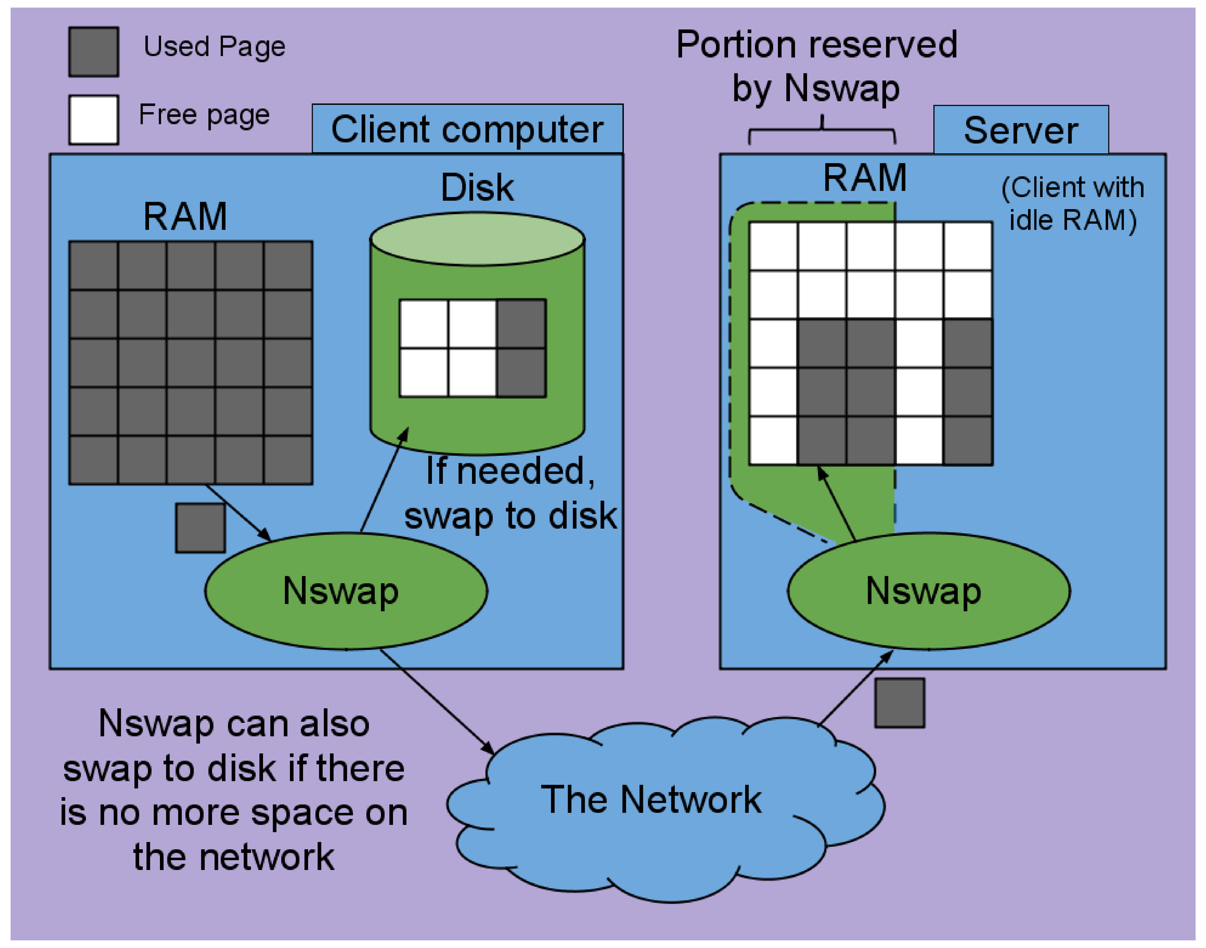
Speeding up computation with a filesystem of network RAM

Colin Schimmelfing '10, Advisor: Tia Newhall ~ Swarthmore College, Swarthmore PA

What is Nswap?



- When a computer runs out of RAM, it tries to 'swap' to disk
- 'Swapping' treats the disk as if it was RAM
- Problem: hard disks are one million times slower than RAM
- Storing data across the network in idle RAM is faster than disk
- Nswap allows a computer to 'borrow' RAM from other computers on the network
- Transparent to the program, handled at operating system level (Programmer does not need to do anything to use Nswap)

Adding an API to Nswap

- Application Programming Interface (API) allows programmer direct interaction with Nswap

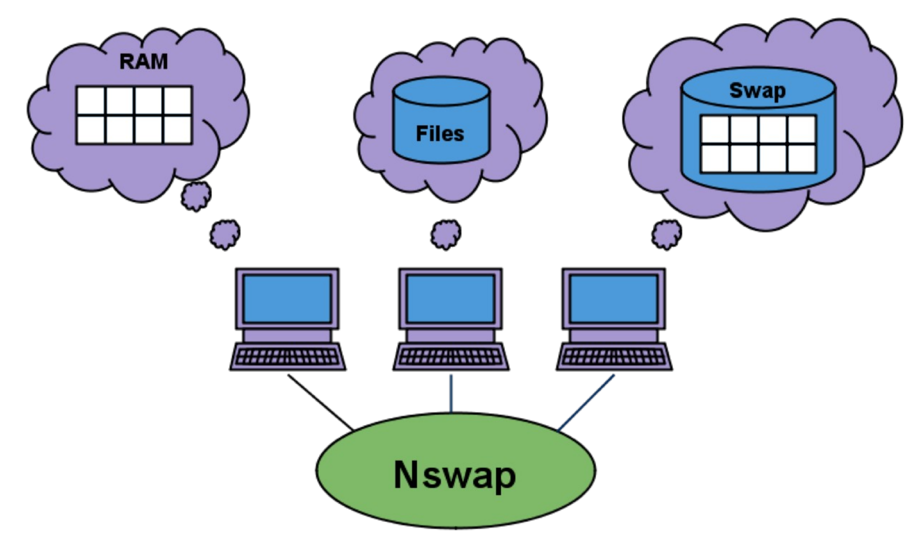
Why do we want this?

- More control to programmers can increase performance- they know the characteristics of their own data
- Specify which elements of memory could be farther away (on the network)
- Could provide greater resources to programs which try not to swap
- Currently programmers have no control; Nswap is transparent

What can an API do?

Nswap could be used as:

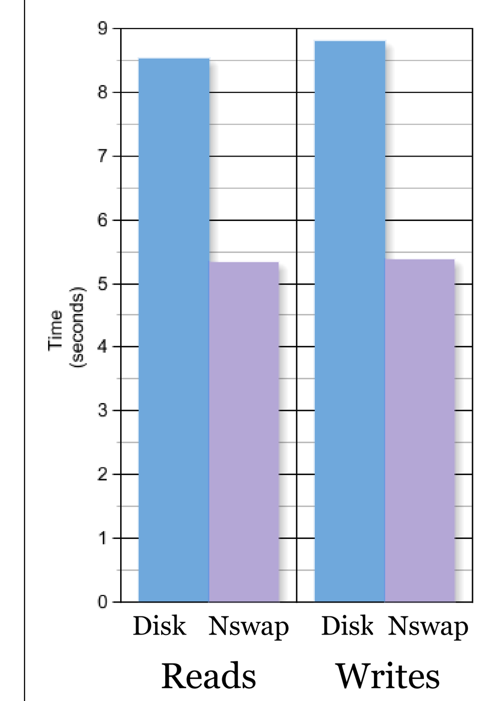
- A filesystem
- Fast swap device
- Large pool of RAM, visible to programs (malloc, mmap, etc.)



- Filesystem interface is useful for data that must be written and read repeatedly to and from the disk.
  - Instead, Nswap is the 'disk'- significantly faster access times
- The large pool of RAM interface is useful because some programs will limit their memory usage and will not use swap space
- Swap space interface is the original Nswap, computer can not distinguish between fast and slow swap devices

Results

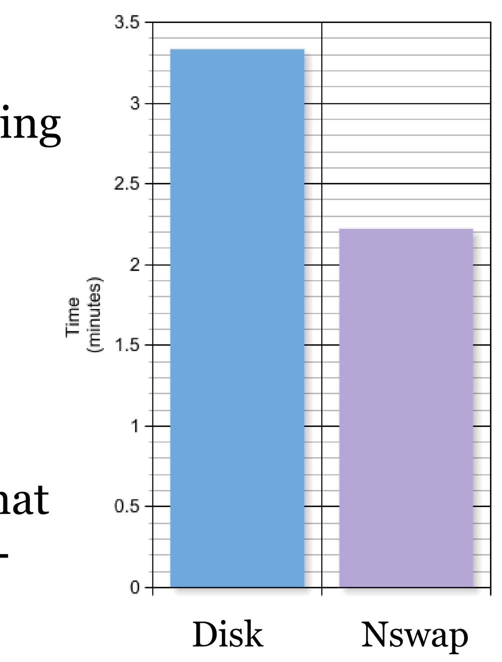
Simple Reads and Writes



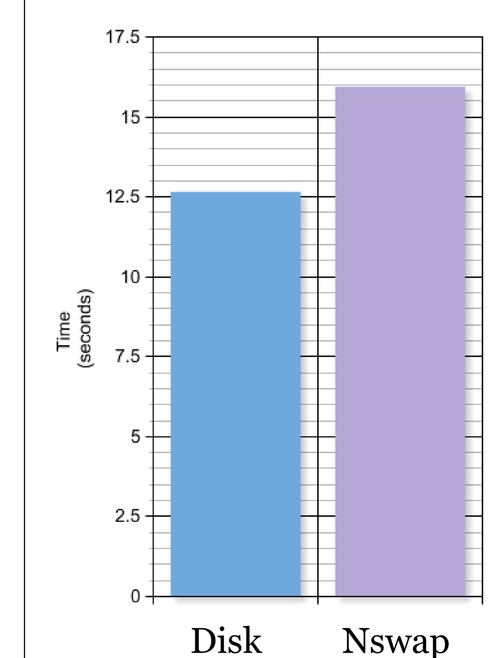
- 390.625 MB of data (100,000 pages at 4 KB each)
  - Simple, sequential reads and writes to a file
  - Nswap is about ⅓ faster than disk

External Merge Sort

- 1600 MB of data, sorting 100 MB at a time in memory
- Nswap is again about ⅓ faster than disk
- Large dataset shows that Nswap can scale to real-sized datasets



TPIE



- 300 MB of data with 30 MB of memory allowed for the merge sort
- The data is stored either on disk or on Nswap- The memory is not swapped- only local memory
- TPIE is about ¼ faster than Nswap!
- This seems impossible, since disk is slower even for large continuous writes, disk's best case

Implementation

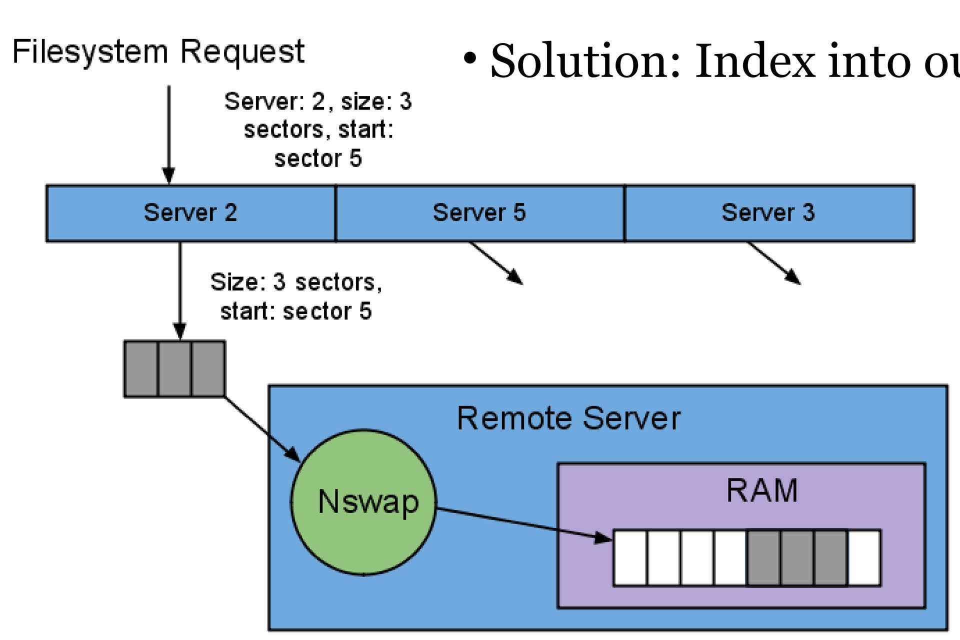
- Created a temporary filesystem that resides on the idle RAM of computers in the local network

Needs to support multiple modes:

Swapping mode, with page-size requests and garbage collection (GC)
- GC is needed to free up memory for future use from programs that have finished- RAM can be used again

Filesystem mode, with smaller requests and no GC
- Can't use GC- no applications claim files, so GC thinks that files can be collected

- One problem: we don't want to use too much memory to keep track of where our pages are.



- Solution: Index into our original page-size structures
- Data structures have page-size (4 KB) granularity
- Data sent in only as many smaller chunks (512 Bytes) as we need saves bandwidth

Experiments

Comparing filesystem performance between Nswap and disk

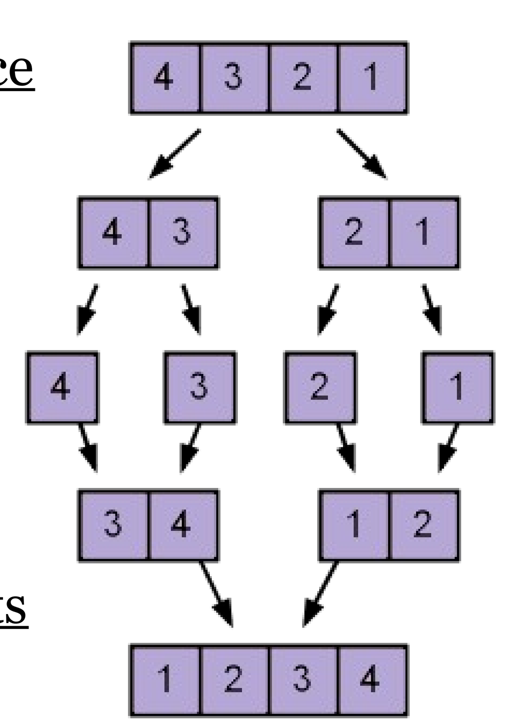
Experiment 1: Direct write to files

- Simple writes and reads to files on disk and files on our Nswap filesystem
- Should be best case for disk compared to Nswap

Experiment 2: External merge sort performance

- External merge sort allows sorting without all of the data in memory at a time
- As long as two of the smallest units can fit in memory, the sort can be completed
- Requires many reads and writes to disk

Merge Sort



Experiment 3: TPIE, a library for large data sets

- TPIE implements fast algorithms for data sets too large to fit in memory
- Optimized for disk, sequential reads and writes
- Ran the TPIE external merge sort algorithm

Future Work

Experiments

- Discover a fix to TPIE issue
- Try running with STXXL, another library for larger-than-memory datasets

New Interfaces

- Malloc-like interface to Nswap
- Other memory interfaces, such as mmap
- Provide an estimate of the network RAM available

Other Improvements

- Tune Nswap parameters to maximize filesystem performance
- Add flash memory capability
- Work on persistence for the filesystem