Memory Affinity in Multi-threading
The Bowtie2 Case Study

Claudia Misale, Marco Aldinucci
University of Torino, Italy

Massimo Torquati
University of Pisa, Italy

Memory bound problem
- Synchronization efficiency problem for very fine grain tasks (<1ms)
- Load balancing problem for coarse grain tasks
- Strongly lock-based due to shared data

Input Dataset as Shared Data
- Mutex access via spin-lock
- Indexed Genome
  - Reference Genome global to all threads

Seed Positions
AAAA 0,8
ACCT 4,16,24
GTGA 12,20
...

Match

W1 W2 Wn-1 Wn

GTGA
AAAA
TCCG
ACTC
GTTC

Bowtie2:
- a fast Alignment tool for Bioinformatics
- Locks removed
- Memory affinity for private data
- Tasks balanced with memory affinity scheduling policy
- Threads pinned on cores

Bowtie2-FF:
- Bowtie2 porting on top of FastFlow programming framework for multi-core platforms

Load Balancing
- Incoming and outcoming tasks are pushed in lock-less (CAS-free) FIFO buffers

Output file
- GTGA Pos: 12,20
- AAAA Pos: 0,8

Affinity Scheduling Policy
- Tasks are allocated in the master to match NUMA node of the selected worker

Pinned Threads
- Get tasks from the Master thread
- Do the alignment
- Send back results

Alignment against the Genome
- The Genome can be allocated interleaved on all NUMA nodes of the shared memory

3M reads, readlen 8 and 68 (SRR078586)

10M reads, readlen 20 (SRR072996)

FastFlow web site: http://mc-fastflow.sourceforge.net/