Hash Function Collision Searching
Team Mordor

Alex Young & Emmanuel Dodoo

October 22, 2018
Outline

● Summary of problem
● Research Paper Analysis
  ○ Paper 1
  ○ Paper 2
  ○ Paper 3
● Conclusion
● References
Problem Overview

- Takes in any length string text
  - Breaks message into 512-bit blocks
- Hash function outputs 128-bit value
- As of Dec. 2008 the US Government discourages further use of MD5 for cryptographic hashing

Source: https://www.k-ict.org/v4/online-security/md5-hash/

Computational Problem

Investigate & quantify collision time of attack using sequential and parallelized approaches
Title: Finding Hash Collisions by Brute Force Parallel Programming

Publish Year: 2016

Publisher: IEEE

Affiliated Organizations: University of North Alabama, Northern Kentucky University, Bryn Mawr College, College of Staten Island (CUNY)
Research Paper One Analysis

Problem: Compare and contrast target string sizes and number of processor

Papers Contribution:

- Hash Collision searching via parallel computation, load balancing is essential to a timely output
- Using a MD5 Hex with 4 bytes, each processor added to solve the problem we saw the solution time reduced by 50%.

How do we use this in our project? Based off of the research done by these authors, we will evaluate how they came up with their load balancing plan and generate one of our own.
Research Paper Two Details

**Title:** Constant Memory Optimization in MD5 Crypt Cracking Algorithm on GPU-Accelerated Supercomputer using CUDA

**Publish Year:** 2012

**Publisher:** International Conference on Computer Science & Education (ICCSE)

**Affiliated Organizations:** National University of Defence Technology, Changsha, Hunan, P.R. China
Research Paper Two Analysis

**Problem:** Improve speed of Brute Force Attack for MD5 Crypt using memory optimizations

**Papers Contribution:**

- Utilize constant memory on CUDA and limit to read-only.
  - Memory can be cached reducing the cost by consecutive reads
- Evaluated on CPU only, GPU only, CPU + GPU, single and multiple nodes, achieving **44.6% increase in performance** with optimizations (GPU)

**How do we use this in our project?** Based off constant memory optimizations, we will evaluate potential implementation on GPU based device for improved performance.
Research Paper Two Analysis

Fig. 3. Performance On The Single Computing Node

Fig. 4. Performance On The Multi-Computing Node
Title: Parallel Hash Collision Search by Rho Method with Distinguished Points

Publish Year: 2018

Publisher: IEEE

Affiliated Organizations: University of Maryland, College of Staten Island (CUNY)
Research Paper Three Analysis

**Problem:** Decrease the memory usage in an MD5 collision search while increasing the probability of finding a collision

**Papers Contribution:**

- Discusses a method in which they evaluate a possible collisions if the message digest has a certain number of leading 0’s
- Confirms our notion of using a Master-Worker architecture

**How do we use this in our project?** Based off of the research done by these authors, they discussed the Master and worker relationship in detail in which we can base our solution off of using PJ2.
THANK YOU

QUESTIONS ?
References

