Factors Governing Throughputs

(or the search for the perfect client)

Brian Pawlowski
beepy@netapp.com

www.netapp.com

This Talk

- **NFS Benchmarks**
- **A simple throughput study**
- **Throughput limits and factors affecting them**
  - **Read and write**
  - **Options (FDDI, 100BaseT, TCP, UDP and xfer size) and their effect**
- **Future**

NFS Benchmarks

- **LADDIS — multi-user workloads**
  - **Configurable, synthetic benchmark**
  - **Measures only the server**
  - **Useful to compare vendor offerings**
  - **Difficult to setup**
- **Throughput tests**
  - **Microbenchmarking of read and write**
  - **Measures the client and server**
  - **Most focus on single client results**
  - **Easy to setup**
- **User application benchmarking**
  - **Best predictor for customer**
  - **Measures the client and server**
  - **Difficult to setup**

My interest?

- **Speed, speed, speed, maximum speed.**
- **Customers are benchmarking systems during evaluation**
  - **LADDIS is intractable**
  - **For some throughput measure is a better predictor of their application performance**
- **Shooting performance problems at customers**
  - **Simple throughput tests often suffice**
- **Effect of changes being made to NFS?**
  - **NFS Version 3 introduced async writes, and large transfer sizes**
  - **TCP becoming the default transport**
  - **100BaseT on the rise**

And finally...

- **Network Appliance is a server company**
  - **At the mercy of client implementations**
  - **Wants to see increased investment in client performance analysis and tuning**
  - **Will work with anyone and share data to achieve this**
- **Start a dialogue on factors governing throughput**
- **Encourage default configuration tunings to be optimal**
- **With 100BaseT ascendant and Gigabit ethernet coming fast, I want to lay groundwork for awesome throughput performance**
- **Find better clients!**

Experiment you can try at home

- **Simulation of a perfect server**
  - **Export "tmpfs" — memory-based file system**
  - **Reduce operations to cached memory access**
- **The perfect client?**
  - **UltraSPARC 1 — can saturate 100mb/s link**
  - **Tunable (and good) read-ahead and write-behind.**
- **Benchmarks**
  - **simple_read and simple_write — do no work, throw data away, source available**
  - **awk scripts tabulate data**

THESE NUMBERS ARE OPTIMISTIC

Is the technique worthwhile?

- **Yes. First, the client is unmodified and with a perfect server you can explore client performance issues.**
- **Second, you can compare different options (such as TCP vs. UDP) because the server is constant in its configuration**
- **Of course, you should question the validity of the absolute numbers. I believe they are optimistic simulations of non-disk bound servers.**
- **We are mostly looking at networking and protocol processing performance with this approach — server cached.**
- **I do not have sources for Solaris 2.5.1, my approach is black box mostly.**

Read results

Read Throughput of 20MB File in KB/s				
	UDP		TCP	
	100BaseT	FDDI	100BaseT	FDDI
NFS V2 8KB	6274	6263		
NFS V3 8KB	9499	9311	8048	6067
NFS V3 32KB	10629	11751	9093	6317

Notes:

1. reference fddi-new-r=5,w=8 and 100tx-new-synsw-hme2.5.1,hd,r=6,w=8

2. 11 samples, remove file between each write/read pair.

3. Used Sun Microsystems 100BaseT (hme) card, and Cisco CDDI cards and hubs.

Write results

Write Throughput of 20MB File in KB/s				
	UDP		TCP	
	100BaseT	FDDI	100BaseT	FDDI
NFS V2 8KB	9292	9723		
NFS V3 8KB	* 8863	10051	7527	7127
NFS V3 32KB	10387	11657	8543	8372

Notes:

1. reference fddi-new-r=5,w=8 and 100tx-new-synsw-hme2.5.1,hd,r=6,w=8

2. 11 samples, remove file between each write/read pair. simple average

3. * Had one low outlier, else would've expected similar to FDDI

Observations

- **UltraSPARC 1 levelled FDDI and 100BaseT results**
  - ☞ **On SuperSPARC 20's running Solaris 2 FDDI was lower performance than 100BaseT — inefficient CDDI driver implementation?**
- **Hot dang! A single client can exhaust a 100mb/s link on reading and writing!**
  - ☞ **As a bounds of what to expect, expect full bandwidth of your pipe.**

Observations continued

- **Is TCP as a transport always a lose for NFS?**
  - **Other measurements of a real server with data forced to come off disk showed TCP a win — but I wonder if there was an artifact in that test.**
  - **Customers have reported lower performance with TCP in naive benchmarking**
  - **But an argument can be made that outside a isolated benchmark network TCP should always win?**
  - **I have not even scratched the surface of tuning the client TCP attributes.**

Interim questions

- **Is TCP necessary? And if so, is performance (overhead reduction?) reachable of the UDP level?**
- **WebNFS and Version 4 are promoting TCP as the transport — are wide area issues of reliability in conflict with local area issue of performance?**
- **How do you position this to customers?**

Side comment: 10BaseT is dead, enter the '90's and start cranking on high speed networks.

Observations continued

❍ Client tunings in Solaris 2.5.1 affect read performance.

Read Throughput of 20MB File in KB/s				
	UDP		TCP	
	FDDI untuned	FDDI tuned	FDDI untuned	FDDI tuned
NFS V2 8KB	4706	6263		
NFS V3 8KB	6914	9311	4399	6067
NFS V3 32KB	9826	11751	6329	6317

Notes:

1. untuned defaults nfs_nra and nfs3_nra to "1", in tuning I changed to "5", increasing the read-ahead. Write performance in excess of read performance suggests poor read-ahead strategy or not aggressive enough.

2. Default behaviour favors 32KB transfer size -- is readahead number of "xfer" size units? For small xfers size, read-ahead should increase.

Conclusion

- **No wonder customers get confused — I'm confused.**
- **More characterization work on high speed links is needed.**
- **Investigation of TCP performance is needed.**
- **We need to look forward now to Gigabit speeds. Can NFS serve this area or do we need custom streaming protocols?**
- **Any changes going into Version 4?**

Please come by and talk to me if you think you have a better client, or have some data on throughput performance to share.