HR-NET: A Highly Reliable Message-passing Mechanism for Cluster File System

Jiang Zhou, Can Ma, Jin Xiong, Dan Meng

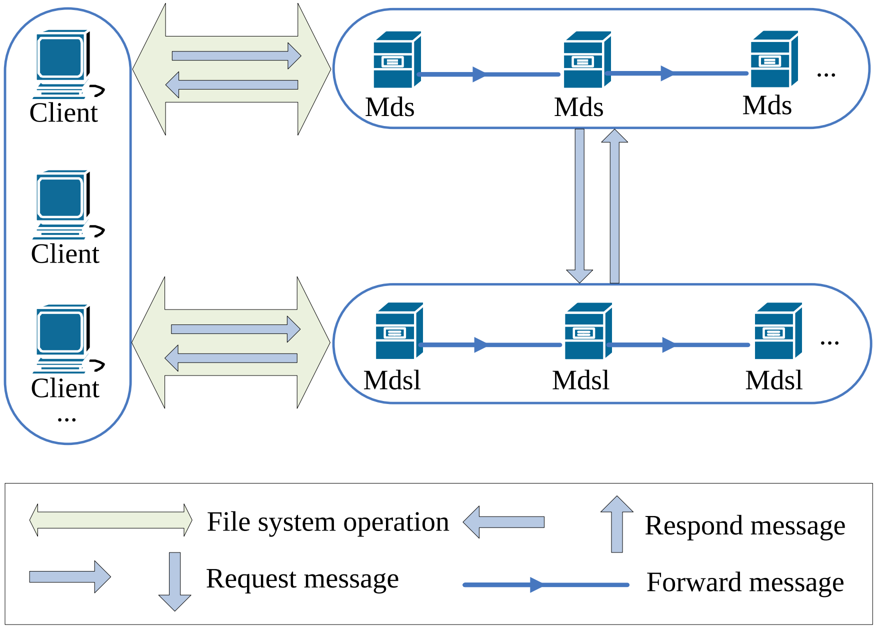
7/29/2011

Outline

- Introduction
- HR-NET overview
- Network Failures Scenarios
- Reliable mechanism
  - Fault tolerance
  - Message priority scheduling
- Evaluation
- Related works
- Conclusion

Why reliability

- File access in a cluster file system often contains several sub-operations
- Each includes one or more network transmissions(request/response)
  - Between Client, Mds (metadata server) and Mdsl (metadata storage layer)
- Any network failures will cause the file system service unavailable



Message-passing framework in cluster file system

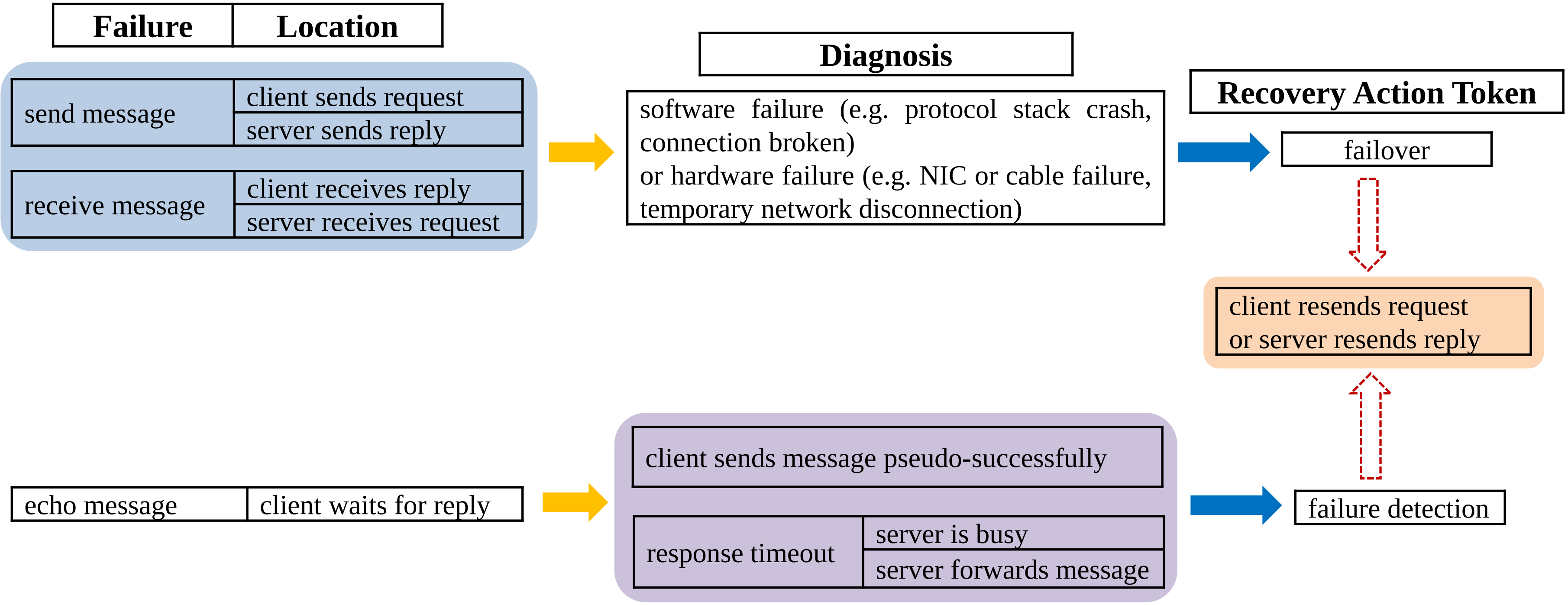
What is HR-NET

- HR-NET is
  - a subsystem supporting for network transmission in HVFS
    - HVFS(Pomegranate) is a cluster file system for Dawning 6000 supercomputer
    - Decouple data and metadata operations
    - Is mainly designed for managing tiny files
  - an efficient, stable and fast communication library
  - Completely transparent to upper applications
- Support
  - Gigabit Ethernet, Infiniband and etc
  - Kernel mode(.ko) and user mode(.a) for Linux
- Goal
  - Tolerate both software and hardware network failures
  - Provide high reliability

Reliable mechanism of HR-NET

- Fault tolerance (w/ failure detection and recovery)
  - Provide fine-grained, connection-level failover across communication path redundancy
  - FS can keep passing messages until it either recovers from network failures or it is failed over to a backup
  - Load balance for messages
- Message priority scheduling
  - Dynamically manages messages in an appropriate order to tolerate request-response failures between clients and servers
  - Enable important messages to be dealt first while others to be handled in a given period of time
- Some other auxiliary actions
  - Retransmission, timeout detection, asynchronous message-passing and etc.

Network Failures Scenarios Analysis



Failure	Location
send message	client sends request server sends reply
receive message	client receives reply server receives request
echo message	client waits for reply

Diagnosis

software failure (e.g. protocol stack crash, connection broken)
or hardware failure (e.g. NIC or cable failure, temporary network disconnection)

client sends message pseudo-successfully

response timeout	server is busy server forwards message

Recovery Action Token

failover

client resends request
or server resends reply

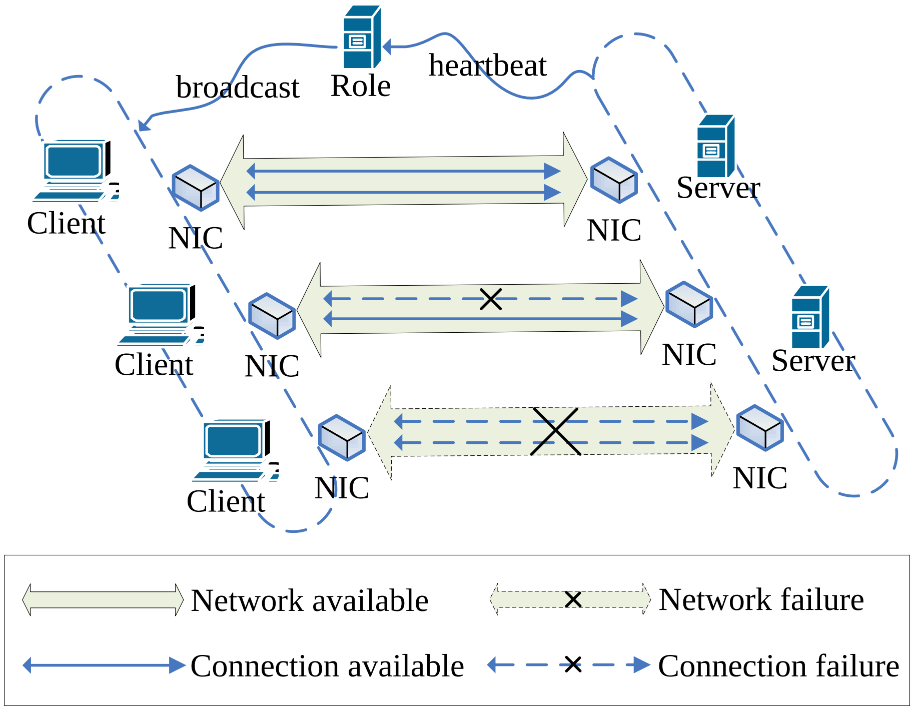
failure detection

Failure detection and recovery

- Communication path redundancy
  - For the node which is connected to one network, builds multiple connections within the same network
    - If software failures occur in one, break it and selects others to continue transmission
    - A daemon is running to monitor the state of each connection and tries to re-establish the broken one periodically
  - When a node is connected to two or more networks
    - Uses different networks to build various connections and pass messages simultaneously
    - A special “heartbeat” mechanism is maintained
      - One node of servers is elected (from conf) as the "role server”
      - Others broadcast heartbeat messages from their network interfaces periodically
      - Messages between clients and servers are also treated as valid heartbeats
    - Communication path are built or broken dynamically

Failover with communication path redundancy

- The “role server" gathers network status and broadcasts to clients
- Client updates its routing table (first be created from conf) and uses alternate paths for transmission
- A balance algorithm is used to distribute messages among different paths

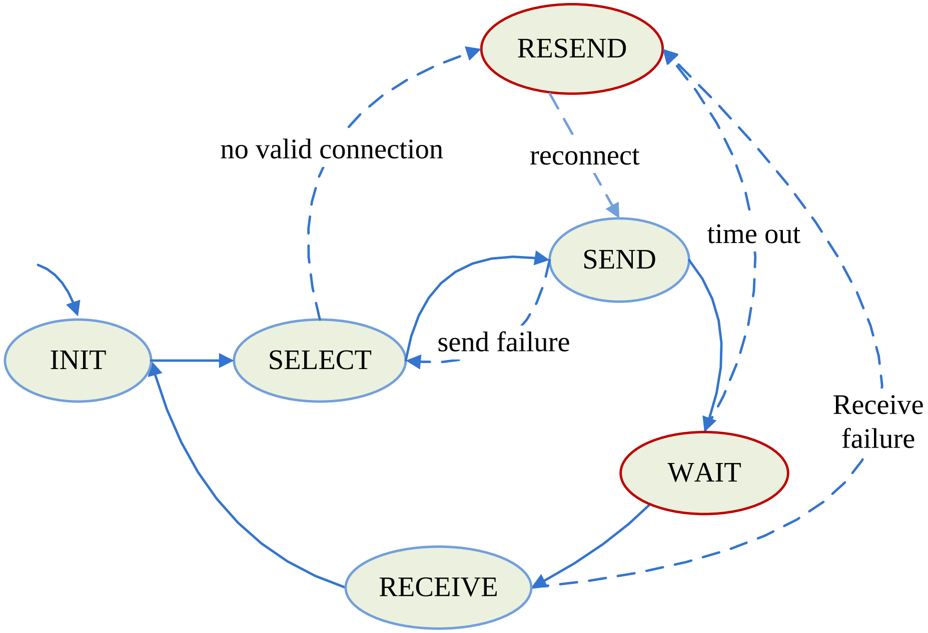


Message priority scheduling

- Message priority scheduling
  - To tolerate echoing message failures (request-response)
  - Dynamically manages the message in proper order
  - Important messages can be dealt first
  - Others to be handled in a given period of time
- Describe the implement of it associated with message state transition graph
  - At client
  - At server
- Prevent incorrect message state transition

Message state transition (client)

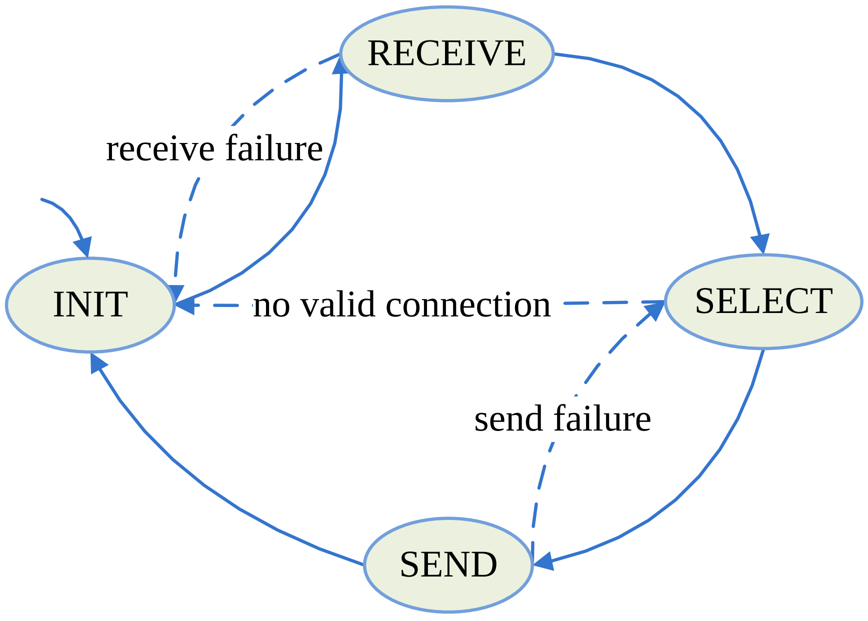
- Mostly clients will expect response from the server after sending requests
  - If no response comes after a timeout period, clients would resend messages
  - Perhaps the server is busy, forwards messages or others
  - Clients often do not receive replies in time and will re-transmission
- Cause the false loop of message state



(I) client

Message state transition (server)

- Due to the retransmission dependence of the client, the server does not need to resend messages when network fails



(II) server

Adjustment of message priorities

- Client
  - The rank "normal" means a low priority while "high" indicates a high one
  - Any messages with high ranks are submitted prior to others at clients
  - For synchronous transmission, no need to adjust the priority

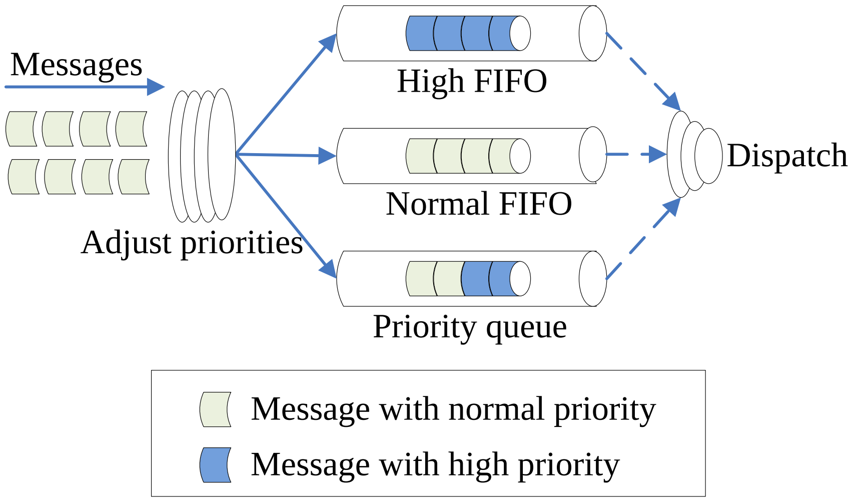
		Default Priority	Adjustive Priority
INIT		normal	normal
SELECT		normal	normal
SEND		normal	normal
RESEND	RECEIVE ERROR	normal	high
	SEND ERROR	normal	normal
	TIMEOUT	normal	high

- Server
  - Sets different priorities according to request types (from client)
  - requests with high ranks are processed first

	Request Type	Default Priority	Adjustive Priority
INIT		normal	normal
RECEIVE	SEND	normal	normal
	RESEND	normal	high
	FORWARD	normal	high

Queues for message priority scheduling

- Client (for synchronous mode and retransmission)
  - requests are added to a message queue and wait to be submitted
- Server
  - maintains a message queue to receive arrived messages
- priority queue
  - When a message is changed into high priority, it is moved to the head of the queue and handled prior to others both at the client and server
- FIFO queues
  - Avoid scheduling starvation
  - Timeout detection



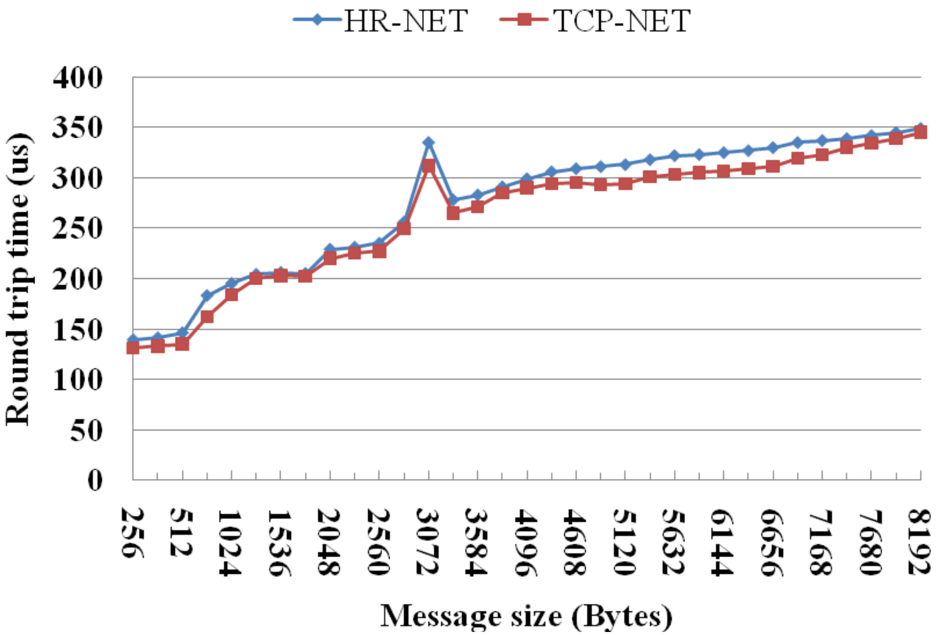
three queues for message priority scheduling at client or server

Performance

- Environments
  - Two nodes
  - Each has four Intel Xeon E5310 Processors, 4-Gbyte memory and two 1000 Mbit NICs
  - Linux kernel 2.6.18
  - HVFS configuration (one node with MDS and MDSL, another with client)
- Three categories
  - Overhead introduced by reliable mechanisms
    - Implement an original TCP-NET library as the baseline
  - Failover time it takes with redundant communication paths
    - Choose Lnet as the reference
  - Overall measurement based on the cluster file system HVFS

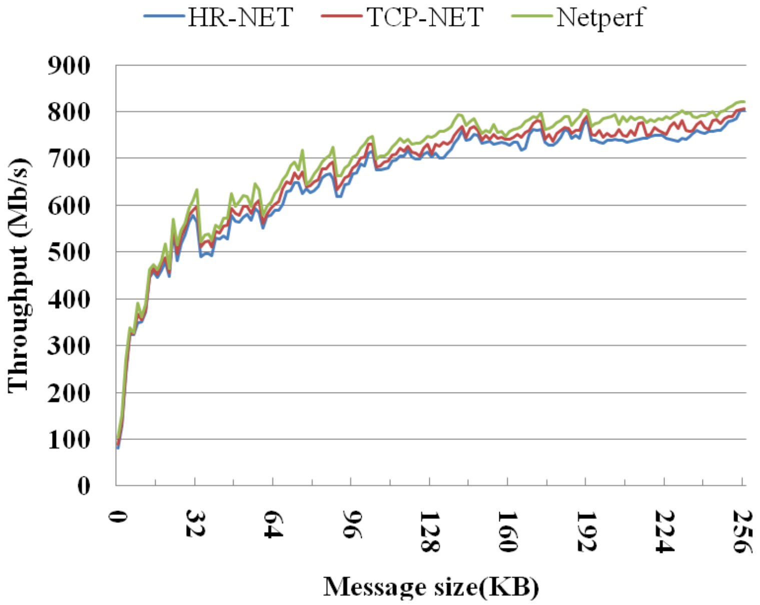
Performance overhead(round trip latency)

- There is an abnormal increase in latency sometimes
  - Due to the fragmentation and reassembly mechanism of IP layer
- HR-NET exhibits a round trip time that is 3 to 19 us slower than TCP-NET
  - Use additional threads, redundant connections and message queues for fault-tolerance
  - Affect the latency even if no network failures occur



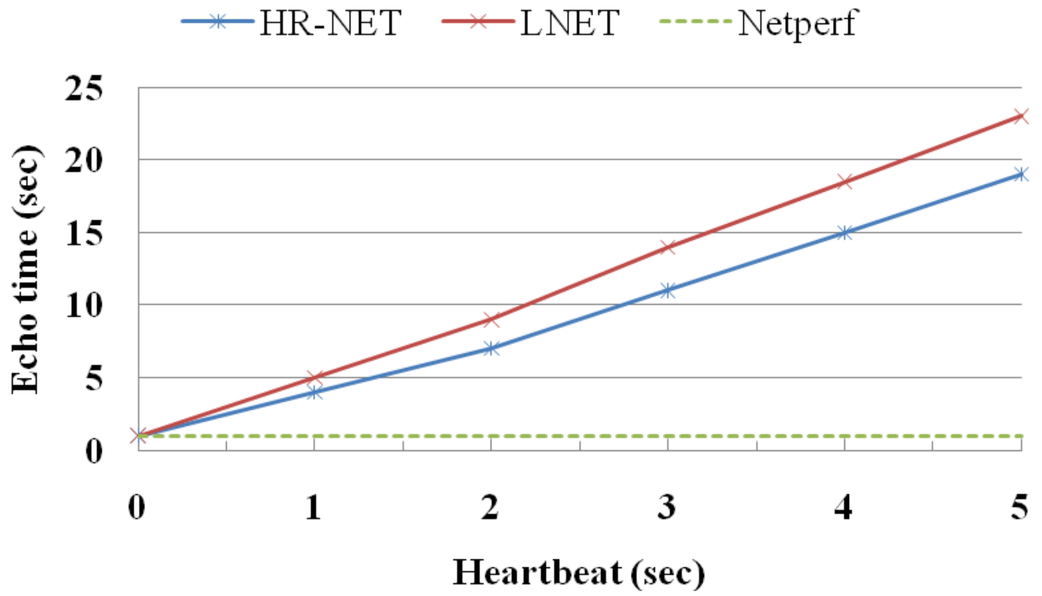
Performance overhead(throughput))

- Netperf indicates the practical bandwidth
- HR-NET
  - Achieved maximum performance with 802.15Mb/sec
  - Shows an average throughput decrease by 6.17% compared with TCP-NET
- It is worthy for this cost since the system has provided high reliable mechanism while the performance has only been degraded by 8.19% at the most



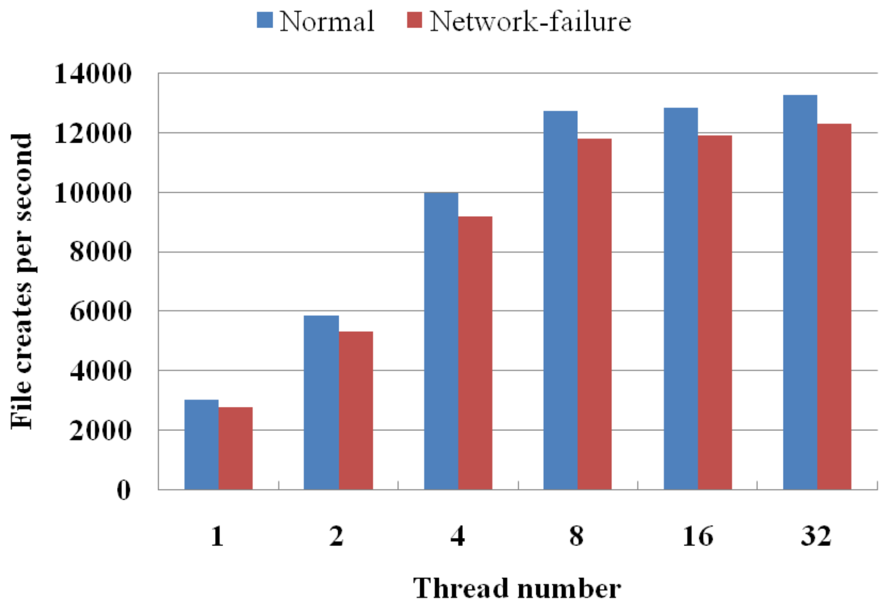
Failover time

- Specified in-progress TCP connections were killed
  - Use tcpkill command in the toolkit Dsniff
- Failover time depends on
  - HB frequency
  - Round trip latency (omitted)
- The lower green curve shows the failure free case
- Fine-grained, connection-level failover provides a faster recovery



Overall measurement

- Simulation of network failures by taking out/plugging back network wires, breaking connections
- Use IOPS of metadata operations in HVFS with Mdtest-1.8.3
- Each thread create 100K files
- The system can recover form failures with HR-NET and performance degradation is less than 8%



Related works

- Aims at transport-level protocols (TCP)
  - Connection migration , FT-TCP and ST-TCP
  - Primary-backup approach
  - Require re-implementation or changes to standard TCP protocol stack (at server)
- Based on software binding of multiple Ethernet links
  - RI2N: difficult to detect failures at link level
  - HA-NFS: Overhead for switch between links is high
- Combine network fault-tolerant mechanism with cluster file system
  - NFS: based on RPC and can not deal with network partition
  - Lustre: expensive cost for redundant network routing
  - PVFS2: address fault tolerance at level of file system and not fit for scalability

Conclusions

- HR-NET
  - is a highly reliable message-passing mechanism
  - detect and recover network failures by
    - fault tolerant mechanism
    - message priority scheduling strategy
  - Ensure availability of each pair transmission and sub-operations in a cluster file system
  - completely transparent
    - neither any changes to standard protocol stacks nor modifications at the client or server
  - can obviously improve the system reliability while keep the performance with little degradation
  - but is not appropriate for bulk data transmission with various timeout mechanism

Thank You

Q & A