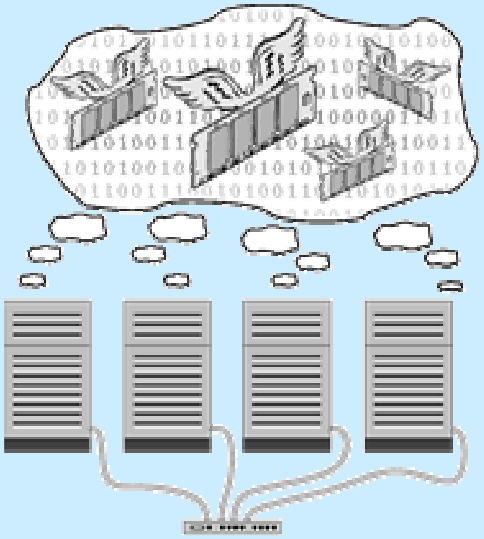
Nswap: A Network Swapping Module for Linux Clusters

Sean Finney, Kuzman Ganchev, Michael Spiegel, Matti Klock, Advisor: Tia Newhall — Swarthmore College

Network Swapping on Linux Clusters

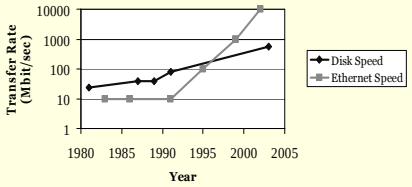
Cluster nodes use the remote idle memory of other cluster nodes as their swap device (rather than swapping to local disk)

- Robust system for sharing memory between cluster nodes
- Cluster memory shared over standard network with commodity computers
- Idle nodes donate free physical memory as surplus to nodes with overcommitted memory.



Why Network Swap?

Trends in Sustained Transfer Rates



Ethernet technology is currently outperforming disk technology by an order of magnitude (this trend will likely continue)

On average, two-thirds of the memory in a network of workstations is idle.

Goals

Transparency
- Users should not need to do anything special to use Nswap
- Non-invasive implementation
- **How**: implemented as a loadable kernel module

Speed
- runs entirely at kernel level
- minimum overhead in time and space for protocols and state data

Scalability
- Must scale well between small and large clusters
- **How**:
  - Point-to-Point design: no central server
  - Nodes make remote caching information using incomplete info.
    - keep partial cache of most available servers

Flexibility
- Must quickly adapt to changing needs of every node
- **How**: Each node grows/shrinks/migrates stored remote pages transparently in response to changes in its local memory use

Portability
- Should be architecture independent as possible
- Nswap runs on any architecture Linux supports

Implementation

Nswap Loadable Kernel Module on Linux

Developed on GNU/Linux as loadable kernel module with minimal kernel patching

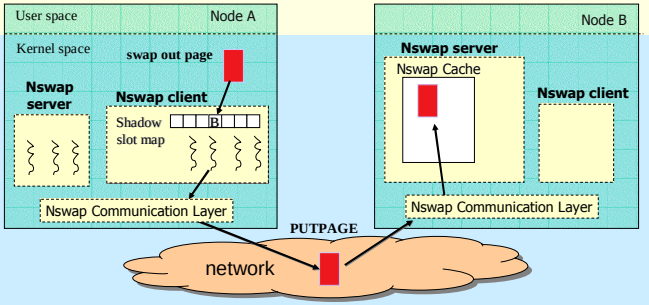
Can be dynamically added and removed from a running system

Runs entirely in kernel space

Each node runs:

Nswap client
- interacts with kernel as a standard swap partition
- keeps a cache of most available servers
- handles swap out requests: finds remote Nswap server and sends it the page
- handles swap in requests: looks up page's location and fetches it from remote server
- garbage collector thread for removing stale pages

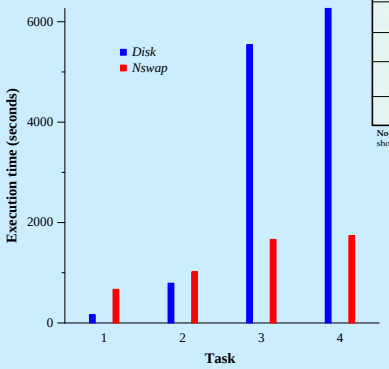
Nswap server
- manages stored remote pages of other nodes
- receives incoming page fetch and store requests
- grows/shrinks available memory based on local memory requirements
- migrates pages to other servers when resources low



Results

Environment:
Four Pentium III machines each with 512 MB RAM and with IBM Deskstar disk with a sustained data rate between 167 and 326 MBits/sec and a max rate of 494 Mbits/sec. The available interconnect was either 10 Mbit or 100 Mbit Ethernet. The bar graph shows results for four tasks with either sequential or random memory access patterns and with or without File I/O.

Results:
- **Nswap is faster than swapping to disk even when the network is slower than the disk for several workloads.**
- **On faster networks (1 and 10 Gbit) Nswap will be even faster.**



Speed up on faster networks

Task	Disk	10BaseT	100BaseT	1 Gbit	10Gbit
1	12.27	306.67	56.80	28.90	26.30
3	266.79	847.75	**153.54**	**77.30**	**70.30**
4	6265.39	9605.91	**1733.93**	**866.18**	**786.72**

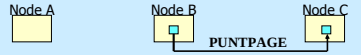
Note: Times for 1 Gbit and 10 Gbit networks are estimated using Amdahl's Law and measurements for 10 and 100 Mbit. Times shown are for a single iteration of the testing program.

Task	
1	Sequential no I/O
2	Sequential with I/O
3	Random no I/O
4	Random with I/O

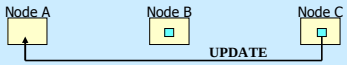
Tasks ran for four iterations, and file I/O was a separate process.

Page Migration Between Nswap Servers

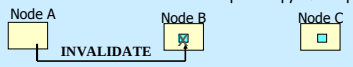
When Server B is full, it migrates A's page to server C



Server C tells A that it now has A's page



Client A tells Server B that it can drop its copy of A's page



Future Work

Reliability: Developing and Adding a Reliability Scheme

Adaptability: Developing best Nswap Cache growing and shrinking scheme
Adaptive policy based on workload:

Scalability: Testing on larger and faster clusters

Speed: Developing a reliable UDP layer for faster remote page transfers