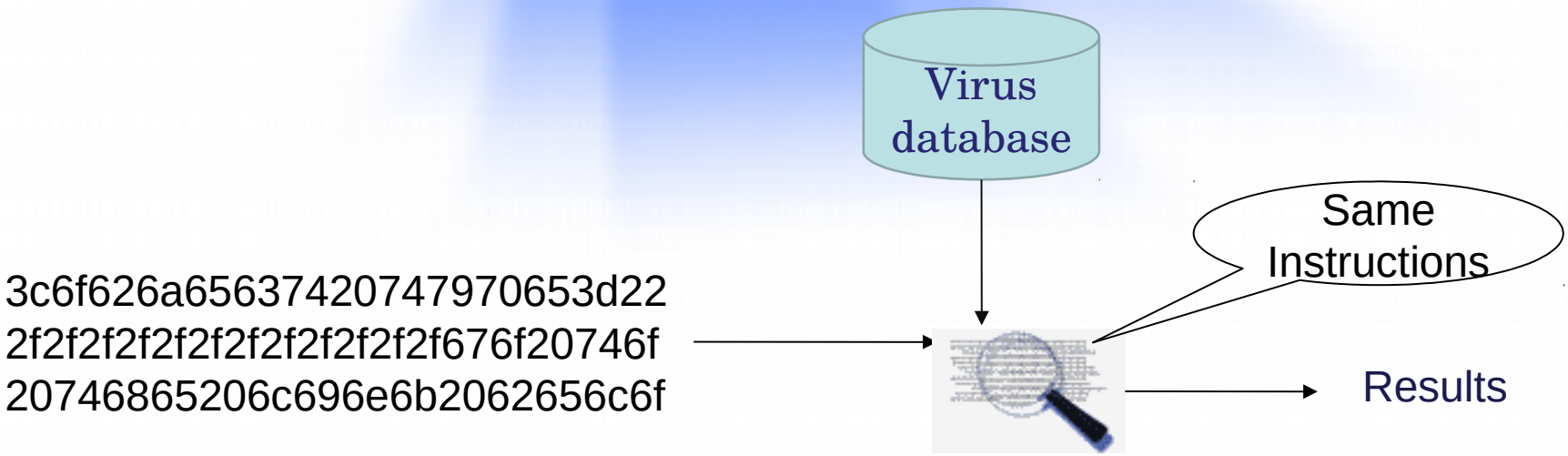
ACCELERATING VIRUS SCANNING WITH GPU

Project by: Sinthuja K.
Thipakar S.

Computer Engineering Department,
University of Peradeniya

Motivation


Virus database
Same Instructions
3c6f626a6563742074797065 3d22
2f2f2f2f2f2f2f2f2f2f2f2f2f676f20746f
20746865206c696e6b2062656c6f
Results

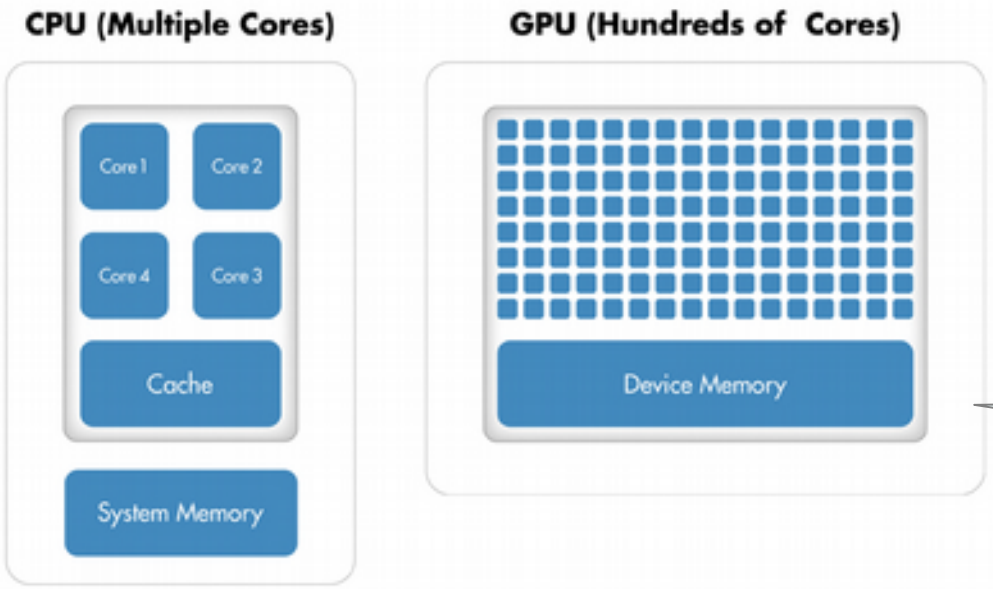

ATTACK VIRUS..?

CPU vs. GPU

Normally scanning in CPU
Speed up scanning through GPU

NVIDIA
CUDA
C/C++

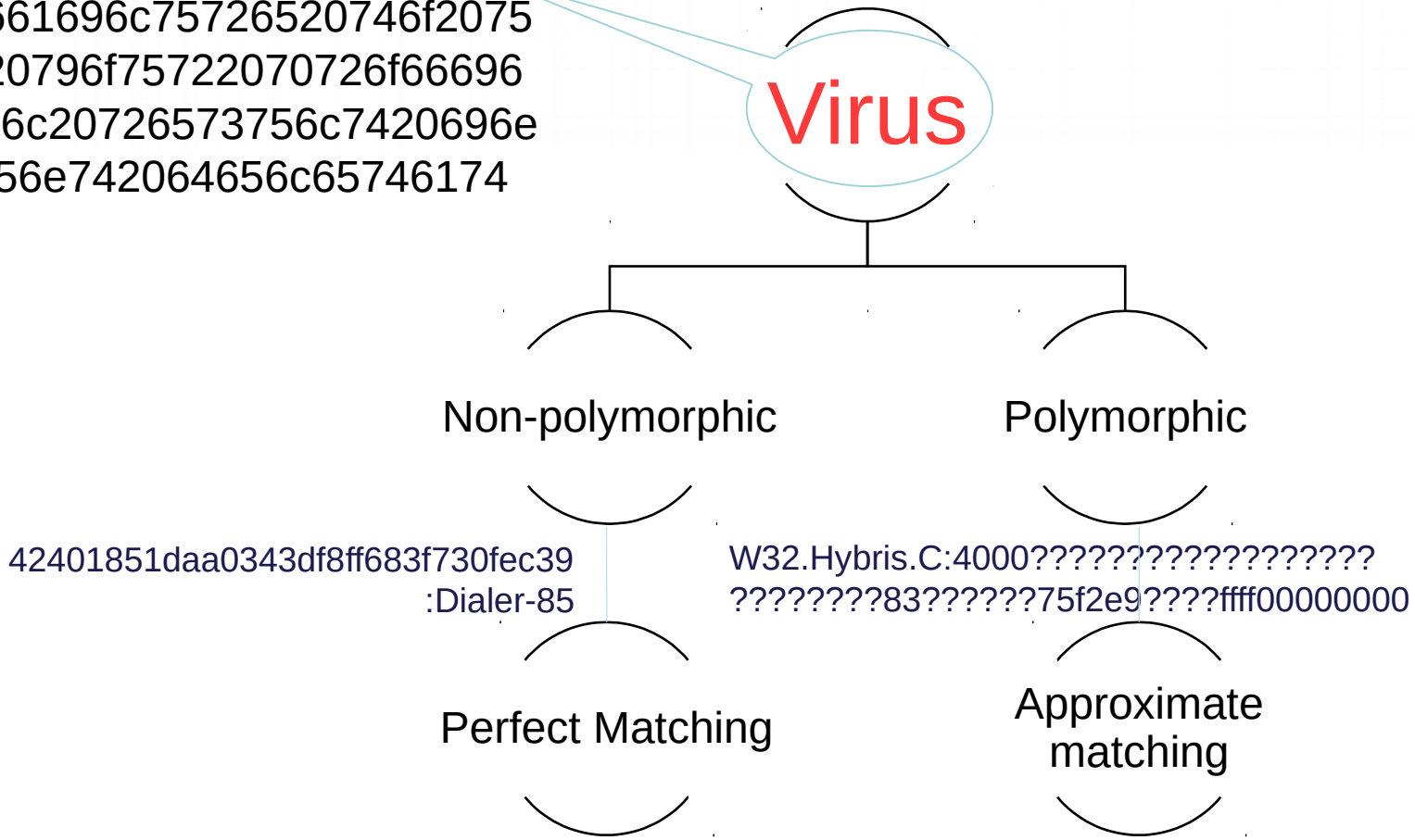
CPU vs GPU



- SIMD Architecture
- Can use for general purpose calculations
- Number of cores is higher than CPU
- Number of threads is higher than CPU

Virus Patterns

3c6f626a6563742074797065 3d222f2f2f
2f2f2f2f2f2f2f2f2f676f20746f2074686520
6c696e6b2062656c6f7720746f2075706
4617465 20796f7572206163636f756e74
20696e6666661696c75726520746f2075
706461746520796f7572207072 6f66696
c652077696c6c20726573756c7420696e
206163636f756e742064656c65746174



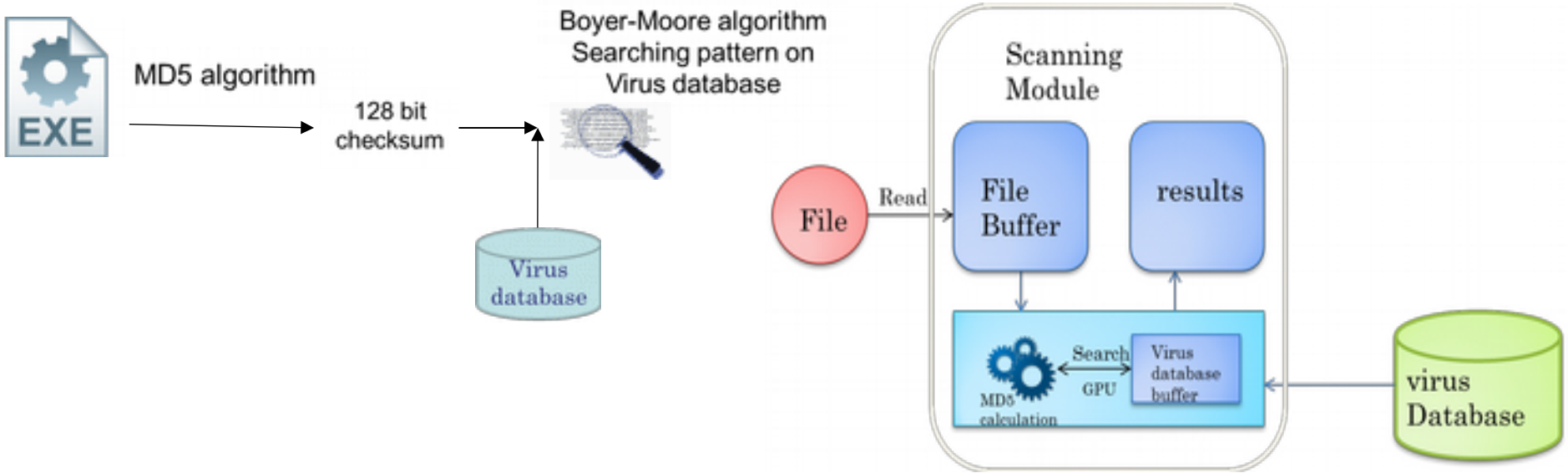
Perfect pattern matching

Non-polymorphic Virus pattern

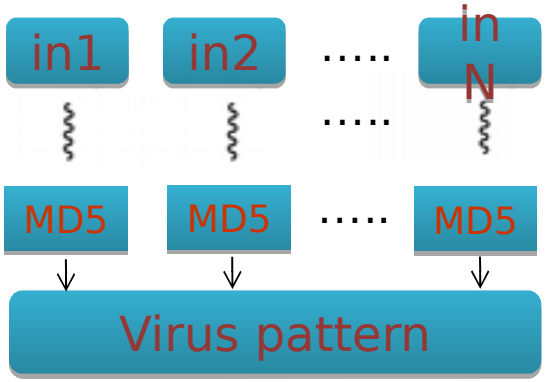
42401851daa0343df8ff683f730fec39

- MD5 Algorithm
- Boyer-Moore Algorithm

Fast string searching Algorithm



Parallelism for Boyer-Moore

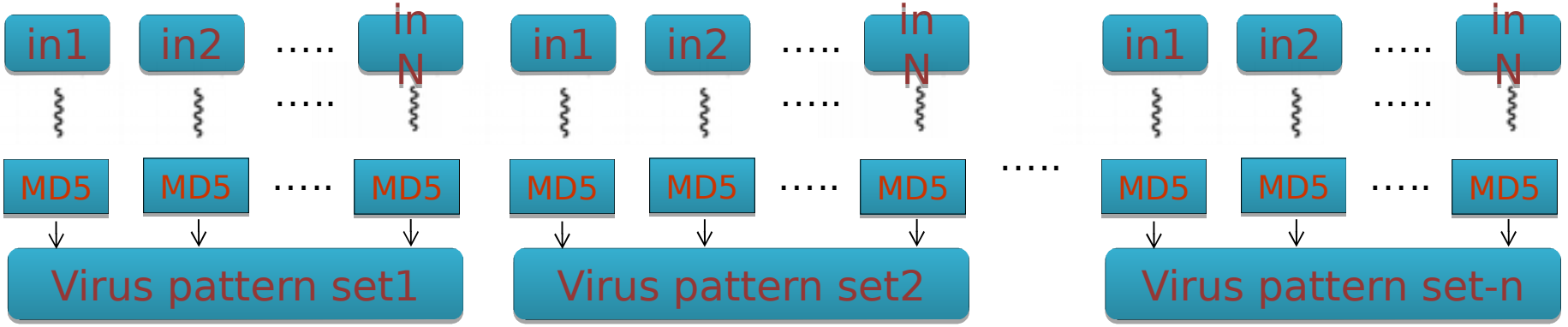


In1 - Input 1

- Thread

MD5 - MD5 Calculation

Method1: Parallelism with number of inputs



Method2: Parallelism with number of inputs + divided pattern sets

Approximate pattern matching

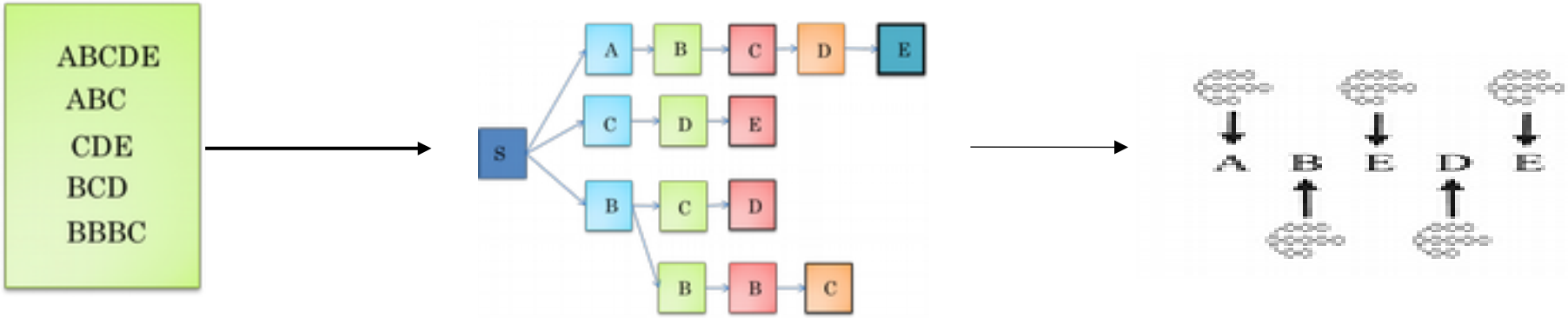
Polymorphic Virus pattern

4000??????????????????????????
83???????75f2e9????ffff00000000

- Aho-Corasick algorithm
- PFAC Library

Aho-Corasick algorithm implemented for GPU

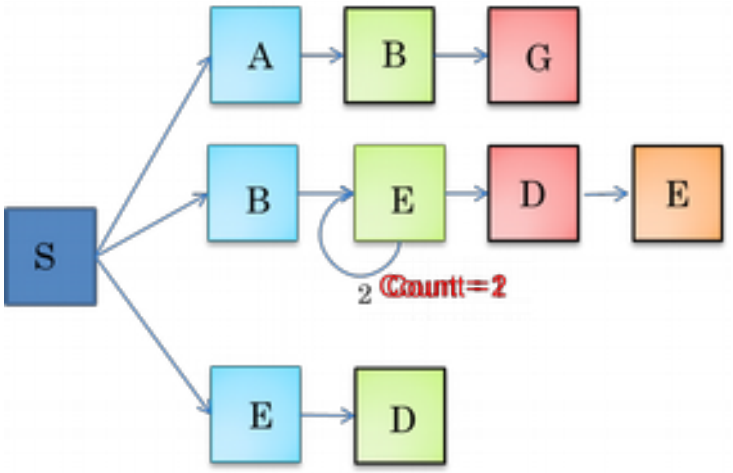
Normally how PFAC works:



Approximate pattern matching

?	Fixed number of Wild character
{-n}	0 – n number of wild characters
{m-n}	m – n number of wild characters

AB
ABG
ED
BE??DE



B E P Q D E

Conclusion

- Implemented Boyer-Moore & MD5 algorithm in CPU & GPU
- Changed PFAC implementation from perfect p attern matching to approximate pattern matchi ng
- Performance analysis for both algorithms is in progress

Thank You