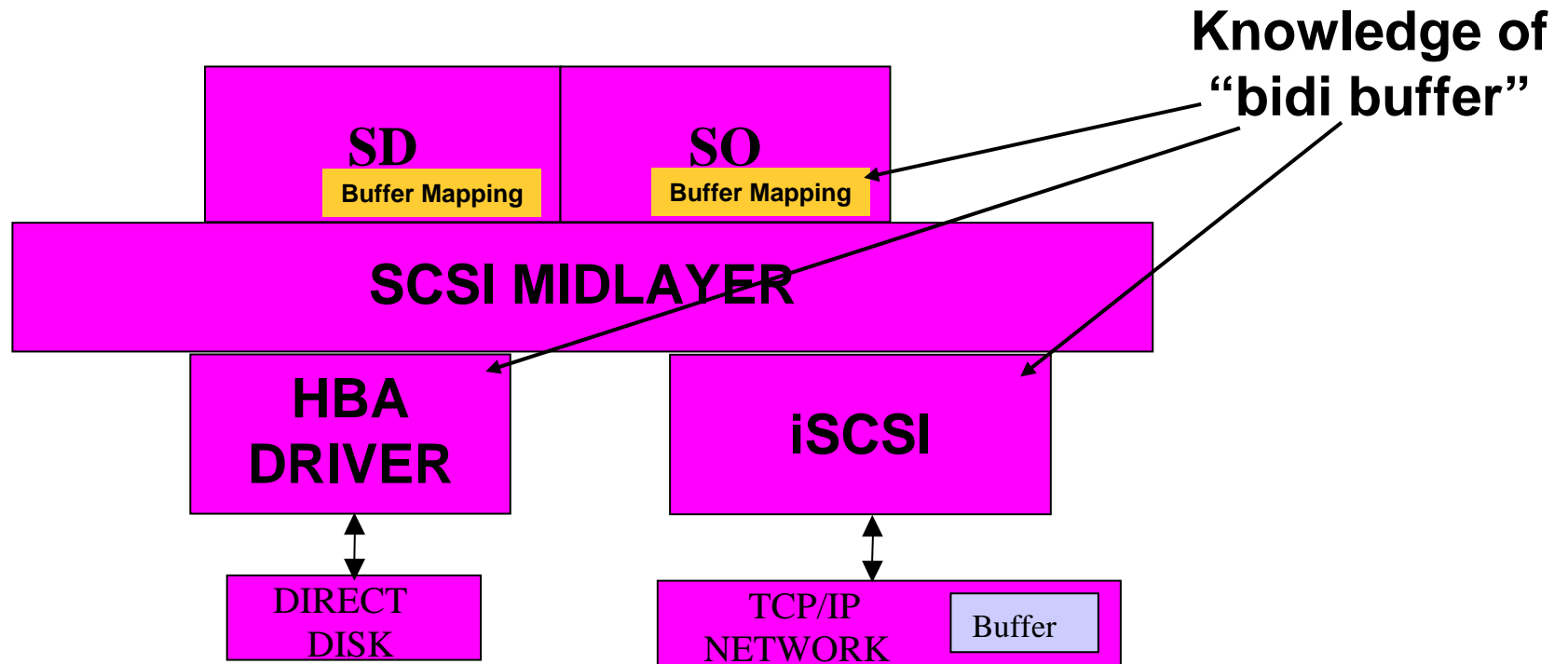
- Open source support for Linux 2.6.18 and beyond
- A fully functional VFS file system for OSD that includes an I/O scheduler (coalescing and throttling).
- Bidirectional buffer support through the SCSI stack that the Linux community supports
- Large CDB support in the stack that the Linux community supports

# Software Stack – OSD I/O Scheduler

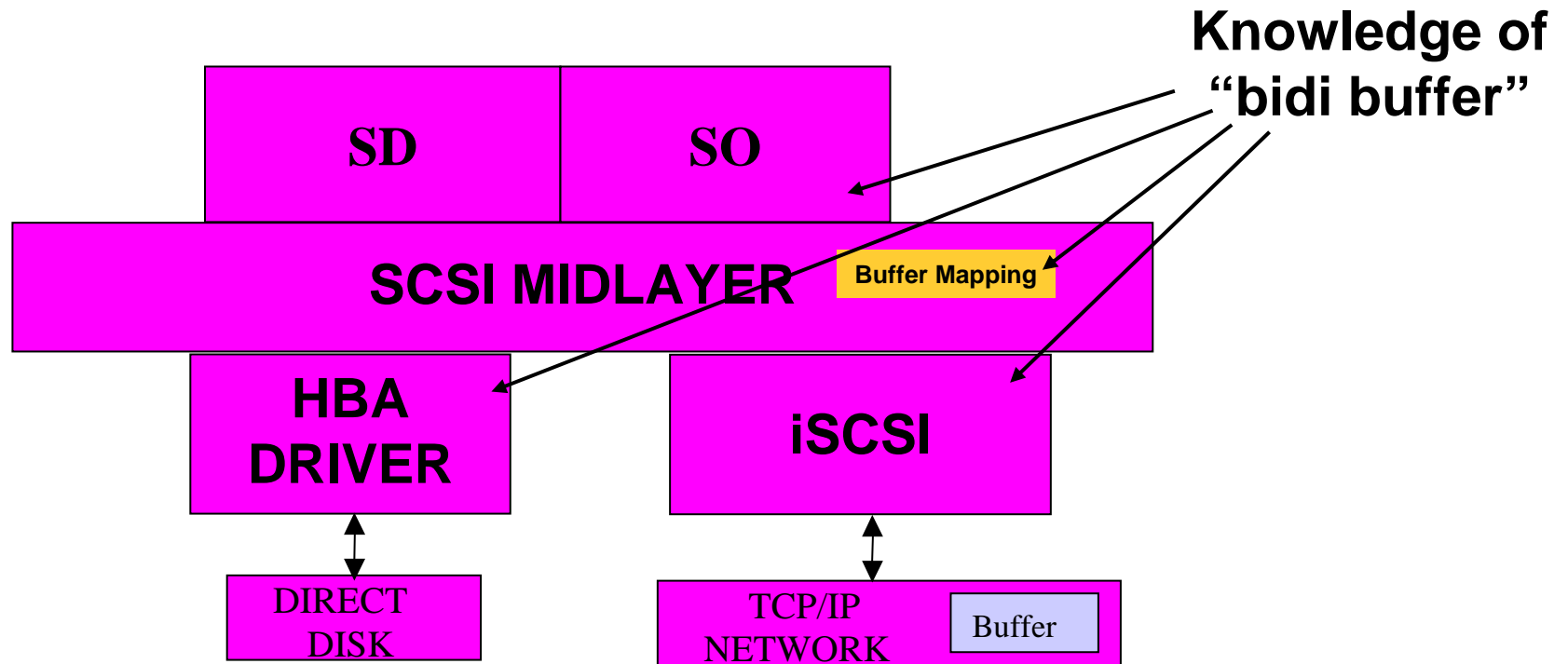
## Host Side



# Software Stack – bidirectional



# Software Stack – Recent Linux change



# Linux changes

- Constant Linux changes are constantly breaking OSD implementation.
- We need SCSI features required by OSD to be part of the core SCSI stack.

# What's in the Works

The Linux iSCSI developers are currently working on:

- Bidirectional buffer support through the SCSI stack  
Pass two pointers (e.g., data-in and data-out) through the stack.  
This is also needed for SAS development.
- No non-s/g. Data pointers ALWAYS point to scatter gather list.
- Support for CDB's larger than 16 bytes by dynamically allocating buffer space.
- Patch set for consideration recently made available for Linux 2.6.18

# Not yet in the works

## I/O scheduler for OSD

- This could take some time. How do you coalesce commands, throttle the number of outstanding commands and give all processes a shot at the OSD? This is by object & offset, not LBA.
- Could require some research to find a good scheduler algorithm.
- Uncertain where this function should fall in the stack.

## Remove dependence on SCSI from “SO” driver

- There may be non-SCSI OSD targets in the future.

## Object extensions to user level interface

# Who needs to be involved

Need to get agreement on a solution from all parties involved with supporting open source:

- Linux SCSI developers  
(Christoph, ...)
- Open iSCSI developers  
(Mike Christie – Redhat, Benny Halevy – Panasas, ...)
- Block Layer developers  
(?)
- IBM (SO driver)
- Emulex (FC driver)
- Other interested parties  
(Seagate, Voltaire)

# What they need to do

Define an implementation that all parties are comfortable with.

Modify ALL effected SCSI modules and submit them to be included in Linux.

Update SO driver to be make use of the above, and ideally not require further patches to the kernel (IBM).

Update Emulex FC driver to support OSD using the above (Emulex).

Fixes to Open-iSCSI to correctly support R2T's.