



Yu-Shiang Lin

Software Engineer, Software Developer, Research Engineer

<https://coldfunction.github.io/>

Education

- National Tsing Hua University
 - Ph.D in Department of Computer Science (Jul. 2012 – Jan. 2018)
 - Doctoral Dissertation: A New API Remoting Policy for General-Purpose Computing on GPU Virtualization and its Application on Biological Tools
 - Chang Guang University
 - Master in Computer Science & Information Engineering (Aug. 2010 – July. 2012)
 - Masters Dissertation: Efficient Constrained Multiple Sequence Alignment Tool and Parallel RSA Decryption Algorithm for Many-core GPUs with CUDA

Work Experience

Oct 2018 - Mar 2021 (3 years)

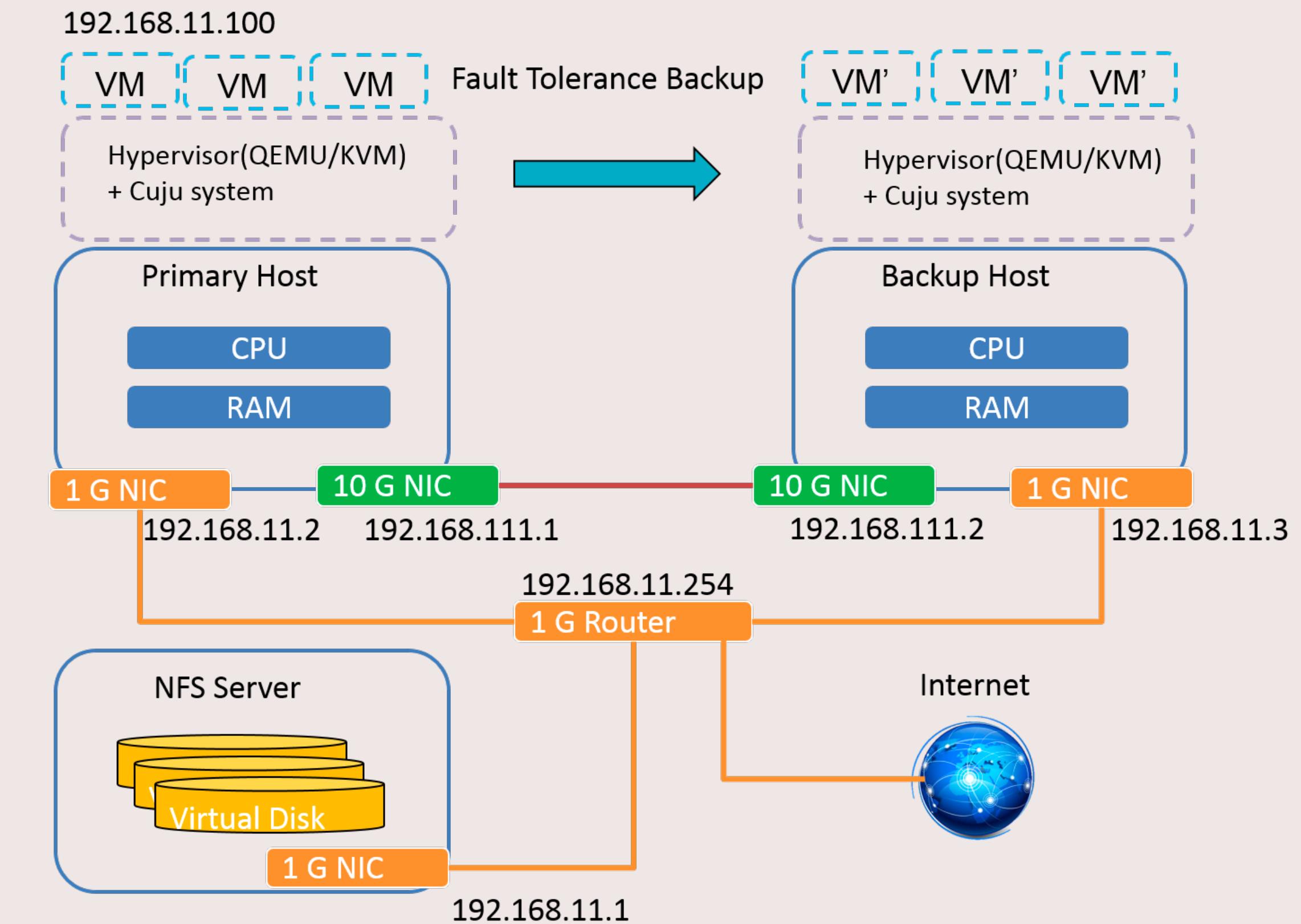
- Industrial Technology Research Institute
 - Research and Development Engineer
 - Virtualization-based fault tolerance
 - Cuju Project
 - Patents: FAULT-TOLERANT SYSTEM AND CONTROL METHOD THEREOF, P52080043TW, P52080043US & P52080043CN
 - Paper: 3 international conferences
 - Outstanding Research Award, Industrial Technology Research Institute, 2020



Cuju

Oct 2018 - Mar 2021 (3 years)

- Cuju: An open source project for Virtualization-Based Fault Tolerance
- Latency Bounding
- Group Fault Tolerance
- <https://github.com/Cuju-ft/Cuju>



< 10 ms

Reduce the fault tolerance latency to less than 10 milliseconds

> 90 %

90% or more epochs can be bounded on 2% error for 10 milliseconds latency

- 88%

Group FT can reduce 88% system latency of the Individual FT

Work Experience

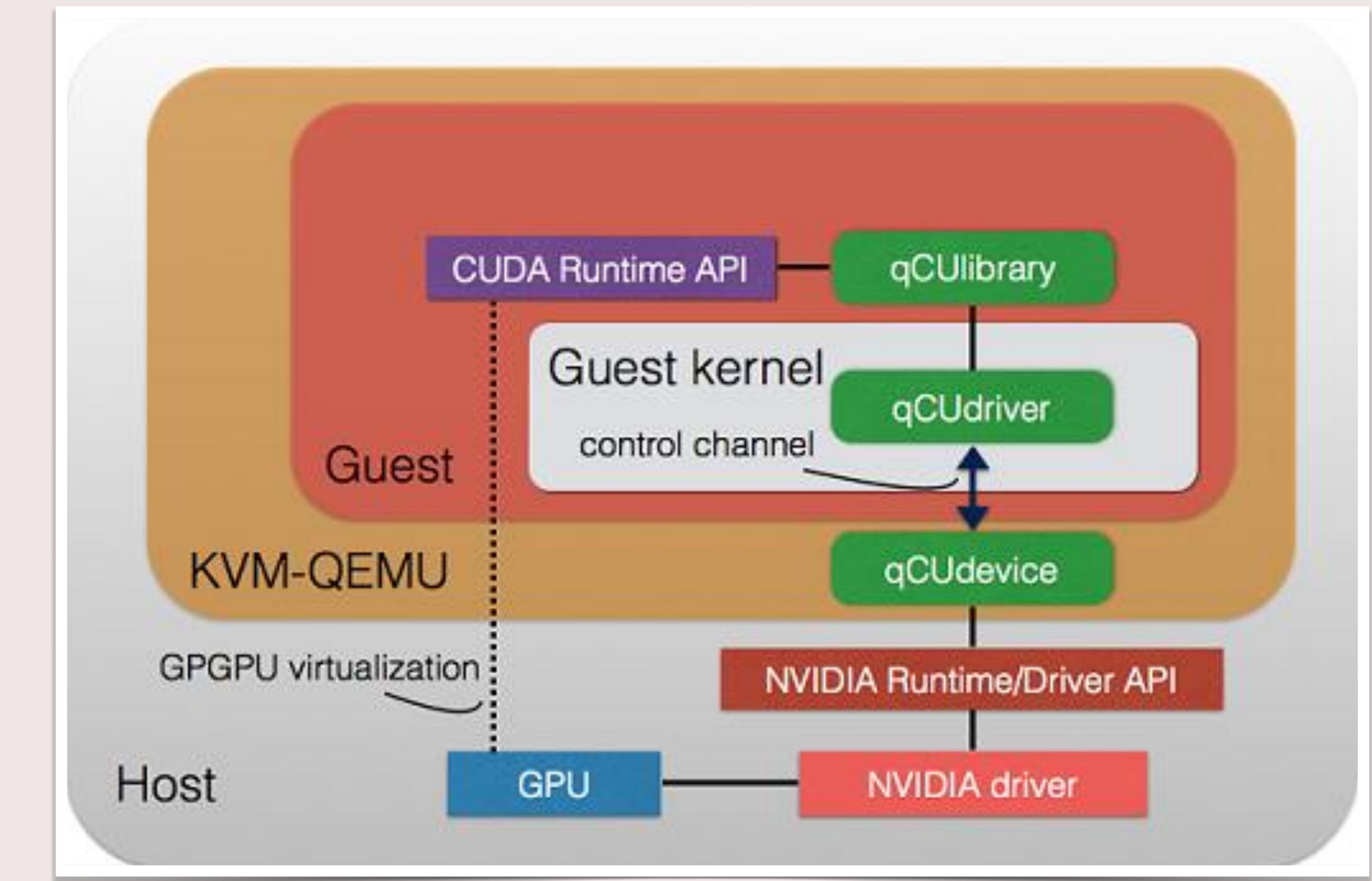
2010 - 2018 (8 years)

- Lecturer (2010 - 2018)
 - Parallel, GPU and Virtualization Technique: Design and teaching related courses for the industry and academic.
- Tsinghua University (Summer in 2013 & 2015)
 - Parallel Programming course: Design the course and teach students to implement the parallel computing algorithm by pthread, MPI and CUDA for solving the practical problem.

qCUDA

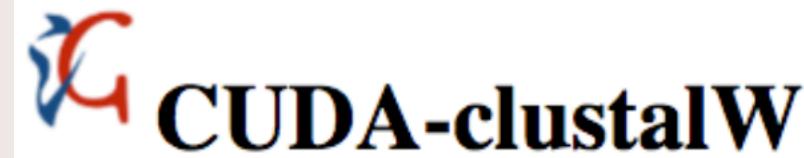
Oct 2016 - Dec. 2017

- qCUDA - A framework of GPGPU virtualization
- NTHU System Software Lab
- <https://github.com/coldfunction/qCUDA>
- Doctoral Dissertation: A New API Remoting Policy for General-Purpose Computing on GPU Virtualization and its Application on Biological Tools



CUDA-clustalW & GPU-REMuC Tool

2010 - 2012



<https://github.com/coldfunction/CUDA-clustalW>

[downloads](#) | [Documents](#) |

In computational biology, sequence alignment is of priority concern and many methods have been developed to solve sequence alignment-related problems for biological applications. ClustalW is a progressive multiple sequence alignment tool to align a set of sequences by repeatedly aligning pairs of sequences and previously generated alignments. Several algorithms or tools have been ported on GPUs with CUDA in computational biology, such as MUMmerGPU, CUDA-ClustalW, and GPU-REMuC. This work presents the development of CUDA-ClustalW, a CUDA port of ClustalW v2.0.9 processing pipeline by using inter-task parallelization. CUDA ClustalW v1.0 is implemented in C++ and achieves a performance of 22x speedups for 1000 sequences with length of 1532 in the distance of Synchronous Diagonal Multiple Threads type. Several optimization methods were designed in CUDA-ClustalW v1.0 can achieve about 22x speedups for 1000 sequences with length of 1532 in the distance of Synchronous Diagonal Multiple Threads type. Several optimization methods were designed in CUDA-ClustalW v1.0 can achieve about 33x speedups by comparing to ClustalW v2.0.1.

Downloads and version history

Some descriptions here:

Version 1.0.0 (March 2013) [download](#) (2.6 MB)

- Linux x86 64-bit
- CPU/GPU coprocess
- Support Multiple GPUs
- NVIDIA CUDA support
- Base on clustalW 2.0

[Contact Us](#) | ©2010-2014 YSL

The screenshot shows two windows side-by-side. The left window is titled 'RE-MuC' and contains a text input field for genomic sequences in FASTA format, a scrollable text area displaying sequence alignments, and several input fields for parameters like scoring matrix (set to GONNET 250), gap open penalty (10.0), and gap extension penalty (0.2). The right window is titled 'RE-MuC Results' and displays the aligned sequences in a large text area, showing various protein identifiers and their corresponding aligned amino acid sequences.

RE-MuC

Help Examples

Enter your genomic sequences with FASTA format below (copy & paste): GPU-REMuC

>P22238|DHA CRAPL Desiccation-related protein clone PCC27-04 - Craterostigma plantagineum.
MAHSLGEQYDLGKPTEEHHESHESHPAHQAPHAGGELGAGQKTSQLARSNSSSSSSEDDGQ
GGRRKKPIKEVKKEKLPGGGAGGKGQTGECACTTTAAAGGHEEKKGVMEMIKEKLPGQH

>P30185|DHR18 ARATH Dehydrin Rab18 - Arabidopsis thaliana (Mouse-ear cress).
MASYONRPGQATDEYGNPIQQYDEYGNPMGGGGTGGGGATGGQGYGTGGQGYGSG
GQGYGTGGQGYGTGTGTEGFTGGGARHHGQEQLHKESGGGLGMLHRSNSGSNSSSSED
GQGGRRKKGITQKIKEKLPGHHDQSGQAAMGGMGSGYDAGGYGGEHHEKKGMDKIKE
LPGGR

>Q07322|ECP40 DAUCA Embryogenic cell protein 40 - Daucus carota (Carrot).
MADLRDEKGNPIQLTDQHGNPVQLTDEYGNPVHITGVATTGATTGHDHGVGGAHSQVG
STGLGGVAGAAGLATAAATGGSHGGTGTGVGAAGHGGTGTGTTGTT
GQTHGMGPTGIGGTHGVAAGGGTGGVQLGQTHGMGPTGTAAGHGLGTGQSGLGSS
YATHGLLGTGIGTGSAPASAGSHSHAPEKKTALGEOLHRSNSSSSSEDDGQGGRRKKGF
TTKIKEKLGGGKHKKDEHTTATTAAHPGGAAVAEVHEHEKKSMLDKIDKLPGH
HSPTSH

Parameters:

Scoring matrix: GONNET 250

Gap open penalty: 10.0 Gap extension penalty: 0.2

Regular expression constraint(s): S(4)-[SD]-[DE]-x-[DE]-[GVE]-x(1,7)-[GE]-x(0,2)-[KR](4) [KR]-[LIM]-

Execute Reset ShowResults

RE-MuC Results

P46524|CO410 WHEAT
P31168|COR47 ARATH
P09442|DH11 GOSHI
P22911|DH16B ORYSA
P22238|DHA CRAPL
P30185|DHR18 ARATH
Q07322|ECP40 DAUCA

MEDERSTQS-----YQGGEAAEQVEVTDR-----GLLGNLLG--
MAEYKNNVPEHETPTVATESPATTEVTDRG-----LPFLPLGKKEE--
MAHPQN-----QVSAPETQTDAYGNP-----TRRTDEYEG--
MDNYQGQHG-----YG-----ADRVDVYGNP-----VGAGQYG--
MAHSLG-----EQYDLGKP-----TEEEH--
MASYONRPGG-----QATDEYGNPIQQYDEYGNP-----MGGGYG--
MADLRDEKGNPIQLTDQHGNPVQLT-DEYGNPVHITGVATTGATTGHDHGVGGAHSQVG
*-----
-----KKKAEEDK-----EKEEE
-----EVKPQETTLES-----EFDPHKQISE
-----NP1PTQET-----GRGI
-----GGATAPCG-----GHGA
-----ESHPP-----AHQA
--TGGG-----GGATGGQG-YGTG-----GQGYGS
GSTGLGGVAGAAGLAGATAAAATHGSHGGTGTGVGVGAHHGGTGTGTTGLGTGTG

-----LVTG--MEKVSVE-----EPEVKK
-----PELAEEHEEVKENK-----ITLLEELQE
-----LGIG-----GHHHLR
-----MGMG-GHAGAGAG-----GQFQPAR
-----PHAG-----QRTSQLAR
-----GGQGYGTGGQGYG-----TGTG-----TEGFG-TGGGARH-HQQECLIK
TGQTHGMGPTGIGGTHGVGSTGIGGAHHGGTGVLGQTHGMGPTGTGAAGHGLGTGQSGLGSS

-----EEHEDGEKKE-----TLFSKLHRSNSSSSSSSDEEEEVIDDNN
-----KTEEDEENKP-----SVIEKLHRSNSSSSSSDEE-----
-----TGSSSS-----SSSSSEDE-----
-----EDHKTG-G-----ILHRSNSSSSSSSSEDD-----
-----SNSS-----SSSSSEDD-----
-----ESGGGLGG-----MLHRSNSGSSSSSEDD-----
-----SYATHGLLGTGIGTGSAPASAGSHSHAPEKKTALGEOLHRSNS-----SSSSSEDD-----
****.:::

-----GEVIKRKK-----KKGLKEKLQGKLPG-HKDTEGEHTVGLPAPAAPASVQTHGGHH
-----GEKKKEKKK-----IVEGEEDKKGKLVKEIKEKLPG-HHDKTAEDDVFPVST-TIPVFPV-----
-----GIG-KKKK-----GLKE-----
-----GMGGRKK-----GIKEKIKEKLPGGNKGNNHQQQMMGNTGGAYGQGQGHAGMT
-----P22238|DHA CRAPL-----KQ-----TG-----ECGTTTT
-----P30185|DHR18 ARATH-----GQGGRKK-----GTTQKIKEKLPG-----HHDQSGQAAMGGMGSGYDAGG-----
-----GQGGRKK-----GTTKIKEKLGGGKHKKDEHTTATTAAHPGGA-----
* : ** :

Save as FASTA format

Other Projects

2011 - 2014

- 3LAllocator
 - A Non-Blocking Approach on GPU Dynamical Memory Management
- GOOS-SM
 - GPU-Oriented Operations Server for Sparse Matrix
- CrackRSA
 - An efficient parallel RSA decryption algorithm for many-core GPUs with CUDA
- CUDA programming competition 2012 & 2011: The Second Place Award and Honorable Mention in the CUDA programming competition 2012 & 2011 in Taiwan

Publish

<https://coldfunction.github.io/2021/01/06/Publish/>

Conference

- Chieh-Yu Yu, Che-Rung Lee, Po-Jui Tsao, [Yu-Shiang Lin](#), Tzi-Cker Chiueh, “Efficient Group Fault Tolerance for Multi-tier Services in Cloud Environments”, ICC, 2020.
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Che-Rung Lee, Yeh-Ching Chung, “qCUDA: GPGPU Virtualization for High Bandwidth Efficiency”, CloudCom, 2019.
- Ming-Ting Wei, [Yu-Shiang Lin](#), Che-Rung Lee, “Performance Optimization for InfiniBand Virtualization on QEMU/KVM”, CloudCom, 2019.
- Chun-Yuan Lin, Chung-Hung Wang, Che-Lun Hung, [Yu-Shiang Lin](#), “Efficient parallel algorithm for compound comparisons on multi-GPUs”, IEEE -BIBM, 2014.
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Jon-Yu Lee, “A New Non-Blocking Approach on GPU Dynamical Memory Management”, IWCSE, 2013.
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Yeh-Ching Chung, “GPU-Based Cloud Service for Multiple Sequence Alignments with Regular Expression Constrains”, CloudCom, 2012.
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Der-Chyuan Lou, “Efficient Parallel RSA Decryption Algorithm for Many-core GPUs with CUDA”, ICTSM2012.
- Yu-Rong Chen, Che Lun Hung, [Yu-Shiang Lin](#), Chun-Yuan Lin, Tien-Lin Lee, Kual-Zheng Lee, “Parallel UPGMA Algorithm on Graphics Processing Units Using CUDA”, FGC2012.
- Chun-Yuan Lin, [Yu-Shiang Lin](#), Jiayi Zhou, and Chuan Yi Tang, “GPU-REMuSiC: Efficient Constrained Multiple Sequence Alignment Algorithm on Graphics Processing Units”, CTHPC, 2011.
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Sheng-Ta Li, Joy Lee, and Chuan Yi ang, “GPU-REMuSiC: the implementation of Constrain Multiple Sequence Alignment on Graphics Processing Units”, NVIDIA GPU Computing Seminar, 2010.

Journal

- Che-Lun Hung, [Yu-Shiang Lin](#), Chun-Yuan Lin, Yeh-Ching Chung, Yi-Fang Chung, “CUDA ClustalW: An efficient parallel algorithm for progressive multiple sequence alignment on Multi-GPUs”, CBAC(2015).
- Chun-Yuan Lin, Chung-Hung Wang, Che-Lun Hung, and [Yu-Shiang Lin](#), “Accelerating Multiple Compound Comparison Using LINGO-Based Load-Balancing Strategies on Multi-GPUs”, IJG (2015).
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Che-Lun Hung, Yeh-Ching Chung and Kual-Zheng Lee: GPU-UPGMA: high-performance computing for UPGMA algorithm based on graphics processing units.CCPE(2014).
- Chun-Yuan Lin, [Yu-Shiang Lin](#): Efficient parallel algorithm for multiple sequence alignments with regular expression constraints on graphics processing units. IJCSE 9(1/2): 11-20 (2014).
- [Yu-Shiang Lin](#), Chun-Yuan Lin, Hsiao-Chieh Chi, Yeh-Ching Chung: Multiple Sequence Alignments with Regular Expression Constraints on a Cloud Service System. IJGHPC 5(3): 55-64 (2013).

Book

- CUDA 輕鬆上手，新世代GPU應用技術：林俊淵、周嘉奕、[林郁翔](#)、李昇達等人合著 (ISBN: 9572239171)

