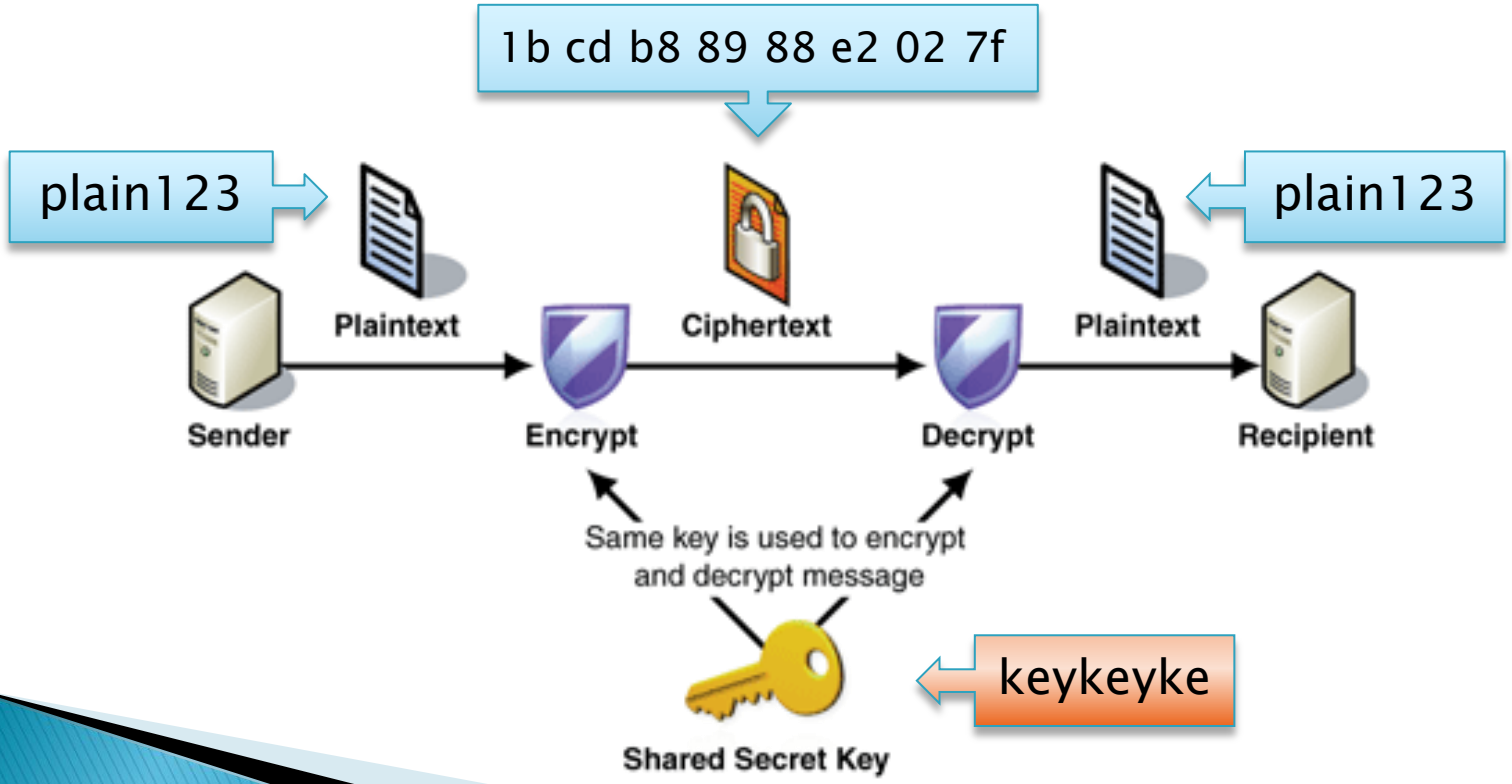
Bruteforcing DES

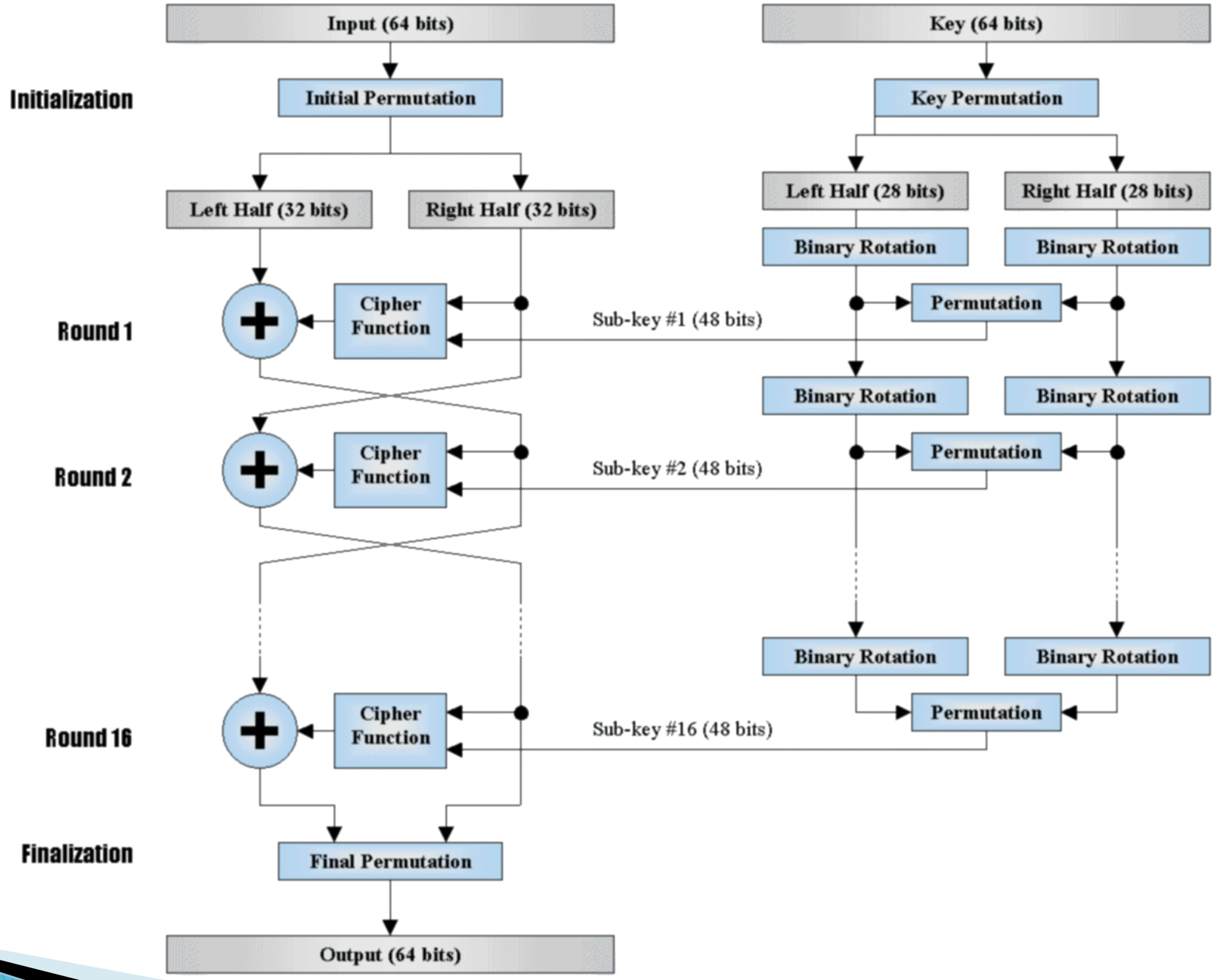
with CUDA on a GPU

What is DES?

- DES = Data Encryption Standard
- Symmetric Encryption Algorithm by IBM (1975)

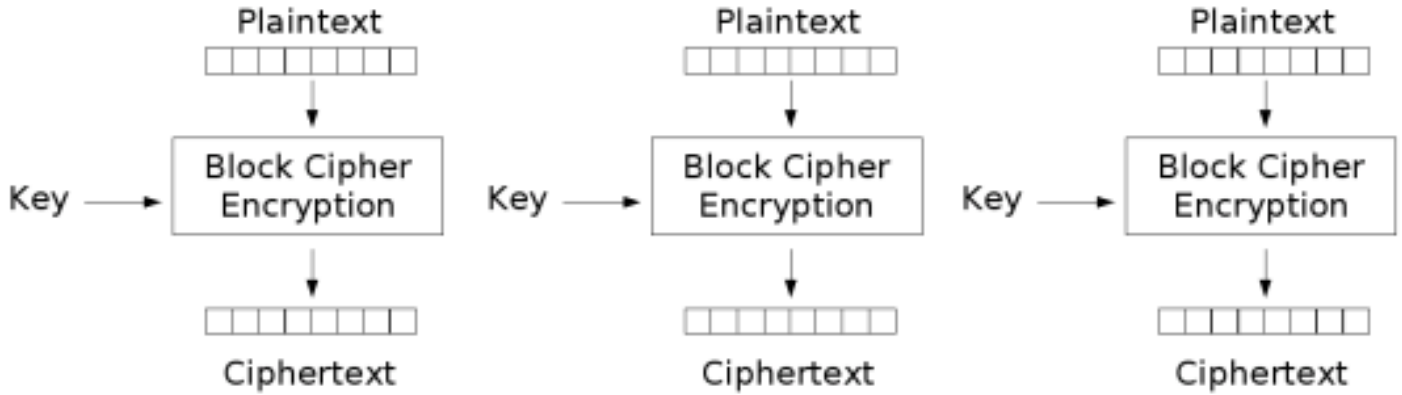


How does it work ?



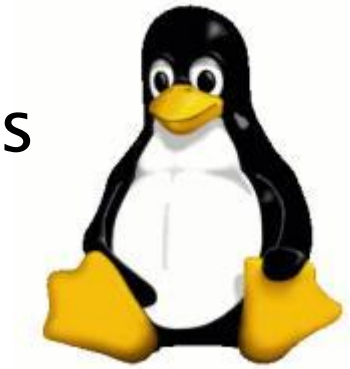
Modes of operation

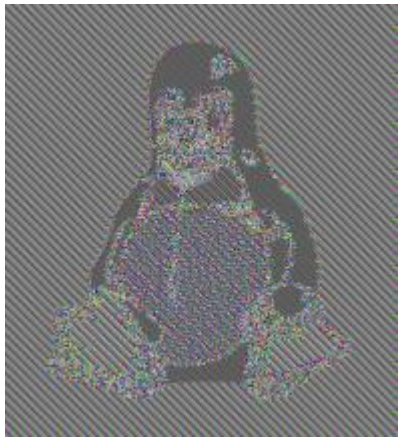
- ECB (Electronic Code Book)



Electronic Codebook (ECB) mode encryption

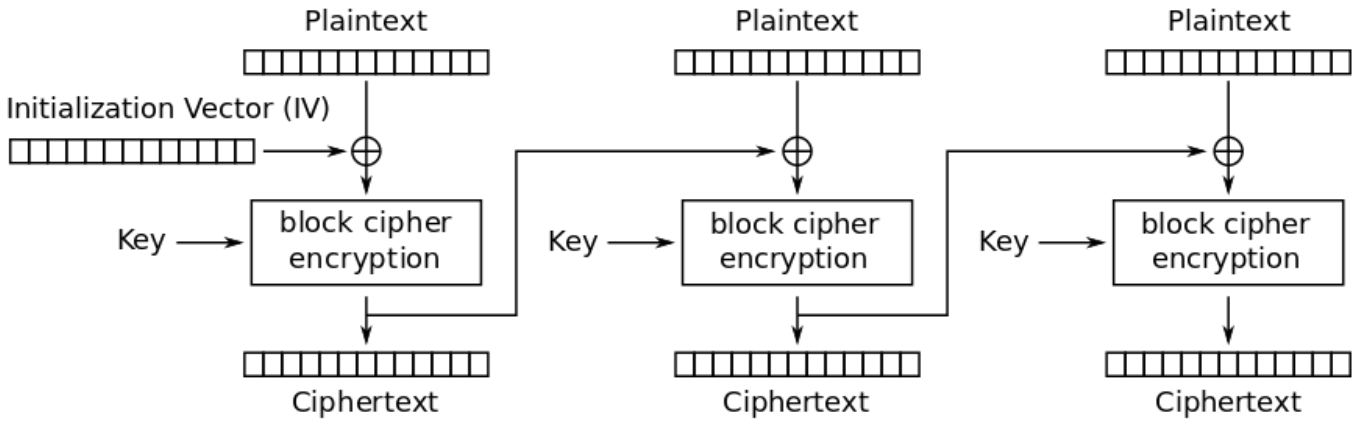
- Blocks can be encrypted in parallel
- Weak encryption results





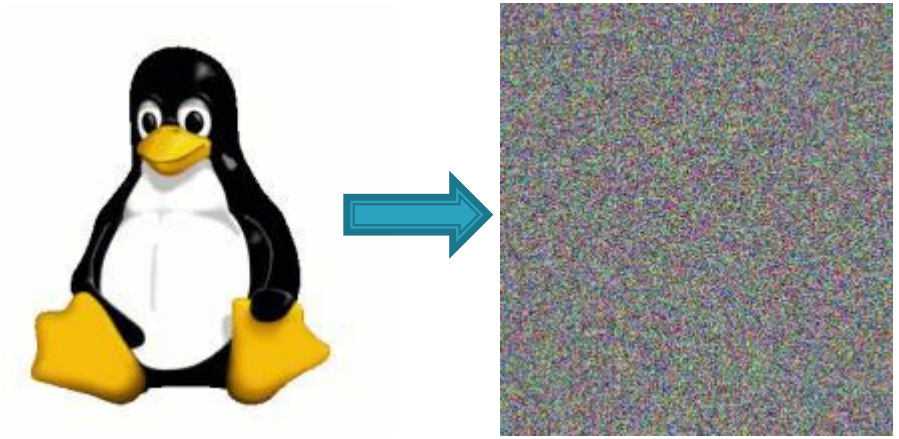
Modes of operation

- CBC (Cipher Block Chaining)



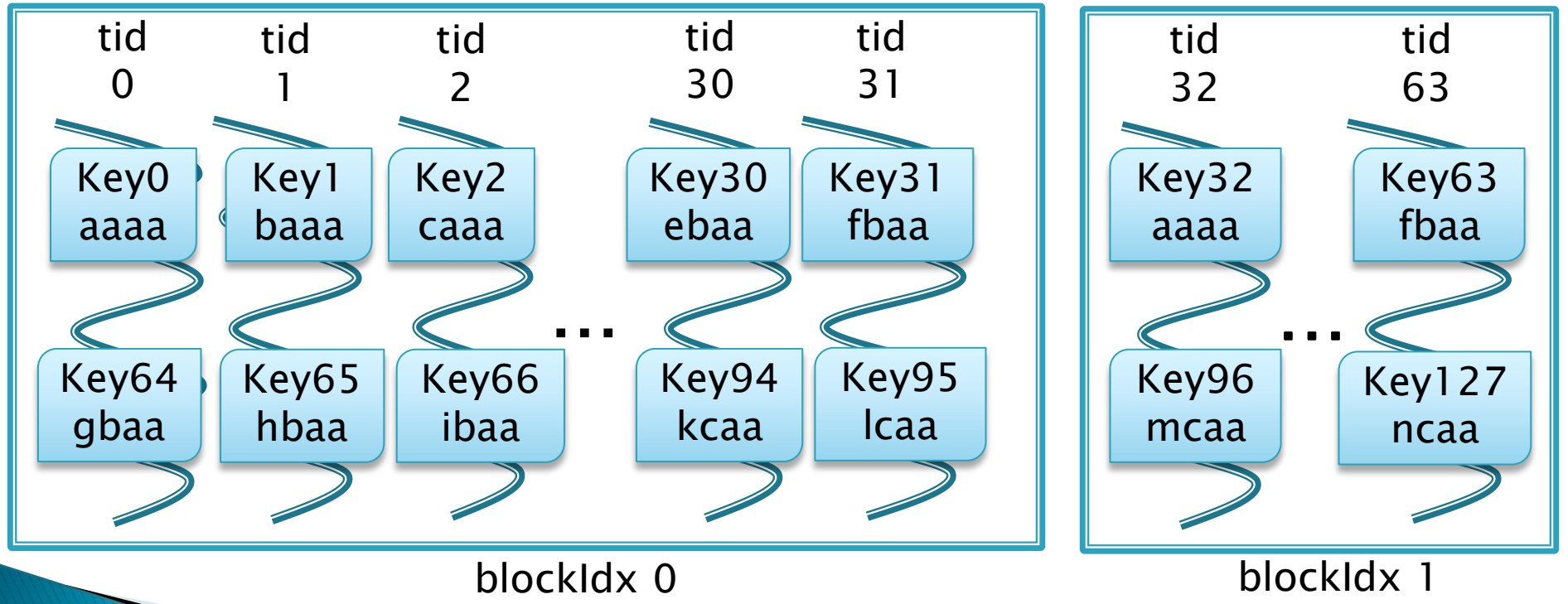
Cipher Block Chaining (CBC) mode encryption

- Better encryption results
- Not parallelizable



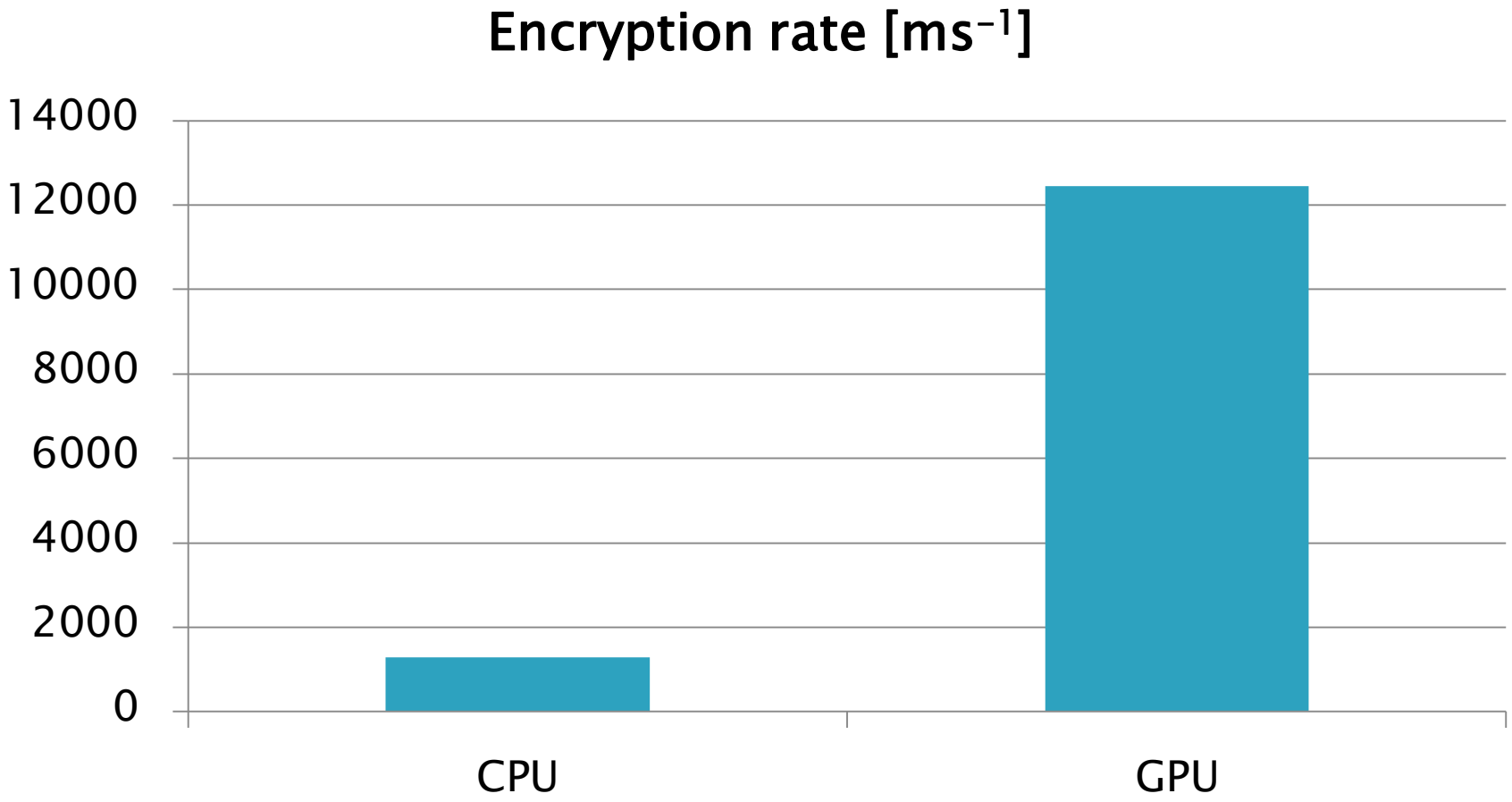
How to bruteforce

- CUDA kernels try keys k_i
- Until `cipher` == enc_k(`plain`)

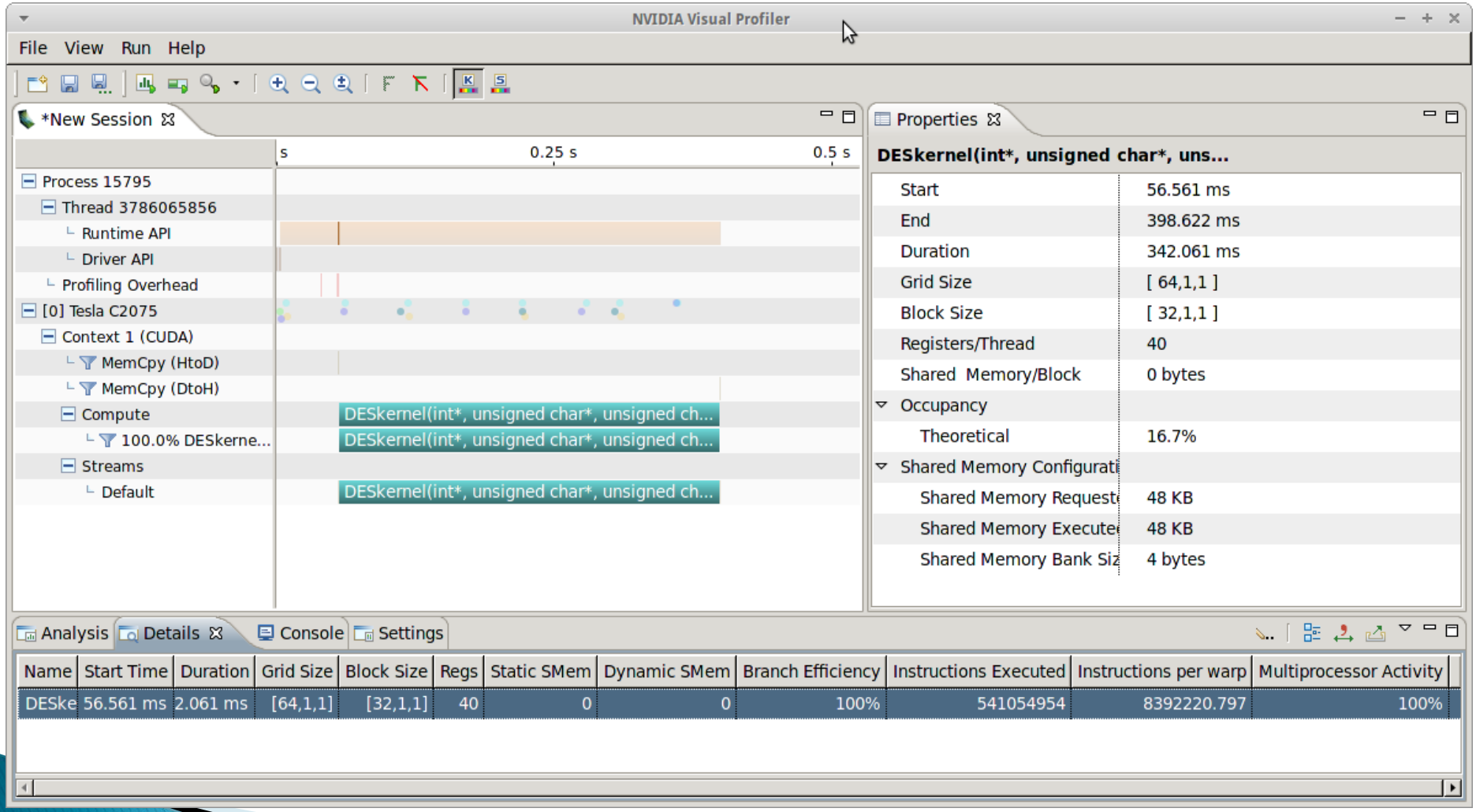


Each thread tries a different key
and jumps by blockDim.x * gridDim.x

Performance



Profile


NVIDIA Visual Profiler
File View Run Help
*New Session
s
0.25 s
0.5 s
Process 15795
Thread 3786065856
Runtime API
Driver API
Profiling Overhead
[0] Tesla C2075
Context 1 (CUDA)
MemCpy (HtoD)
MemCpy (DtoH)
Compute
100.0% DESkerne...
Streams
Default
DESkernel(int*, unsigned char*, unsigned ch...
DESkernel(int*, unsigned char*, unsigned ch...
DESkernel(int*, unsigned char*, unsigned ch...
Properties
DESkernel(int*, unsigned char*, uns...
Start 56.561 ms
End 398.622 ms
Duration 342.061 ms
Grid Size [64,1,1]
Block Size [32,1,1]
Registers/Thread 40
Shared Memory/Block 0 bytes
Occupancy
Theoretical 16.7%
Shared Memory Configurati
Shared Memory Request 48 KB
Shared Memory Execute 48 KB
Shared Memory Bank Siz 4 bytes
Analysis
Details
Console
Settings
Name Start Time Duration Grid Size Block Size Regs Static SMem Dynamic SMem Branch Efficiency Instructions Executed Instructions per warp Multiprocessor Activity
DESke 56.561 ms 2.061 ms [64,1,1] [32,1,1] 40 0 0 100% 541054954 8392220.797 100%

Outlook

- Optimize parallel code
- Analyze best combination of threads per block and blocks per grid. (I used 64,32)
- Utilize multiple cards
- Use found weakness of DES
  - Keys with distance 1 per byte yield same cipher.
  - Key keykeyke == jdxjdxjd == jdxkeyje

Thank you for your attention!

Questions?