



**AFRL-RY-WP-TR-2022-0155**

## **OPENTIMER AND DtCRAFT**

**Guanxi Chen**  
**University of Illinois**

**MAY 2022**  
**Final Report**

**DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.**

*See additional restrictions described on inside pages*

STINFO COPY

**AIR FORCE RESEARCH LABORATORY**  
**SENSORS DIRECTORATE**  
**WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320**  
**AIR FORCE MATERIEL COMMAND**  
**UNITED STATES AIR FORCE**

## NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with The Under Secretary of Defense memorandum dated 24 May 2010 and AFRL/DSO policy clarification email dated 13 January 2020. This report is available to the general public, including foreign nationals.

Copies may be obtained from the Defense Technical Information Center (DTIC)  
(<http://www.dtic.mil>).

AFRL-RY-WP-TR-2022-0155 HAS BEEN REVIEWED AND IS APPROVED FOR  
PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

\*//Signature//

---

CHRISTOPHER A. BOZADA  
Program Manager  
Aerospace Components and Subsystems Division

\*//Signature//

---

LESTER C. LONG, Lt Col, USAF  
Deputy Chief  
Aerospace Components and Subsystems Division  
Sensors Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

\*Disseminated copies will show “//Signature//” stamped or typed above the signature

## REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

<b>1. REPORT DATE</b> May 2022	<b>2. REPORT TYPE</b> Final	<b>3. DATES COVERED</b>	
		<b>START DATE</b> 11 June 2018	<b>END DATE</b> 11 June 2020
<b>4. TITLE AND SUBTITLE</b> OPENTIMER AND DtCRAFT			
<b>5a. CONTRACT NUMBER</b> FA8650-18-2-7843	<b>5b. GRANT NUMBER</b> N/A	<b>5c. PROGRAM ELEMENT NUMBER</b> 62716E	
<b>5d. PROJECT NUMBER</b> N/A	<b>5e. TASK NUMBER</b> N/A	<b>5f. WORK UNIT NUMBER</b> Y1TD	
<b>6. AUTHOR(S)</b> Guanxi Chen			
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> University of Illinois Chicago 601 South Morgan Street Chicago, Illinois, 60607			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> Air Force Research Laboratory, Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command, United States Air Forces te	<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b> AFRL/RYPD		<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b> AFRL-RY-WP-TP-2022-0155
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.			
<b>13. SUPPLEMENTARY NOTES</b> This material is based on research sponsored by the Air Force Research Lab (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under agreement number FA8650-18-2-7843. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Labs (AFRL), the Defense Advanced Research Projects Agency (DARPA) or the U.S. Government. Report contains color.			
<b>14. ABSTRACT</b> The effort produced several releases of OpenTimer on the 2.x line. The releases included a major one to implement a new incremental timing engine and new API concepts that improved the performance of parallel incremental timing. The team delivered several minor releases that enhanced the support for surrounding tasks such as reader/writer, interpreter, and so on. The team evaluated the path and slack accuracy of OpenTime with respect industrial tools. OpenTimer accurately identifies timing critical paths 10 times faster. In addition to OpenTimer, the team created a project Cpp-Taskflow for efficient parallel task programming using the modern C++ technology.			
<b>15. SUBJECT TERMS</b> opentimer; slack accuracy; timing critical paths; parallel computing			
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified	SAR
			<b>18. NUMBER OF PAGES</b> 11
<b>19a. NAME OF RESPONSIBLE PERSON</b> Chris Bozada			<b>19b. PHONE NUMBER (Include area code)</b>

# Table of Contents

Section	Page
List of Figures.....	ii
1 PHASE 1-A FINAL REPORT: OPENTIMER AND DTCRAFT .....	1
1.1 Summary of Phase 1-a Achievements .....	1
2 OPENTIMER .....	2
3 TASKFLOW (CPP-TASKFLOW).....	3
4 PUBLICATION UNDER PHASE 1-A .....	4
5 AWARD UNDER PHASE 1-A .....	5
LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS .....	6

## List of Figures

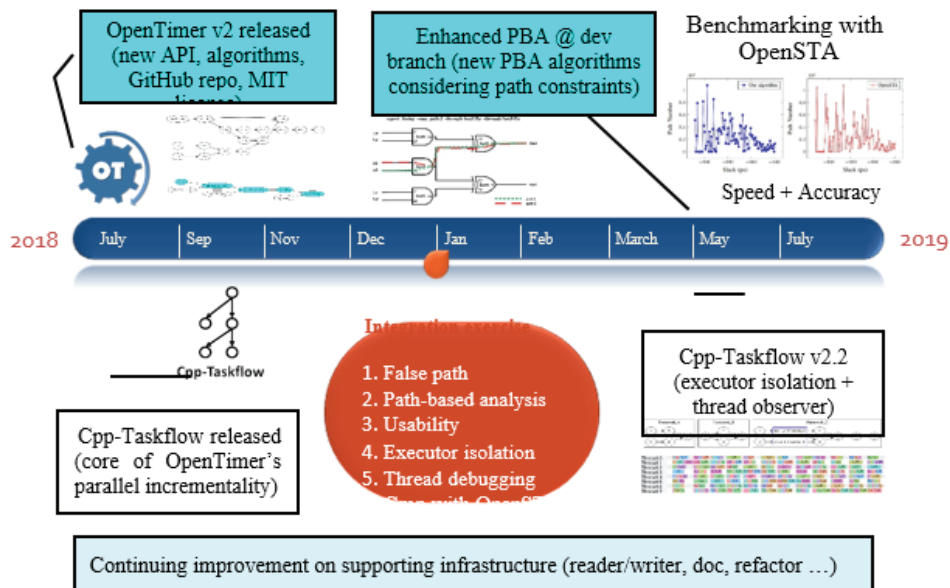
<b>Figure</b>	<b>Page</b>
Figure 1: Release Milestone of Phase 1-a .....	1
Figure 2: Screenshot of the OpenTimer Repository .....	2
Figure 3: Performance Comparison between OpenTimer and OpenSTA .....	2
Figure 4: Screenshot of the Taskflow project.....	3

# 1 PHASE 1-A FINAL REPORT: OPENTIMER AND DTCRAFT

## 1.1 Summary of Phase 1-a Achievements

In year one (Phase I-A), we made several releases of OpenTimer on the 2.x line. This included a major release that implemented a new incremental timing engine and new API concepts to largely improve the performance of parallel incremental timing, and minor releases that enhanced the support for surrounding tasks such as reader/writer, interpreter, and so on. We also evaluated the path and slack accuracy of OpenTimer to that of industrial tools. The results showed that OpenTimer can accurately identify timing critical paths with more 10x speed-up.

In addition to OpenTimer, we have created a project Cpp-Taskflow for efficient parallel task programming using the modern C++ technology. The motivation behind this project is to provide a new task-based programming model to support incremental timing with strong scaling of parallel computations. Cpp-Taskflow is the core of OpenTimer and can be used for generic parallel task programming. We have developed a very simple and user-friendly API that lets developers enable efficient parallel decomposition strategies, particularly for workloads with irregular compute patterns such as EDA computing and machine learning.



**Figure 1: Release Milestone of Phase 1-a**  
*Kickoff to the second integration exercise*

## 2 OPENTIMER

OpenTimer is available at: <https://github.com/OpenTimer/OpenTimer>

### OpenTimer



build passing c++ 17 download latest version v2.0-alpha Ask me anything doc wiki License MIT

A High-Performance Timing Analysis Tool for VLSI Systems

### Why OpenTimer?

OpenTimer is a new **static timing analysis (STA)** tool to help IC designers quickly verify the circuit timing. It is developed completely from the ground up using *C++17* to efficiently support parallel and incremental timing. Key features are:

- Industry standard format (.lib, .v, .spef, .sdc) support
- Graph- and path-based timing analysis
- Parallel incremental timing for fast timing closure
- Award-winning tools and golden timers in CAD Contests

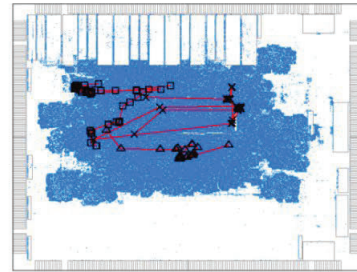


Figure 2: Screenshot of the OpenTimer Repository

OpenTimer develops a unique software architecture to efficiently enable parallel incremental timing. We group each timing operation into one of the following three categories, *builder*, *action*, and *accessor*, to largely improve the performance of parallel incremental timing. The following figure shows the comparison with the OpenSTA tool:

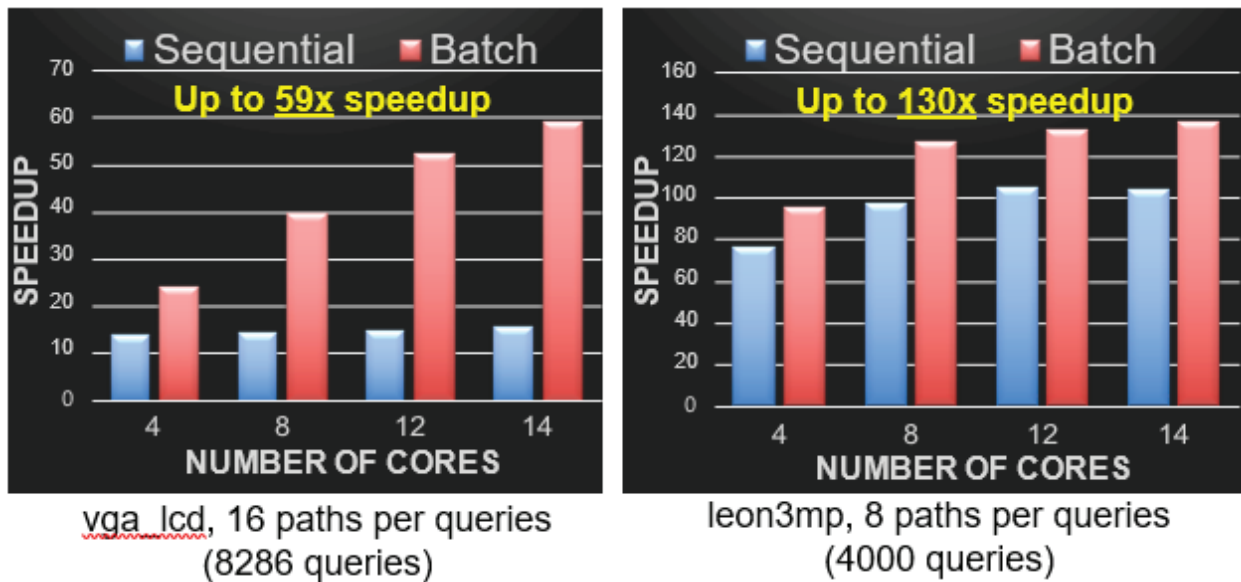


Figure 3: Performance Comparison between OpenTimer and OpenSTA

### 3 TASKFLOW (CPP-TASKFLOW)

Taskflow is available at: <https://taskflow.github.io/>

## Taskflow



code quality **A** build passing build passing API docs profiler tfp cs.DC arXiv:2004.10908

Taskflow helps you quickly write parallel tasks programs in modern C++

## Why Taskflow?

Taskflow is faster, more expressive, and easier for drop-in integration than many of existing task programming frameworks in handling complex parallel workloads.

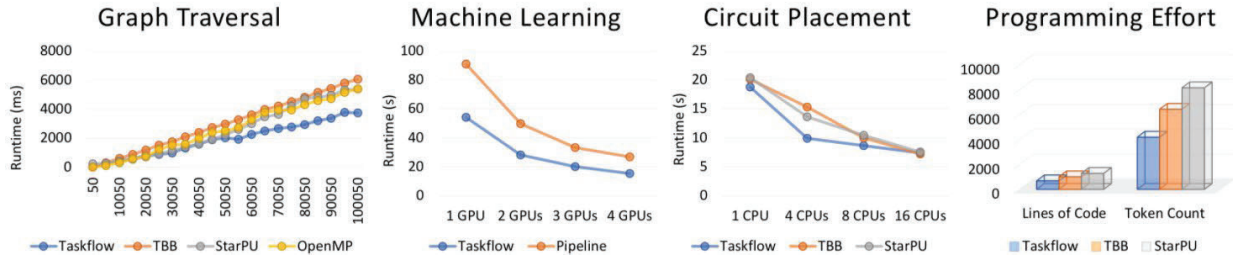


Figure 4: Screenshot of the Taskflow project

Taskflows introduces a new powerful programming model to help developers quickly program parallel and heterogeneous workloads with high performance and simultaneous high productivity. The project has received almost 1M downloads and is being used by many industrial and academic research projects. Taskflow is the core parallel computing engine of the OpenTimer project.

## 4 PUBLICATION UNDER PHASE 1-A

1. C.-X. Lin, T.-W. Huang, G. Guo, and Martin D. F. Wong, "A Modern C++ Parallel Task Programming Library," ACM Multimedia Conference (MM), Nice, France, 2019.
2. C.-X. Lin, T.-W. Huang, G. Guo, and Martin D. F. Wong, "An Efficient and Composable Parallel Task Programming Library," IEEE High-performance and Extreme Computing Conference (HPEC), Waltham, MA, 2019.
3. T.-W. Huang, C.-X. Lin, G. Guo, and Martin D. F. Wong, "Cpp-Taskflow: Fast Task-based Parallel Programming using Modern C++," IEEE International Parallel and Distributed Processing Symposium (IPDPS), Rio de Janeiro, Brazil, 2019.
4. K.-M. Lai, T.-W. Huang, and T.-Y. Ho, "A General Cache Framework for Efficient Generation of Timing Critical Paths," ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, 2019.
5. T.-W. Huang, C.-X. Lin, G. Guo, and Martin D. F. Wong, "Essential Building Blocks for Creating an Open-source EDA Project," ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, 2019.
6. T.-W. Huang, C.-X. Lin, and Martin D. F. Wong, "Distributed Timing Analysis at Scale," ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, 2019.

## **5 AWARD UNDER PHASE 1-A**

1. Open-source Software Competition Finalist (Cpp-Taskflow), ACM Multimedia Conference, 2019 (to compete this Oct)
2. ACM SIGDA Outstanding PhD Dissertation Award, 2019 (1 out of N PhDs worldwide)
3. Best Open-source Software Award (DtCraft), ACM Multimedia Conference, 2018 (2 out of 14)
4. Best Open-source EDA Tool Award (OpenTimer), Workshop on Open-source EDA Technology at ICCAD, 2018 (1 out of 31)
5. Best Poster Award (Cpp-Taskflow), the Official C++ Conference, 2018 (voted by >1000 developers)

## **LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS**

AFRL	Air Force Research Laboratory
API	Application Programming Interface
DARPA	Defense Advanced Research Projects Agency
EAR	Export Administration Regulation
EDA	Electronic Design Automation
ITAR	International Traffic in Arms Regulation