

# Cassini UDP Performance

## A case study

Kevin Colwell  
April 2004

# Disclaimer:

The tuning examples provided in this document do not constitute an endorsement or recommendation by me or Sun Microsystems. Use of any of these examples is at your own risk.

NO GUARANTEE IS GRANTED AS TO THE BEHAVIOR RESULTING FROM THESE SYSTEM AND DRIVER TUNING PARAMETERS, AND NO WARRANTY SHALL BE ISSUED, IMPLIED OR OTHERWISE, BY THE AUTHOR, OR BY SUN MICROSYSTEMS.

# The Challenge

Customer asked us to demonstrate “wire speed” data transfer over gigabit Ethernet.

# The Challenge

Customer asked us to demonstrate “wire speed” data transfer over gigabit Ethernet.

More specifically:

- *Receive* UDP packets at up to 942Mbps
- Store data to disk
- Zero data loss
- Packet size of 1080 bytes

# The Challenge

To test the receiver, we must also have a sender. Same requirements apply.

Specifically:

- *Send* UDP packets at up to 942Mbps
- Zero data loss
- Packet size of approx. 1100 bytes

# The Application

- Customer provided “recorder” app
  - Incoming packets stored in memory
  - When enough packets are available, switch to next buffer and issue write to disk
  - Circular packet buffers, disk I/O buffers

# Out-of-the-box Experience

First time we ran the application

- Sender topped out at 574Mbps
- Receiver got only 70% of the packets
  - Lost 30,000 out of 100,000 packets!

# Tuning the system

By inspecting various counters (from kstat or netstat -k output), and comparing actual packet counts with expected counts, we were able to determine which resources was being exhausted and should be adjusted.



# Tuning the system

- Application

- Multithreading, to ensure that a thread is dedicated to packet receiver
- Sender multithreaded as well, but this broke packet ordering

- Driver

- set ce:ce\_ring\_size=2048
- set ce:ce\_comp\_ring\_size=8192
- set ce:ce\_srv\_fifo\_depth=8192
- set ce:ce\_inst\_taskqs=1

# Tuning the system

- UDP

- `ndd -set /dev/udp udp_max_buf 536870912`
- `ndd -set /dev/udp udp_do_checksum 0`
- `ndd -set /dev/ce rx_intr_pkts 128`
- `ndd -set /dev/ce rx_intr_time 1800`
- `ndd -set /dev/udp udp_do_checksum 0` (on transmit side)

- STREAMS

- Increase the size of the STREAMS synchronized queues
  - `set sq_max_size=0`

# Tuning the system

- System – CPU management
  - Create processor sets:
    - `psrset -c 0 1`
    - `psrset -c 3`
  - Restrict interrupt handling:
    - `psradm -i 0 1 2`
  - Bind the application to a processor set:
    - `psrset -b 1 <pid>`

# Results

Zero UDP packet loss at these rates:

- 691Mbps sustained for 24 minutes
  - With customer application and system tuning
- 940Mbps sustained for 1 to 2 minutes
  - With heavily modified application
  - Used bufmod(7M) STREAMS Buffer Module
  - Bottleneck on write to disk, not on network
- 864Mbps sustained for 30 minutes
  - With modified application

# Conclusion

Cassini hardware and drivers can support wire speed data transfer.

UDP can be used reliably, but capturing the incoming packets at wire speed was very challenging.