

# **Workshop on Computer Programming and Advanced Tools for Scientific Research Work & Quantum ESPRESSO Developer Training**

Project Presentation:

## **Optimization of MD OpenCL legacy code for CPUs and GPUs**

Group Members:

Oscar Najera, Ajasja Ljubetic, Rodrigo Neumann, Moses Sokunbi,  
Johannes Voss

Group Leader:

Ivan Girotto

# Overview

- Creating a collaborative platform
  - Source code management platform
  - Task assignment
  - Communication
  - Testing
- Optimization of MD OpenCL legacy code
  - Optimization of data transfer
  - Streamlining of OpenCL optimization
  - Scaling with the number of spawned threads
  - Testing: Implementation
  - Ongoing work

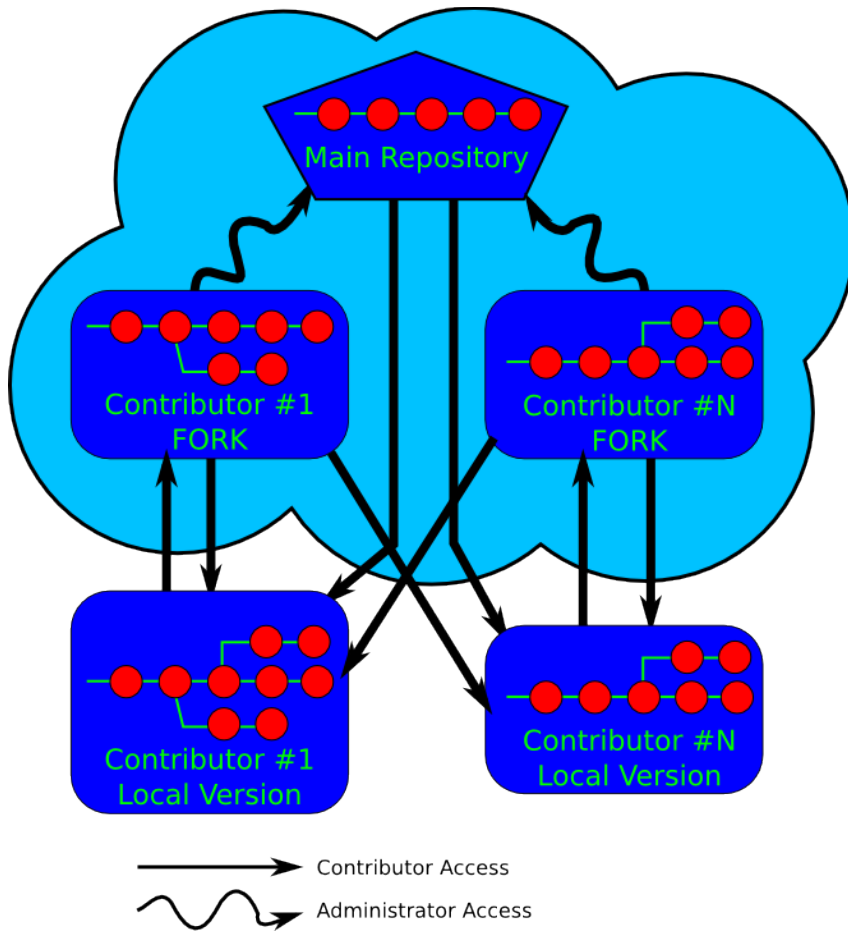
# Introduction

## OpenCL: Open Computing Language

- A framework for writing programs that execute across heterogeneous platforms (e.g. CPUs and GPUs).
- Provides parallel computing using task-based and data-based parallelism.

**Aim/Objective:** The aim of this project is to create an enabling environment for collaboration and to improve the OpenCL legacy code for CPUs and GPUs.

# Source code management platform for vision control



LJMD-OpenCL / ljmd-openc1

Pull Request Unwatch Star 0 Fork 6

Code Network Pull Requests 2 Issues 7 Wiki Graphs Settings

Collaborative development of OpenCL version of LJMD by Axel K. — Read more

ZIP HTTP SSH Git Read-Only https://github.com/LJMD-OpenCL/ljmd-openc1.git Read+Write access

branch: master Files Commits Branches 1 Tags

ljmd-openc1 / 105 commits

Merge pull request #25 from neumannr/master

Titan-C authored 4 hours ago latest commit 18b111946e

MAKFILES	18 hours ago	Preparing for merge with upstream [neumannr]
examples	3 days ago	Added example directory [igirotto]
include	a day ago	Merge remote-tracking branch 'upstream/master' [msokunbi]
references	3 days ago	Repository refresh [igirotto]
src	4 hours ago	Included comments regarding the _UNBLOCK [neumannr]
test	18 hours ago	Preparing for merge with upstream [neumannr]
.gitignore	2 days ago	Undoing some edits in order to synchronize [neumannr]
AUTHORS.md	19 hours ago	Preparing for merge with upstream/master [neumannr]
README.md	a day ago	Readme formatting for command line instructions [Titan-C]
TODO	3 days ago	Modified TODO list file [igirotto]
makefile	18 hours ago	Preparing for merge with upstream [neumannr]

README.md

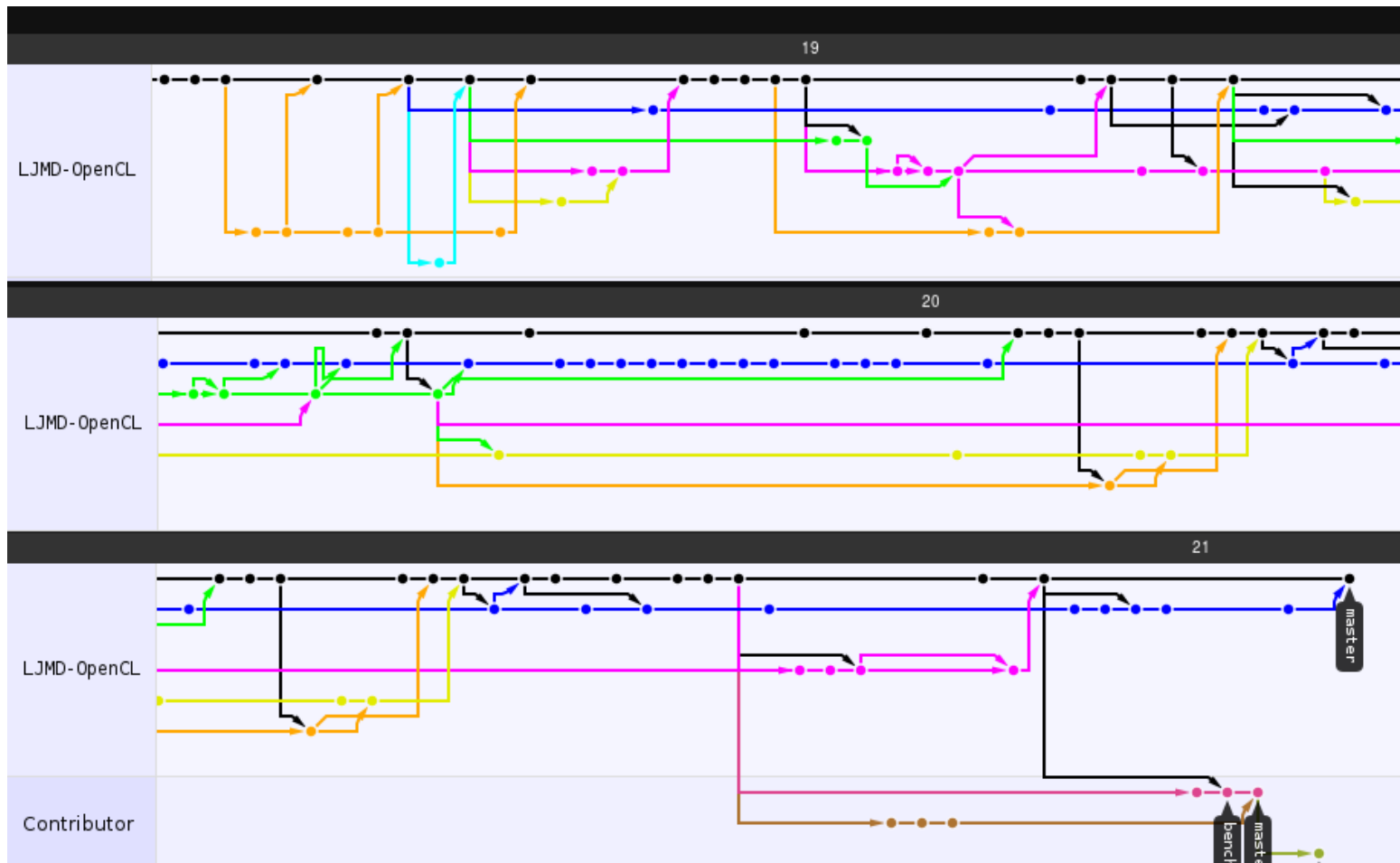
## LJMD-OpenCL

Repository for ICTP School-Lab. Collaborative development of OpenCL version of LJMD by Axel K.

### Install our software

<https://github.com/LJMD-OpenCL/ljmd-openc1.git>

# Source code management platform for vision control (Contd.)



# Task assignment and communication

- Mailing list (ljmd-opencl-project@ictp.it)
- Personal meetings
- Trello

The image shows a Trello board with four columns: Proposals (brainstroming), To Do, Doing, and Done. Each column contains task cards with various icons and user avatars.

- Proposals (brainstroming)**
  - Include a library for option passing (edit icon)
  - Visualize trajectory (comment icon 1)
  - Add a card...
- To Do**
  - Doxygen documentation for OpenCL\_utils
  - Add a card...
- Doing**
  - Work on multiple gpu support. (JV)
  - Measure the performance in as a function of the number of threads (1 vote, edit icon, star icon)
  - Benchmark blocking / non blocking performance on ARGO (RN)
- Done**
  - Platform detection (cpu-gpu). (comment icon 1, star icon)
  - automatic test to compare outputs of the original to the openCL version (notification icon 4, comment icon 1, eye icon, user avatars)
  - Make github repository (user avatars)
  - File-based ...

# Testing

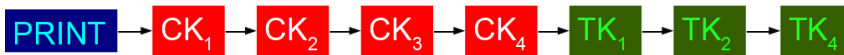
- Python framework for testing
- Carryout tests on the codes
  - on local version (Contributor)
  - on main repository (by administrator)
- Check that results match

# Optimization of data transfer

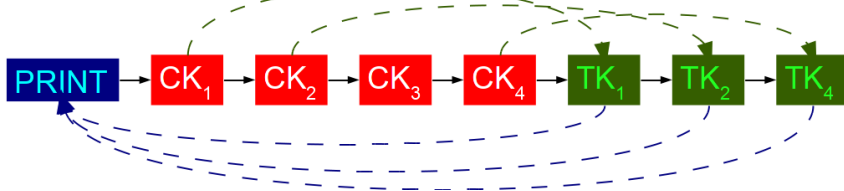
## Synchronous (blocking) Data Transfer



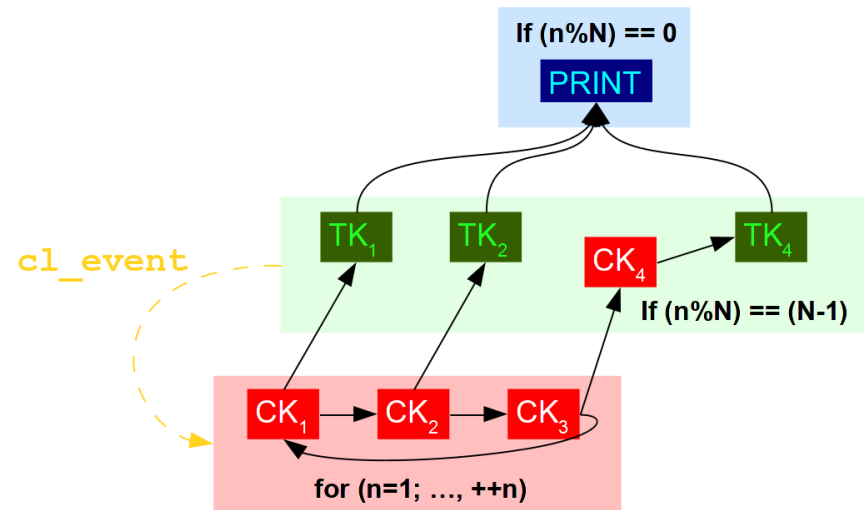
...breaking into pieces...



...identifying dependencies...



## Asynchronous (non blocking) Data Transfer





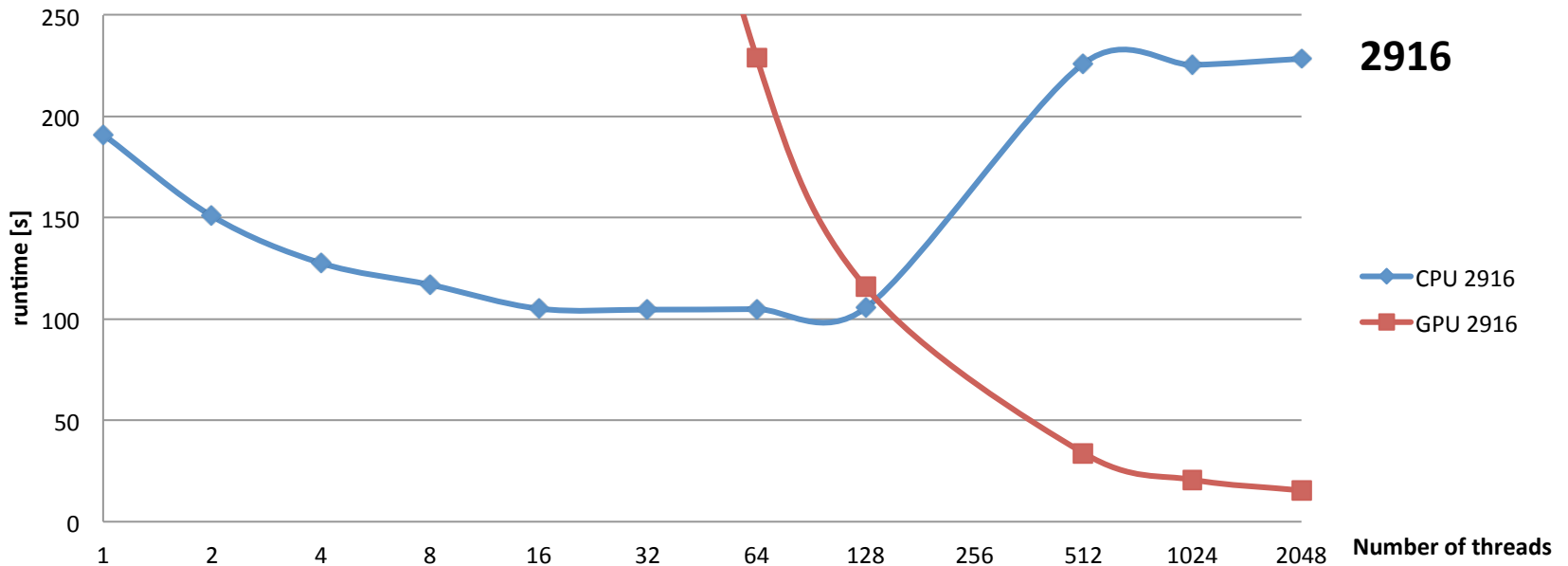
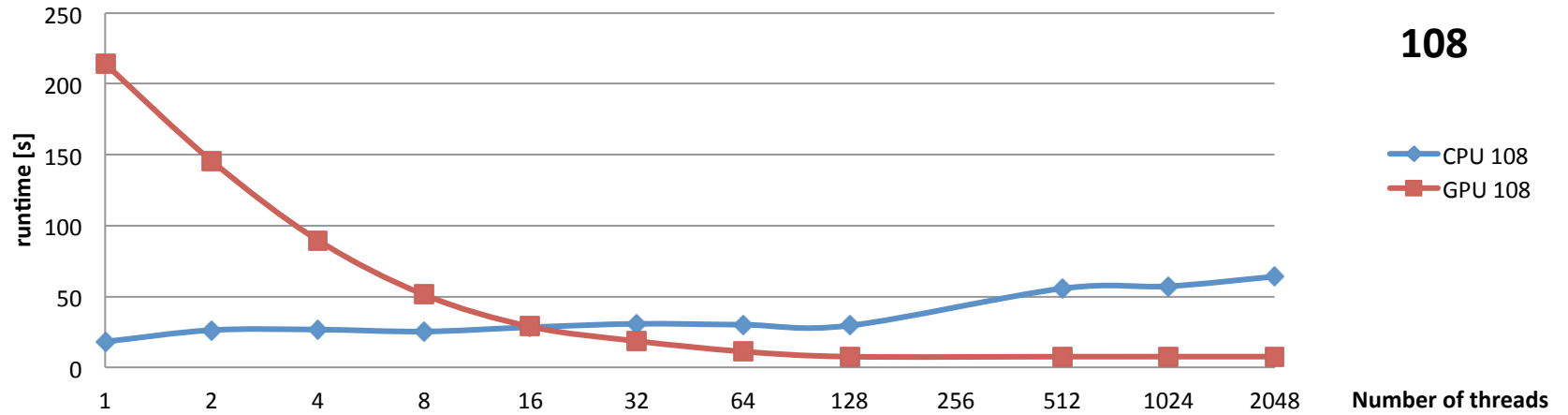
# Streamlining of OpenCL optimization

```
$(INC_DIR)/opencl_kernels_as_string.h: $(SRC_DIR)/opencl_kernels.cl  
awk '{print "\""$$0"\n\""}' <$> >$@
```

**Parsing with AWK for including kernels into the binary**

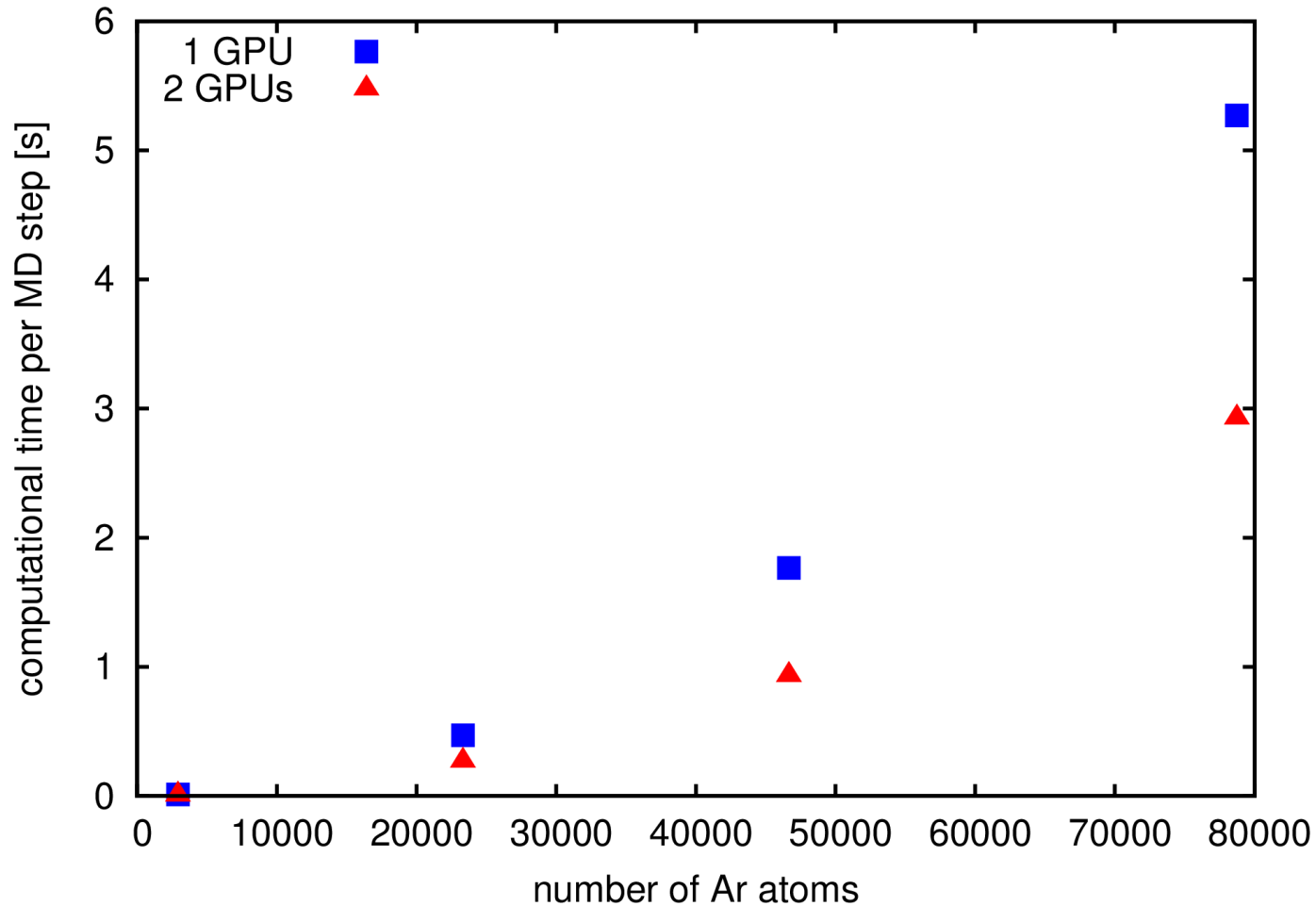
```
--uu-:---F1 makefile      68% L45   (BSDmakefile)-----  
globalWorkSize[0] = NTHREADS;  
  
char * sourcecode = source2string( "opencl_kernels.cl" );  
  
cl_program program = clCreateProgramWithSource( context, 1, (const char **) &sourcecode, NULL, &status );  
  
status |= clBuildProgram( program, 0, NULL, kernelflags, NULL, NULL );  
  
#ifdef __DEBUG  
--uu-:---F1 ljmd-cl.c      40% L205   (C/l Abbrev)-----  
char * source2string( char * filename ){  
  
    char line_buffer[STRINGSIZE];  
    char * string_buffer = NULL, * tmp = NULL;  
  
    size_t buffer_size = 0;  
    FILE * fp_file;  
  
    fp_file = fopen( filename, "r" );  
    if( !fp_file ){  
        fprintf( stderr, "Unable to open the source file %s. Program will be ended", filename );  
        exit(1);  
    }  
  
    while( fgets( line_buffer, sizeof( line_buffer ), fp_file ) != NULL ){
```

# Scaling with the number of spawned threads





# Ongoing work: Processing on multiple GPUs



# Conclusions

From our results we can conclude that we have successfully implemented an enabling collaborative environment and also developed an improved and optimized version of the MD OpenCL legacy code for CPUs and GPUs.

Thank you