



BioBITs

Euromicro PDP 2010 - Pisa Italy - 17th Feb 2010

Efficient Smith-Waterman on multi-core with FastFlow



Computer Science Dept. - University of Torino - Italy

Marco Aldinucci

Massimo Torquati Computer Science Dept. - University of Pisa - Italy

Massimiliano Meneghin IBM Technology Campus, Dublin Software Lab, Ireland

http://mc-fastflow.sourceforge.net

Outline

Motivations FastFlow Performance Smith-Waterman benchmark Performance **Conclusions & Commercial**



ERCIM







[< 2004] Shared Font-Side Bus (Centralized Snooping)





[2005] Dual Independent Buses (Centralized Snooping)





[2007] Dedicated High-Speed Interconnects (Centralized Snooping)

This and next generation Multi-cores

- Are programmed at "concurrent assembler" level
 - Complex, not portable, not efficient

Fine-grained parallelism is off-limits

- Exploit cache coherence
- Memory fences are expensive
- Will worsen with core count



(oreGRID

BOBIS

ERCIM

I/O bound problems, High-throughput, Streaming, Irregular DP problems Automatic and assisted parallelisation

E.g. farm (a.k.a. master-workers)

- Model foreach loop and Divide&Conquer
- Exploit it as a high-order language construct
 - Why should we re-code it from scratch each application?
 - A C++ template factory exploiting highly optimised implementation

http://mc-fastflow.sourceforge.net

E.g. farm with POSIX lock/unlock

Wednesday, February 17, 2010

Lock-free and CAS-free (fence-free)

- **Single-Producer-Single-Consumer FIFO queues**
- Lamport et al. 1983 Trans. PLS (Sequential consistency only passive)
- Giacomoni et al. 2008 PPoPP (Relaxed consistencies (e.g. TSO) passive)
- Multiple-Producers-Multiple-Consumers FIFO queues
- with CAS (at least one) passive ... a plethora
- without CAS passive 🖙 Cannot be done
- without CAS active INF FastFlow

Wednesday, February 17, 2010

BioBITs

(oreGRID

ERCIM

FastFlow: A step forward

High-level programming

Implemented on top of lock-free/fence-free non-blocking synchronizations

C++ STL-like implementation

http://mc-fastflow.sourceforge.net

Coarse grain (50 µS workload)

http://mc-fastflow.sourceforge.net

Medium grain (5 µS workload)

Medium grain (5 µS workload)

Fine grain (0.5 µS workload)

-	_	_		_	_	_			_				
		G	Α	Α	Т	Т	С	Α	G	Т	Т	Α	GAATTCAG GAATTCAG
	0	0	0	0	0	0	0	0	0	0	0	0	
G	0	5	1	0	0	0	0	0	5	1	0	0	
G	0	5	2	0	0	0	0	0	5	2	0	0	GGA-TC-G GCAT-C-G
Α	0	1	10	7	3	0	0	5	1	2	0	5	
Т	0	0	6	7	12	8	4	1	2	6	8	4	GAATTC-A GAATTC-A
С	0	0	2	3	8	9	13	9	5	2	4	5	
G	0	5	1	0	4	5	9	10	14	10	6	2	
Α	0	1	10	6	2	1	5	14	10	11	7	11	GGA-TCGA GCAT-CGA

Smith-Waterman algorithm Local alignment - dynamic programming - O(nm)

Fast Smith-Waterman

E Smith-Waterman algorithm

- Local alignment
- Time and space demanding O(mn), often replaced by approximated BLAST
- Dynamic programming
- Real-world application
- It has been accelerated by using FPGA, GCPU (CUDA), SSE2/x86, IBM Cell
- Best software implementation up to now - SWPS3: evolution of Farrar's implementation
 - SSE3 + POSIX IPC

A matrix
$$H$$
 is built as follows:

$$\begin{split} H(i,0) &= 0, \ 0 \leq i \leq m \\ H(0,j) &= 0, \ 0 \leq j \leq n \\ H(i-1,j-1) + w(a_i,b_j) \\ H(i-1,j) + w(a_i,-) \\ H(i,j-1) + w(-,b_j) \\ \end{split} \\ \text{Where:} \\ \end{split}$$

$$\end{split}$$

- a, b = Strings over the Alphabet Σ
- m = length(a)
- n = length(b)
- H(i,j) is the maximum Similarity-Score between the substring of a of length i, and the substring of b of length j
- $w(c,d), \ c,d \in \Sigma \cup \{'-'\}$, '' is the gap-scoring scheme
- Substitution Matrix: describes the rate at which one character in a sequence changes to other character states over time
- Gap Penalty: describes the costs of gaps, possibly as function of gap length

Experiment parameters Affine Gap Penalty: 10-2k, 5-2k, ... Substitution Matrix: BLOSUM50

Smith-Waterman testbed

- Each query sequence (protein) is aligned against the whole protein DB
 - E.g. Compare unknown sequence against a DB of known sequences
- SWPS3 implementation exploits POSIX processes and pipes
- Faster than POSIX threads + locks
- http://people.inf.ethz.ch/sadam/swps3/

Smith Waterman (10-2k gap penalty)

ERCIM

OFFGRID

Smith Waterman (10-2k gap penalty)

ERCIM

OFFGRID

Smith Waterman (5-2k gap penalty)

ERCIM

Smith Waterman (5-2k gap penalty)

ERCIM

OFFGRID

s FastFlow easy to use? Yes

3

 $\mathbf{5}$

6

7

8

9

10

11

12

13

14

15

16

17 }

201 // Original code 21 **#define** N 1024 2 **#define** N 1024 long A[N][N], B[N][N], C[N][N];1 23 int main() { 4 int main() { 24// < init A, B, C >2526for(int i=0;i<N;++i) { (2)27**for**(**int** j=0;j<N;++j) { 2829int $_C=0;$ for(int k=0;k<N;++k)3031 $_{-C} += A[i][k] * B[k][j];$ 32C[i][j] = C;33Г(4) |Г(5)-2 343536 374 3839405 4142// Includes 43**struct** task_t { 4445**int** i; **int** j;}; 464748495051int $_{-}C=0;$ 52538 5455delete t; 5657

// FastFlow accelerated code 22 **long** A[N][N], B[N][N], C[N][N];// < init A, B, C > $ff :: ff_farm <> farm(true /* accel */);$ std::vector<ff::ff_node *> w; **for**(**int** i=0;i<PAR_DEGREE;++i) w.push_back(**new** Worker); farm.add_workers(w); farm.run_then_freeze(); for (int i=0;i<N;i++) { for(int j=0; j<N;++j) { $task_t * task = new task_t(i,j);$ farm.offload(task); farm.offload((void *)ff::FF_EOS); farm.wait(); // Here join $task_t(int i, int j):i(i), j(j) \{\}$ **class** Worker: **public** ff::ff_node { **public**: // Offload target service void * svc(void *task) { $task_t * t = (task_t *)task;$ for(int k=0;k<N;++k) $_{-C} += A[t->i][k]*B[k][t->j];$ C[t->i][t->j] = C;return GO_ON; } 5859 };

Conclusions

- FastFlow efficiently supports streaming applications on commodity SCM (e.g. Intel core architecture)
 - More efficiently than POSIX (standard or CAS lock), Cilk, OpenMP, TBB
- **Smith Waterman algorithm with FastFlow**
- Obtained from SWPS3 by syntactically substituting read and write on POSIX pipes with fastflow push and FastFlow pop an push
 - In turn, POSIX pipes are faster than POSIX threads + locks in this case
 - Scores twice the speed of best known parallel implementation (SWPS3) on the same hardware (Intel 2 x Quad-core 2.5 GHz)

DON'T BELIEVE?

CHECK IT OUT!

QUESTION

Started in Nov '09 at today over 600 downloads

792

< > + dhttp://www.reddit.com/r/cpp/ C Q+ Coople III Gatzy (259) * Moodle DI * Rumors (203) * News (2551) * TBI man Diz* FastFlow++* Bookmark on Delicious HES RECORDER BUNKY POLITICS WIT ADRESDIT SCIENCE CARING WORLDATING RECORDERS ATHEIRS TECHNOLOGY LANA VIDEOS CI MORE-Presit or what's hot new controversial top saved many in part register in seconds I Erglish std::cout magic (huntmann search reddit admitted 5 days ago by jostnetion 8 comments share LCA: Static analysis with GCC plugins (ten.net) remember me recover passwort (login) submitted 9 days ago by last_useful_man 3 comments_share LLVM+CLang successfully self-hosts (step.lvm.org) cpp submitted 11 days ago by last_weeful_m + frontpage 3,872 readers Boost release 1.42.0 available (http://www.org) Discussions and articles about the C++ programming language or programming in C++. sub-Witted 13 days equilibres 3 comments share a conveniently for 1 wea Order I Say! [C++-Next] (con-next.com) MODERATORS 2.2. submitted 14 days ago by the frits _____ comment share Andreas8Wagner gist. * Xerces-C++ 3.1.0 released (minutes) 12 submitted 14 days ago by get Submit a link /r/cpp: Neonate C++ programmer here, any advice to anything interesting: news article, blog entry, video, picture... that you wish you had had when you got started? submitted 18 days ego by BloodyThem \$3 comments share Create your own reddit FastFlow -- a skeletal multi-core programming framework [C++ template library] (conversed, unique) because this room is full of wini for people who know that Han shot first. submitted 19 days ago by me 4 comments share

In the top ten of reddit.com/c++ from several weeks

http://sourceforge.net/projects/mc-fastflow/

Wednesday, February 17, 2010

Visite