XKaapi

- Quick user's guide -

installation, running and basic informations
about XKaapi

21-02-2011
Thierry Gautier, INRIA

Objectives

- XKaapi is an open source library for parallel computing
  - ✓ http://kaapi.gforge.inria.fr
- This tutorial is part of the following tutorials
  - ✓ XKaapi: programming with data flow graph
  - ✓ XKaapi: Low Level Adaptive Application Interface
  - ✓ XKaapi: KaSTL API
  - ✓ XKaapi: Fortran Interface
  - ✓ XKaapi: internal representation & execution
- soon available on the web site

What is Kaapi ?

- C/C++ Library for parallel programming
  - ✓ Target architecture: multicore + GPU + cluster
- Ultimate goal
  - ✓ Simplify the development of parallel application
    - architecture abstraction
  - ✓ Automatic dynamic load balancing
    - theoretically & practically performances
    - **Work Stealing based algorithms**

Design

- Kernel
  - ✓ runtime for API (or compiler)
  - ✓ work stealing internal scheduling
  - ✓ C language, fine grain implementation...
- APIs for different programming models
  - ✓ Data Flow Graph: **DFG**
    - Athapascan (deprecated), Kaapi++
  - ✓ Parallel STL like: **KaSTL**
  - ✓ Adaptive Algorithms Interface: **AAI**
  - ✓ Fortran Interface: **KaFOR**

Quick user’s guide

Source development

- http://kaapi.gforge.inria.fr
  - ✓ tarball of the master (rcxx)
- GIT: ligforge
  - ✓ url = ssh://git.ligforge.imag.fr/git/kaapi/xkaapi.git
- Usage of branches
  - ✓ origin/master: the official master branch
  - ✓ origin/<username>/<branch name>: an user branch
    - The owner is responsible of its branches
    - The user **MUST ONLY** commit on its own branches
- Mailing list:
    - http://lists.gforge.inria.fr/cgi-bin/mailman/listinfo/kaapi-leaders

Installation

1. automake / autotools etc...
   - ✓ ../xkaapi/configure --help
   - ✓ ../xkaapi/configure --prefix=<totodir>
   - ✓ Usefull options:
     - --enable-mode=release for performances
     - --enable-mode=debug for more assertions in the user level API
     - --enable-static: enable static library (default no)
     - --enable-shared: enable shared library (default yes)
2. Compilation
   - ✓ make
3. Basic check
   - ✓ make check

Installation (cont.)

4. Installation
   - ✓ make install (in the --prefix directory, see step 1)
   - ✓ <totodir>/include; <totodir>/lib etc..
     - use pkgconfig to retreive CCFLAGS, LDFLAGS etc
   - ✓ <totodir>/shared/doc/xkaapi/examples
     - all examples are put here + makefile to compile them
   - ✓ <totodir>/shared/doc/slides/
     - several subjects are presented as slides (pdf)
       - quick user's guide: this presentation
       - dfg programming: how to program with macro data flow graph
       - ...

List of examples

- Examples sub directory
  - ✓ cd examples;
  - ✓ make examples : build all examples
  - ✓ make <prog>, e.g. make for_each_rec_xx
- Have a look of subdirectories in <topsrcdir>/examples !

- hello
  - ✓ hello_world.cpp

- for_each
  - ✓ for_each_rec_kaapi++.cpp : recursive C++ version
  - ✓ for_each_0_kaapi++.cpp : basic adaptive C++ version
  - ✓ for_each_0_kaapi.c : basic adaptive C version
  - ✓ for_each_1_kaapi++.cpp : adaptive C++ version, enable steal of thief
  - ✓ for_each_2_kaapi++.cpp : idem + preemption
  - ✓ for_each_0_kaapi++_lambda.cpp : adaptive C++ version with lambda
  - ✓ for_each_kastl.cpp: call to STL kastl implementation

Cont.

- Fibo
  - ✓ fibo_kaapi.c : low level C version
  - ✓ fibo_atha.cpp : old Athapascan C++ API version
  - ✓ fibo_kaapi++.cpp : Kaapi C++ API version
  - ✓ fibo_kaapi++_autopointer.cpp : Kaapi C++ API with autopointers
  - ✓ fibo_kaapi++_autovar.cpp: Kaapi C++ API with Kaapi’s automatic variables
  - ✓ fibo_kaapi++_opt.cpp: simple optimization to avoid one recursive spawn
  - ✓ fibo_kaapi++_opt_thread.cpp: optimization to avoid lookup of current thread
  - ✓ fibo_kaapi++_cumul.cpp : Kaapi C++ API version with cumulative write
  - ✓ fibo_kaapixx_cumul_opt.cpp : Kaapi C++ API version with optimized task creation
  - ✓ fibo_kaapi++_sync.cpp: usage of synch. to avoid sum’ task creation
  - ✓ > make fibo_kaapi fibo_kaapi++ fibo_atha...
- NQueens
  - ✓ nqueens_atha / nqueens_kaapi++
- Cilk
  - ✓ two examples from Cilk distribution (matrix computation/qsort)

Cont.

- Matrix
  - ✓ basic example to illustrate use of 2D range
  - ✓ matrix product
    - matprod_rec_kaapi++.cpp : recursive matrix product
    - matprod_iter_kaapi++.cpp: classic nested loops
  - ✓ LU factorization
    - matlu_kaapi++.cpp
- BFS
  - ✓ breadth first search
- ListRanking
  - ✓ Jaja & Helman based list ranking algorithm
- Poisson3D
  - ✓ stencil computation on 3D Poisson problem

Compilation of examples

- Use pkg-config

```
gautier@idkoiff:~$ export PKG_CONFIG_PATH=<kaapi
install dir>/lib/pkgconfig

gautier@idkoiff:~$ pkg-config --cflags kaapi++
-I/home/gautier/KAAPI/install/xkaapi/include

gautier@idkoiff:~$ pkg-config --libs kaapi++
-L/home/gautier/KAAPI/install/xkaapi/lib -lkaapi++ -
lkaapi
```

Typical use:

```
gautier@idkoiff:~$ g++ -o mytest mytest.cpp `pkg-config
--cflags kaapi++` `pkg-config --libs kaapi++`
```

That all !

Running example

- KAAPI_CPUCOUNT=1 ./fibo_kaapi++ 30

```
Fibo(30)=832040
Time: 4.326541e-01
```

- KAAPI_CPUCOUNT=2 ./fibo_kaapi++ 30

```
Fibo(30)=832040
Time: 2.143562e-01
```

- KAAPI_CPUSET=0:4,6 ./fibo_kaapi++ 30
  - use cores 0,1,2,3,4 and 6 of the machine

Sources organization

- xkaapi/src
  - ✓ everything about workstealing / graph representation is here
- xkaapi/examples
  - ✓ user level examples
- xkaapi/api
  - ✓ Athapascan C++ interface [deprecated]
  - ✓ Kaapi C++ interface
  - ✓ [Fortran interface] etc..

http://kaapi.gforge.inria.fr

Kaapi is a software developped at
http://moais.imag.fr