OpenSolaris NFS/RDMA

http://www.opensolaris.org/os/project/nfsrdma/

Mahesh Siddheshwar
NFS Development

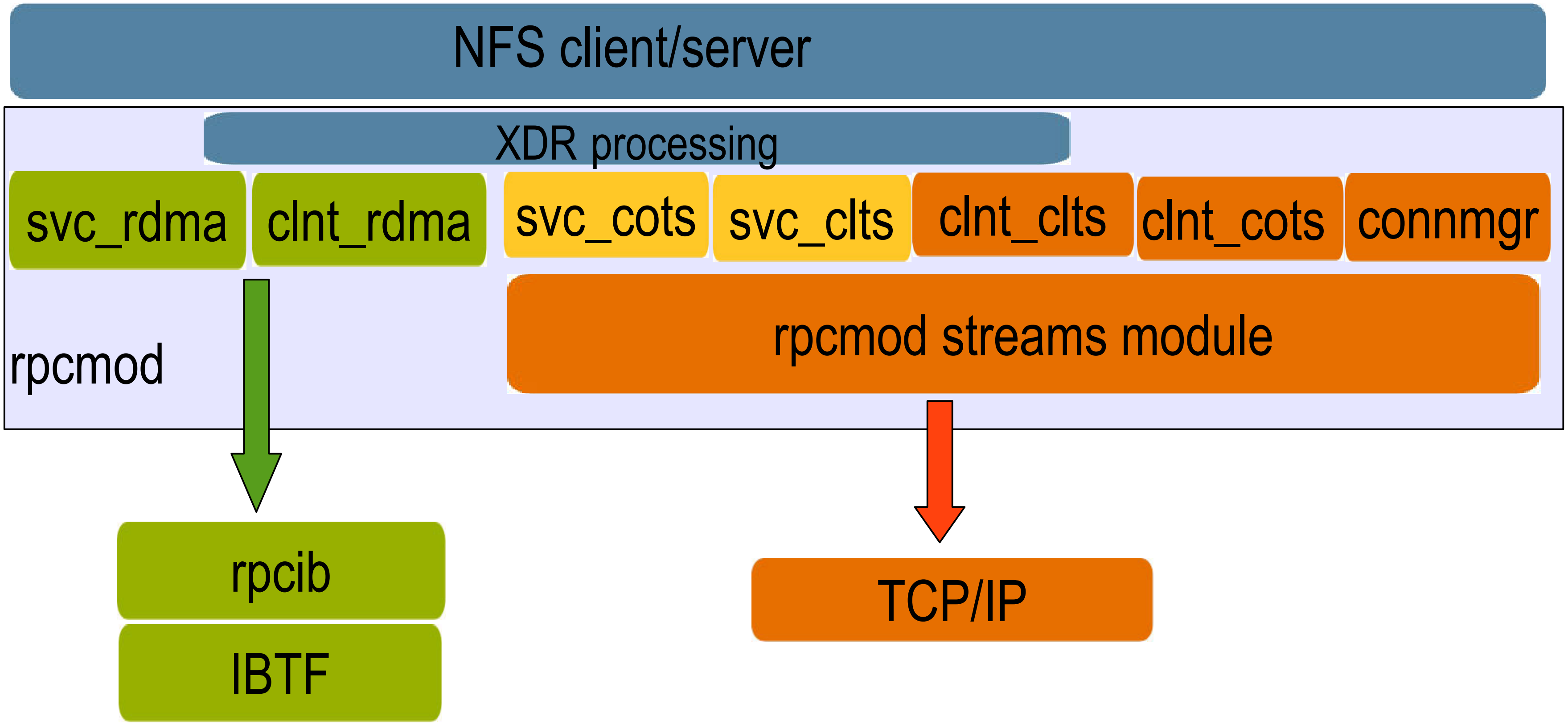
Agenda

- Introduction
- OpenSolaris NFS/RDMA Basics
  - > RPC/RDMA
  - > NFS/RDMA
- Current status and WIP
  - > Linux Interoperability
  - > Future work
  - > pNFS design considerations

OpenSolaris NFS/RDMA

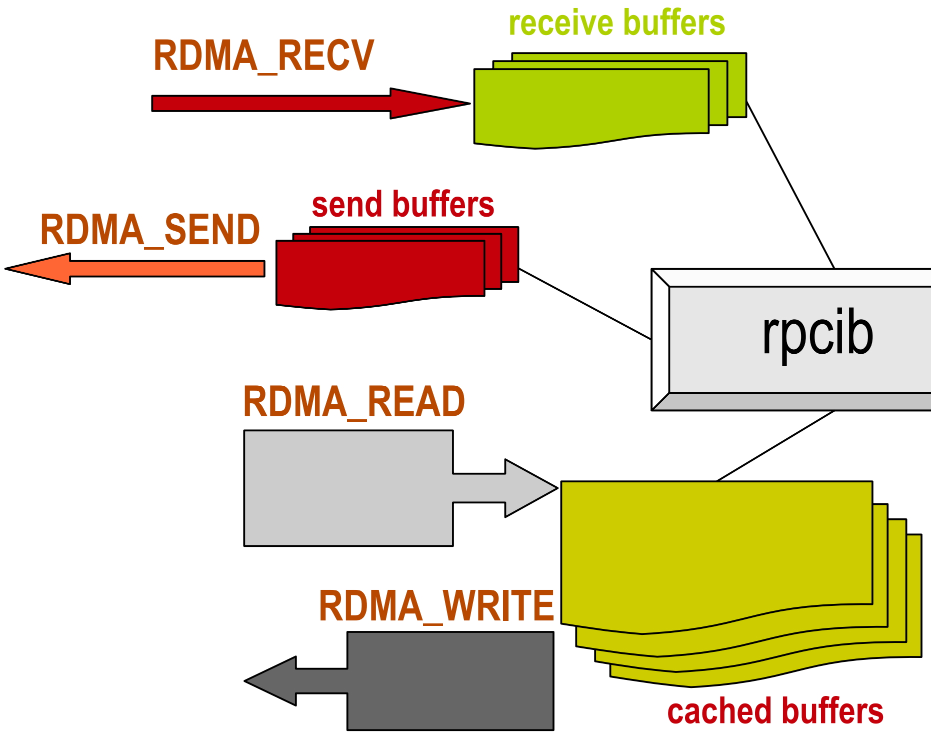
- In Solaris Nevada since snv_b98
  - Initial prototype from OSU
  - OpenSolaris 2008.11
- In compliance with the two IETF drafts
  - Remote Direct Memory Access Transport for Remote Procedure Call
    - http://tools.ietf.org/html/draft-ietf-nfsv4-rpcrdma-09
  - NFS Direct Data Placement
    - http://tools.ietf.org/html/draft-ietf-nfsv4-nfsdirect-08
- Support over IB
  - Default proto=rdma; IPoIB with 'proto=tcp'

NFS/RDMA components

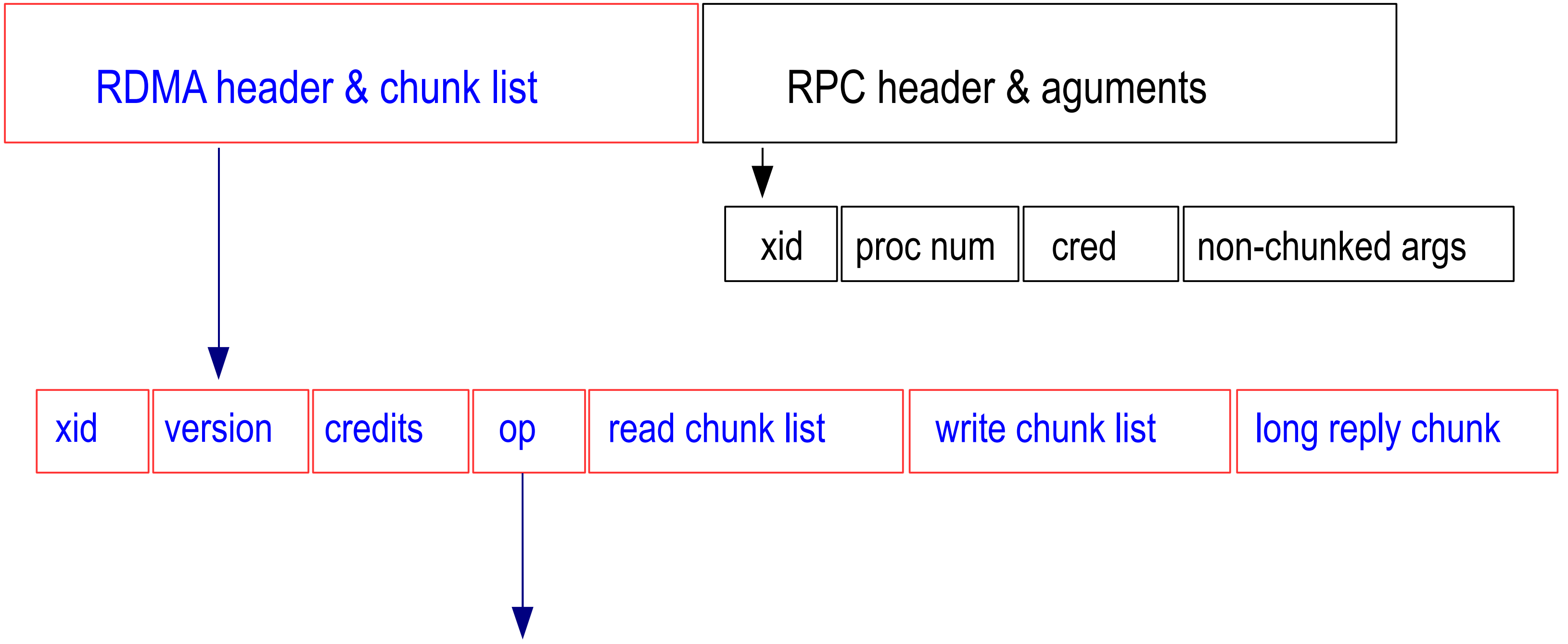


RDMA transport

- Registered memory
  - > 32 bit steering tags, 64bit memory addr
- RDMA SEND
  - > Receiver signaled on completion
  - > Ordered delivery
- RDMA READ
- RDMA WRITE
  - > Ordered w.r.t to RDMA SENDs
- Interface provided by rpcib driver
  - > Uses interfaces provided by IBTF



RPC RDMA header



RDMA header & chunk list

RPC header & aguments

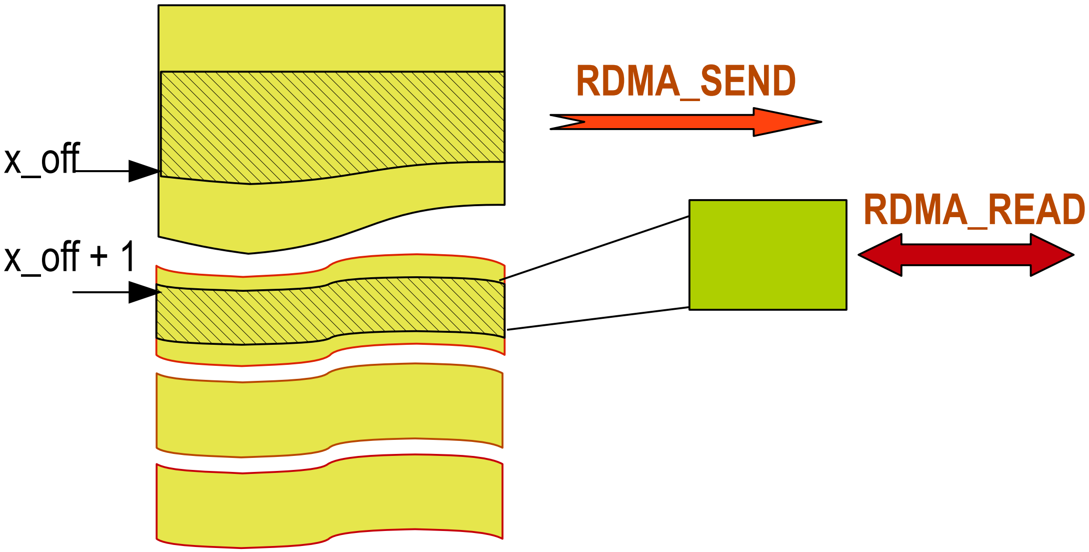
xid	proc num	cred	non-chunked args

xid	version	credits	op	read chunk list	write chunk list	long reply chunk

RDMA_MSG, RDMA_NOMSG, RDMA_MSGP
RDMA_DONE, RDMA_ERROR

RPC/RDMA

- Short messages (<1K)
  - RDMA_SEND to a pre-posted buffer
  - Inline RPC message
- Read Chunk list
  - xdr_rdma_segment, xdr offset
- Write Chunk list
- Long reply chunks (> 1K)

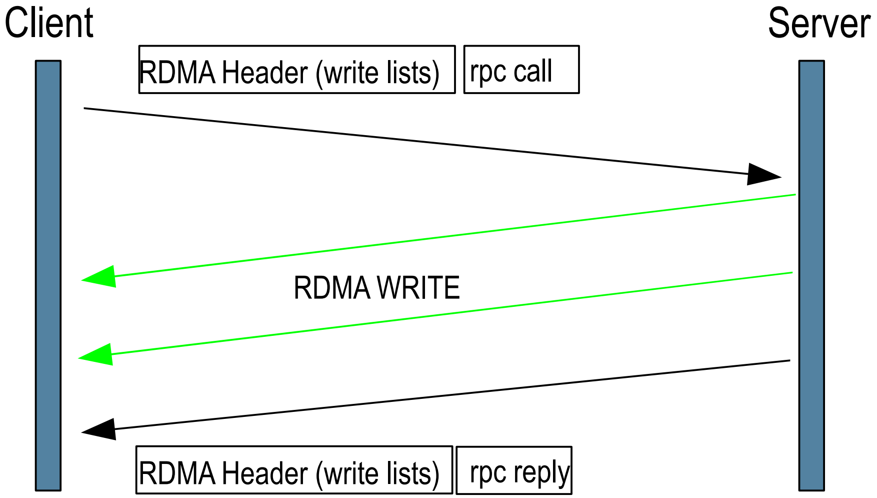


RPC/RDMA ↔ NFS mapping

- NFSv3
  - Read chunk list (WRITE, long RPC call)
  - Write chunk list (READ)
  - Long reply chunk list (READDIR, READDIRPLUS, READLINK)
- NFSv4
  - Read chunk list (WRITE, long RPC call)
  - Write chunk list (READ)
  - Long reply chunk list (READDIR, READLINK, COMPOUND)

NFS/RDMA

- NFS reads
  - Client posts a write chunk list
  - Server transfers the data to the client using RDMA_WRITE
  - Server notifies the client with inline reply



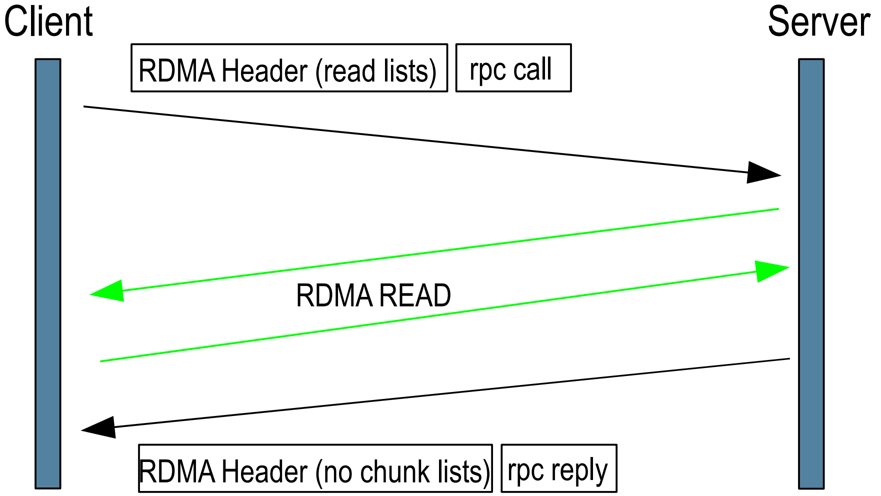
NFS/RDMA

- xdr_READ[3,4]args() adds the chunk to the XDR handle through a XDR_CONTROL()
- flagged and moved as a write chunk list during the CLNT_CALL()
- data directly placed in the uio buffers or file pages

```
struct READ3args {
        nfs_fh3 file;
        offset3 offset;
        count3 count;
#ifdef _KERNEL
        /* for read using rdma */
        char *res_data_val_alt;
        struct uio *res_uiop;
        struct clist *wlist;
        CONN *conn;
#endif
};
```

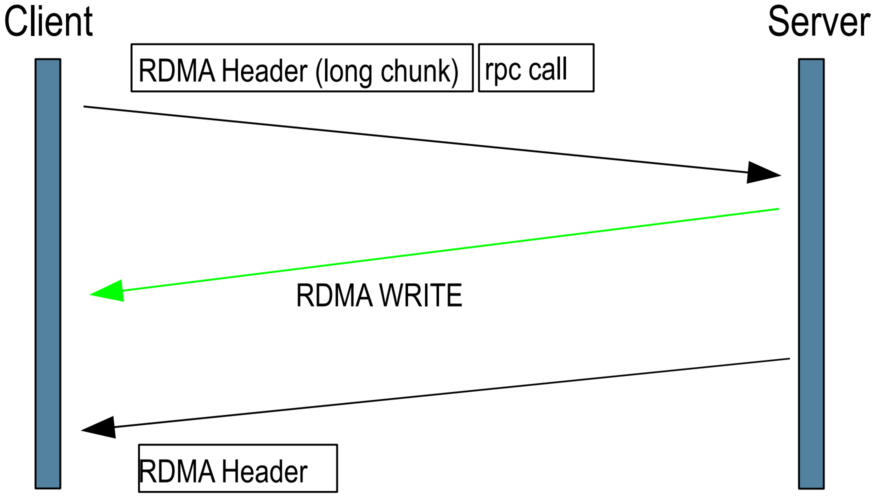
NFS/RDMA

- NFS writes
  - > Client posts a read chunk list
  - > Notifies the server with inline RPC call
  - > Server reads the data from client using RDMA_READ



NFS/RDMA

- NFS readdir
  - > Client posts a long reply chunk
  - > Server response through RDMA_WRITE
  - > Notifies the client with inline RPC call



Prelim. performance results

- Results: (http://opensolaris.org/os/project/nfsrdma/performance/)
  - Writes : ~ 1GB/s
  - Reads: 1.3GB/s
  - DDR IB with memory based filesystem (tmpfs) (with dircetio)
- Configuration details:
  - Sun X2200M2 servers (AMD Opt. 2.6 GHz x2)
  - 8 GB memory
  - Mellanox ConnectX HCAs (hermon)
  - Voltaire DDR switch
  - Solaris onnv b101
  - iozone v. 3.311 (customized)
  - tmpfs

Linux Interoperability

- Linux 2.6.27 vs. OpenSolaris server
- Change to use IANA assigned port # 20049
- XDR encode/decode differences
  - Chunk list management
  - roundup/padding issues
  - Linux NFSv4 client link/rename issues

v4 COMPOUND: PUTFH WRITE [4109 bytes] GETATTR

```
                  - chunk1 - 4k
write data |
                  - chunk2 - 13 bytes(4109 - 4k)
getattr op  - chunk3 - 19 bytes (getattr op starts at byte 4)
```

Current WIPs

- SPARC IB/DR project in snv_109
  - Ability to configure/unconfigure IB HCAs
- Dynamic rdma credit negotiation?
- pNFS/RDMA?
  - resource/rdma credit control
  - connection to sessions binding
  - bi-directional RPC and trunking

NFSv4.1/pNFS RDMA

- fore channel only?
  - numbers of DS > MDS
  - small sized recalls from DS, why pin down recv buffers on the client?
- fore channel – bursty or one-time i/o?
  - how long should a rdma channel stay around?
  - rdma-hibernate – reduce the # of credits (recall slots)
  - re-negotiate rdma credits on demand
  - use backcahannel tcp conn for fore-channel?

NFSv4.1/pNFS RDMA

- trunking
  - different interfaces and transports
  - IB/Ethernet vs. rdma/IPoIB

- clientid trunking vs. session trunking
  - session trunking: ca_maxrequests restricted to number of rdma credits
  - session trunking: choosing of the i/o path?
  - clientid trunking easier? ☺

- long reply buffer considerations

OpenSolaris NFS/RDMA

http://www.opensolaris.org/os/project/nfsrdma/

Mahesh Siddheshwar
maheshvs@sun.com