High-performance computing and symbolic computation

Marc Moreno Maza

University of Western Ontario, Canada

August 9, 2016
High-performance computing and symbolic computation

Research themes

- **Symbolic computation**: computing exact solutions of algebraic problems on computers with applications to mathematical sciences and engineering.
- **High-performance computing**: making best use of modern computer architectures, in particular hardware accelerators (multi-core processors, graphics processing units).

Research projects

- The *RegularChains* library: solving systems of algebraic equations and integrated into the computer algebra system *Maple*.
- The *Basic Polynomial Algebra Subroutines* (BPAS) and *CUDA Modular Polynomial* (CUMODP) libraries: hardware accelerator support for symbolic computation.
- The *Meta_Fork* library: a programming environment for hardware accelerators; this project is supported by IBM Canada.
Lab members and courses

Current students

Visiting: Mahsa Kazemi, Ruijuan Jing
PhD: Parisa Alvandi, Ning Xie, Xiaohui Chen, Steven Thornton, Robert Moir, Egor Chesakov
MSc: Masoud Ataei, Yiming Guan, Davood Mohajerani
Undergrad: Amha Tsegaye.

Alumni

Moshin Ali (ANU, Australia) Jinlong Cai (Microsoft, USA), Changbo Chen (Chinese Acad. of Sc.), Svyatoslav Covano (U. Lorraine, France) Akpodigha Filatei (Guaranty Turnkey Systems Ltd, Nigeria) Sardar A. Haque (GeoMechanica, Canada) Zunaid Haque (IBM Canada) François Lemaire (U. Lille 1, France) Farnam Mansouri (Microsoft, Canada) Liyun Li (Banque de Montréal, Canada) Xin Li (U. Carlos III, Spain) Wei Pan (Intel Corp., USA) Sushek Shekar (Ciena, Canada) Paul Vrbik (U. Newcastle, Australia) Yuzhen Xie (Critical Outcome Technologies, Canada) Li Zhang (IBM Canada) . . .

Courses

CS 2101 Foundations of Programming for High Performance Computing
CS 3101 Theory and Practice of High-performance Computing
CS 3350 Computer Architecture
CS 9535/4402 Distributed and Parallel Systems
Solving polynomial systems symbolically

\[ R := \text{PolynomialRing}([x, y, z]); F := [5^2 x^2 + 2^2 x^2 z^2 + 5^1 y^4 + 15^1 y^5 - 15^1 y^6 - 5^1 y^3]; \]

\[ \text{polynomial\_ring} \]

\[ [5 x^2 + 2 x z^2 + 5 y^6 + 15 y^4 + 5 z^2 - 15 y^5 - 5 y^3] \]

\[ \text{RealTriangularize}(F, R, \text{output} = \text{record}); \]

\[
\begin{align*}
5 x^2 + 2 z^2 x + 5 y^6 + 15 y^4 - 5 y^3 - 15 y^5 + 5 z^2 = 0 \\
25 y^6 - 75 y^5 + 75 y^4 - z^4 - 25 y^3 + 25 z^2 < 0 \\
5 x + z^2 = 0 \\
25 y^6 - 75 y^5 + 75 y^4 - z^4 + 25 z^2 = 0 \\
64 z^4 - 1600 z^2 + 25 > 0 \\
z \neq 0 \\
z - 5 \neq 0 \\
z + 5 \neq 0 \\
x + 5 = 0 \\
y = 0 \\
z - 5 = 0 \\
x + 5 = 0 \\
y - 1 = 0 \\
z + 5 = 0 \\
5 x + z^2 = 0 \\
2 y - 1 = 0 \\
64 z^4 - 1600 z^2 + 25 = 0
\end{align*}
\]

Figure: The RegularChains solver designed in our UWO lab can compute the real solutions of any polynomial system exactly.
Our polynomial system solver is at the core of **Maple**

**Figure:** Maplesoft, the company developing **Maple**, demonstrates the *RegularChains* solver designed in our UWO lab, in order to advertise **Maple**.
Let $K$ be the maximum number of thread blocks along an anti-chain of the thread-block DAG representing the program $P$. Then the running time $T_P$ of the program $P$ satisfies:

$$T_P \leq \left( \frac{N(P)}{K} + L(P) \right) C(P),$$

where $C(P)$ is the maximum running time of local operations by a thread among all the thread-blocks, $N(P)$ is the number of thread-blocks and $L(P)$ is the span of $P$. 

Our UWO lab develops mathematical models to make efficient use of hardware acceleration technology, such as GPUs and multi-core processors.
Our lab develops a compilation platform for translating parallel programs from one language to another; above we translate from OpenMP to CilkPlus through MetaFork. This project is supported by IBM.
High-performance computing: automatic parallelization

Serial dense univariate polynomial multiplication

\[
\text{for}(i=0; \ i<=n; \ i++)\{
    \text{for}(j=0; \ j<=n; \ j++)
        c[i+j] += a[i] \times b[j];
\}
\]

GPU-like multi-threaded dense univariate polynomial multiplication

\[
\text{meta}_\text{for} \ (b=0; \ b<= \frac{2n}{B}; \ b++) \{
    \text{for} \ (u=0; \ u<=\text{min}(B-1, \ 2n - B \times b); \ u++) \{
        p = b \times B + u;
        \text{for} \ (t=\text{max}(0,\ n-p); \ t<=\text{min}(n,\ 2n-p) \ ; \ t++)
            c[p] = c[p] + a[t+p-n] \times b[n-t];
    \}
\}
\]

We use symbolic computation to automatically translate serial programs to GPU-like programs.
Application to mathematical sciences and engineering

Figure: Toyota engineers use our software to design control systems
Research projects with publicly available software

**BPAS**

Basic Polynomial Algebra Subprograms

www.bpaslib.org

**MetaFork**

www.metafork.org

**CUMODP**

DA Polynomial

www.cumodp.org

**REPO**

www.regularchains.org