Debugging SharP: SHARed data-structure centric Programming

Marie McCord – URECA Proposal Fall 2020

System architectures used for both Big-Compute and Big-Data are composed of hierarchical and heterogeneous on-node memory. As a result, Big-Compute and Big-Data applications are increasingly converging, and while system architectures support this convergence, there is a lack of programming models that support either hierarchical-heterogeneous systems or the convergence of Big-Compute and Big-Data, creating a software gap.

Different architectures have various types of heterogeneous memory in terms of organization, quantity, quality, and characteristics. This makes portability for applications for these architectures a challenge. While these architectures are effective for both Big-Compute and Big-Data, applications designed for the convergence of Big-Compute and Big-Data require supporting programming models.

The programming abstraction named SHARed data-structure centric Programming (SharP) addresses these issues by using distributed data objects to encapsulate hierarchical and heterogeneous memories across nodes and map data structures onto these objects, thereby abstracting these memory systems into a simple and unified user interface that is portable across architectures. SharP is data-centric in that it reduces data movement and duplication, and provides the user extensive control over the data organization. Additionally, it is compatible with Big-Compute programming models such as MPI and OpenSHMEM [1].

At this moment, there are two functions in the program related to the distributed hash data structures that present logical errors and require debugging. The first function moves elements around in the hash data structure in order to create a free slot for data when one is not

available. This function is fundamental to the distributed hash data structure, but presently it fails to execute correctly when the data is distributed over three or more blocks in the structure. The second function returns the locality information for a distributed hash data structure, specifically whether the data structure is local and belongs entirely to the provided processing element or if it is remote among multiple processing elements. This function is a main feature of the program but is currently returning inaccurate information. ***The goal of this project is to debug these two functions so that they correctly execute.***

The faculty mentor who will be overseeing my work on this project is Dr. Ferrol Aderholdt. As one of the creatives of SharP, he will be guiding me in this research as well as helping me with any questions or issues that may arise. We intend to maintain contact during the semester through weekly meetings.

Through this project I will gain experience working with hierarchical-heterogeneous memory systems, data-centric models, hashing, and general debugging. These are skills that will serve me throughout my career in computer science. I am aiming to continue my education upon completion of my undergraduate degree in December 2021 through a graduate program in data science or computational science, and to eventually pursue a PhD. This project will also give me insight into the research process and experience with academic writing.

References:

[1] F. Aderholdt, M. Gorentla Venkata, Z. Parchman. SharP: Towards programming extreme-scale systems with hierarchical heterogeneous memory. *IEEE 46th International Conference on Parallel Processing Workshops*, pp. 145-154, August 2017.

Project Timeline

1. **Debug the make_room function for distributed hash data structures.** The first month, approximately 5-6 weeks, will be spent debugging the function that moves elements around to create a free slot for data in the hash data structure when one is not available. This function currently presents a logical error when the data is distributed over three or more blocks in the structure.
2. **Debug the function that returns the locality information for a distributed hash data structure.** The second month, approximately 4-5 weeks, will be spent debugging the function that returns whether a distributed hash data structure is local and belongs entirely to the provided processing element or if it is remote across multiple processing elements. Presently, the function presents issues accurately returning locality but is a main feature of the program.
3. **Update the test program and test SharP on a remote hierarchical-heterogeneous computer via Chameleon Cloud.** The last month, approximately 4 weeks, will be spent updating the test program in order to run SharP and thoroughly test the newly debugged functions. Additional debugging will continue throughout this phase until both functions are properly executing.

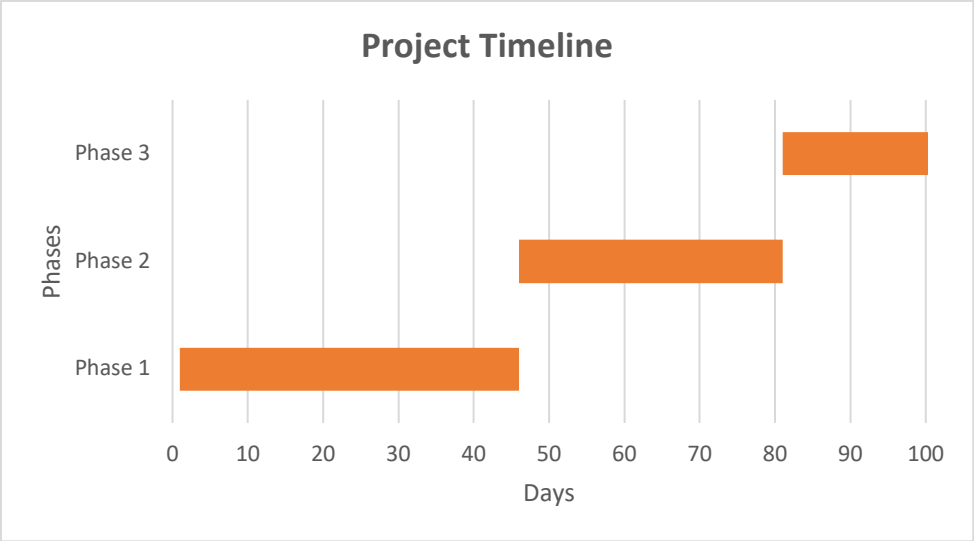

Figure 1 – Phase 1: Debug the make_room function for distributed hash data structures (approximately 5-6 weeks). Phase 2: Debug the function that returns the locality information of a distributed hash data structure (approximately 4-5 weeks). Phase 3: Rewrite the test program and test SharP using a remote hierarchical-heterogeneous computer via Chameleon Cloud (approximately 4 weeks).

Academic Transcript

This is not an official transcript. Courses which are in progress may also be included on this transcript.

Special grades to note are:
FA = Failure and stopped attending
T_ = Transfer grades with leading "T" are not calculated in the overall and overall combined GPAs, but do count in the lottery GPA. Leading “T” grades were started
Summer 2015 for new undergraduate transfer credits regardless of the term the course was completed.
X = Grade not submitted by course instructor and not used in calculating grade point average until final grade submitted by instructor

The repeat indicator column denoted by an "R" after the Quality Points column translates as follows:
E = Excluded from GPA and Earned Hours
A = Included in GPA, but not Earned hours
I = Included in GPA and Earned Hours
F = Frozen and exempt from repeat processing (i.e., repeatable courses)
. = Excluded from GPA and Earned Hours – Academic Fresh Start

Note: Additional information about all grades and repeats are available in the University Catalog

Click here to Print Unofficial Transcript (Chrome and FireFox Only)

Transfer Credit Institution Credit Transcript Totals Courses in Progress

Transcript Data

STUDENT INFORMATION

Student Type:	Continuing

Curriculum Information

Current Program

Bachelor of Science	
College:	Basic and Applied Sciences
Major and Department:	Computer Science, Computer Science
Major Concentration:	Professional Computer Science
Minor:	Statistics

***Transcript type:Advising-Unofficial Transcript is NOT Official ***

TRANSFER CREDIT ACCEPTED BY INSTITUTION -Top-

Fall 2010:	Univ California San Diego

Subject	Course	Title	Grade	Credit Hours	Quality Points	R
ELEC	ELLD	LD:Culture,Art & Technology I	TC+	2.670	0.000	

Current Term:	20.000	14.000	14.000	0.000	0.000	0.000

Unofficial Transcript

Spring 2010: Adv Placement Program

Subject	Course	Title	Grade	Credit Hours	Quality Points	R
ENGL	1010	Expository Writing	TP	3.000	0.000	
SPAN	1010	Elementary Spanish I	TP	3.000	0.000	I
SPAN	1020	Elementary Spanish II	TP	3.000	0.000	I

	Attempt Hours	Passed Hours	Earned Hours	GPA Hours	Quality Points	GPA
Current Term:	9.000	9.000	9.000	0.000	0.000	0.000

Unofficial Transcript

INSTITUTION CREDIT -Top-

Term: Fall 2019

College:	University College
Major:	Integrated Studies
Student Type:	New Transfer
Academic Standing:	Good Standing
Additional Standing:	Dean's List

Subject	Course	Level	Title	Grade	Credit Hours	Quality Points	R	CEU Contact Hours
COMM	2200	UG	(EXL) Fundamentals of Communication	A	3.000	12.000		
CSCI	1170	UG	Computer Science I	A	4.000	16.000		
ENGL	1020	UG	Research and Arg Writing	A	3.000	12.000		
MATH	1910	UG	Calculus I	A	4.000	16.000		

Term Totals (Undergraduate)

	Attempt Hours	Passed Hours	Earned Hours	GPA Hours	Quality Points	GPA
Current Term:	14.000	14.000	14.000	14.000	56.000	4.000
Cumulative:	14.000	14.000	14.000	14.000	56.000	4.000

Term: Spring 2020

College:	University College
Major:	Integrated Studies
Student Type:	Continuing
Academic Standing:	Good Standing

Subject	Course	Level	Title	Grade	Credit Hours	Quality Points	R	CEU Contact Hours
CSCI	2170	UG	Computer Science II	A	4.000	16.000		
CSCI	3080	UG	Discrete Structures	A	3.000	12.000		
MATH	1920	UG	Calculus II	A	4.000	16.000		
PRST	3995	UG	(EXL) Interdisciplinary Research and	A	3.000	12.000		

			Problem Solving					

Term Totals (Undergraduate)

	Attempt Hours	Passed Hours	Earned Hours	GPA Hours	Quality Points	GPA
Current Term:	14.000	14.000	14.000	14.000	56.000	4.000
Cumulative:	28.000	28.000	28.000	28.000	112.000	4.000

Term: Summer 2020

College:	Basic and Applied Sciences
Major:	Computer Science
Student Type:	Continuing
Academic Standing:	Good Standing
Additional Standing:	Dean's List

Subject	Course	Level	Title	Grade	Credit Hours	Quality Points	R	CEU Contact Hours
BIA	2610	UG	Statistical Methods	A	3.000	12.000		
MATH	2010	UG	Elements of Linear Algebra	A	3.000	12.000		
MATH	2050	UG	Probability and Statistics	A	3.000	12.000		
PHYS	2011	UG	Physics Problems Laboratory I	A	4.000	16.000		

Term Totals (Undergraduate)

	Attempt Hours	Passed Hours	Earned Hours	GPA Hours	Quality Points	GPA
Current Term:	13.000	13.000	13.000	13.000	52.000	4.000
Cumulative:	41.000	41.000	41.000	41.000	164.000	4.000

Unofficial Transcript

TRANSCRIPT TOTALS (UNDERGRADUATE) -Top-

	Attempt Hours	Passed Hours	Earned Hours	GPA Hours	Quality Points	GPA
Total Institution:	41.000	41.000	41.000	41.000	164.000	4.000
Total Transfer:	172.580	109.540	109.540	0.000	0.000	0.000
Overall:	213.580	150.540	150.540	41.000	164.000	4.000

	Attempt Hours	Passed Hours	Earned Hours	GPA Hours	Quality Points	GPA
Institution Combined:	41.000	41.000	41.000	41.000	164.000	4.000
Transfer Combined:	172.580	109.540	109.540	0.000	0.000	0.000
Overall Combined:	213.580	150.540	150.540	41.000	164.000	4.000

Unofficial Transcript

COURSES IN PROGRESS -Top-

Term: Fall 2020

College:	Basic and Applied Sciences
Major:	Computer Science
Student Type:	Continuing

Subject	Course	Level	Title	Credit Hours

CSCI	3110	UG	Algorithms and Data Structures	3.000
CSCI	3130	UG	Assembly and Computer Organization	4.000
CSCI	3180	UG	Introduction to Numerical Analysis	3.000
MATH	2530	UG	Applied Statistics II	3.000

Unofficial Transcript

RELEASE: 8.7.1 PROD - SSBPROD3