11/5/2023
Reading projects!

Some proposals

Λ8
Συστήματα & Λογισμικό Υψηλών Επιδόσεων

Contemporary locking

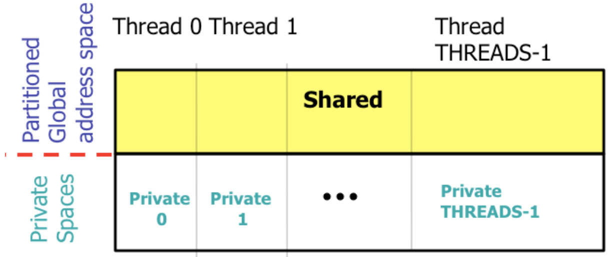
- Πρόσφατες τεχνικές για κλειδαριές (Lock cohorting, Compact NUMA-aware locks, Fissile locks)

Parallel File Systems and MPI I/O

- Many high-performance parallel applications, need to read or write large amounts of data from/to disks. For example, reading (huge) initial matrices or writing simulation results, or store checkpointing data for fault tolerance. While this is more pronounced in MPI-based parallel applications (see MPI-I/O), OpenMP applications also need high-performance I/O.
- A common scenario is to have a single process (or thread) read or write to the disk; all others have to retrieve/store data through this process. This however is obviously not peformant. High performance I/O must allow concurrent accesses to storage, which in turn requires *parallel file systems*. **Lustre** and **GPFS** are arguably the most prominent, popular ones.
- **Study, summarize and present the world of parallel file systems (design, organization, operation, taxonomy, etc) and specialize in Lustre.**

PGAS
Partitioned Global Address Space

- Global (shared) address space, but each “thread” knows that it owns a small portion of the space.
  - A collection of “threads” (processes) operating in a partitioned global address space that is logically distributed across threads.
  - Each thread has affinity with a portion of the globally shared address space. Each thread has also a private space.
  - Elements in the partitioned global space co-located with a thread are said to have affinity to that thread.



- Programmer has control over performance-critical factors—data distribution and locality control—computation partitioning—communication placement.
- A number of languages but **UPC** most prominent; **XScalableMP** another one; both are C extensions. There are many others (Java-based mostly).
- Start with:
  - M.D. Wael, S. Marr, B.D. Fraine, T.V. Cutsem and W.D. Meuter, “Partitioned Global Address Space Languages”, *ACM Comput. Surv.*, Vol. 47, No. 4, pp. 1–27, July 2015.

High-performance interconnects

- Large parallel systems (found in the top500 list, etc) are in essence really big clusters that rely on some kind of *fast interconnect*. Traditional commercial Ethernet-based networking is still the most popular networking solution but when it comes to supercomputer-grade performance (HPC), it is not cut for the job. Higher-performance 10G/25G/40G/100G ethernet may be promising but it is still not popular.
- The state-of-the art technologies in this field include Myrinet, Quadrics, Infiniband and more recently Slingshot; the last two seem to dominate the market: *Infiniband* (mostly backed by Mellanox—now NVidia) and *Omni-Path* (Intel), an InfiniBand-derived technology.
- Both interconnects already sport bandwidths of 200 Gb/sec; 400Gb/sec and 800Gb/sec are planned (but are dependent on the availability of PCI-Express 5.0 slots, since a NIC must connect to the local CPU as well…).
- To maintain very low latency, **Infiniband** and **Slingshot** perform a lot of processing on the *NICs* and the *switches* of the network.
- To enable huge network configurations, *high-radix switches* are employed (up to 800 ports!!). Using such switches, low-diameter topologies can be utilized. Instead of older meshes, tori, Clos's or fat-trees, topologies like ***Dragonfly*** *(2008)*, or ***Slimfly*** *(2014)* seem to attract interest because they reduce latency and power requirements.

- **Study, summarize and present the Dragonfly topology and how it is used in HPC.**

Non-blocking algorithms/data structures

- In plain words:
  - Blocking: uses locks to protect critical regions (a waiting thread is *blocked* until the lock is released; what if the thread that holds the lock dies?)
  - Non-blocking: no locks; failure of any thread cannot stop the system of progressing.
    - They employ atomic instructions (fetch-and-add. Compare-and-swap, etc)
    - Lock-free: guaranteed system-wide progress (usually c-a-s)
    - Wait-free: lock-free AND in addition, per-thread progress (no thread may starve) (usually f-a-a)

- Difficult to be general in algorithms
  - Non-blocking *data structures* usually
  - Linked lists, stacks, etc

Others

- Scheduling non-rectangular loops in OpenMP
- RDMA and MPI One-sided communications
- Infiniband and Slingshot interconnects