

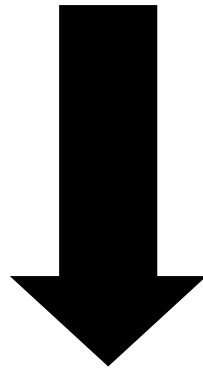
SymEngine

Symbolic Execution of OpenCL Kernels

Alberto Magni



Optimize code for GPUs

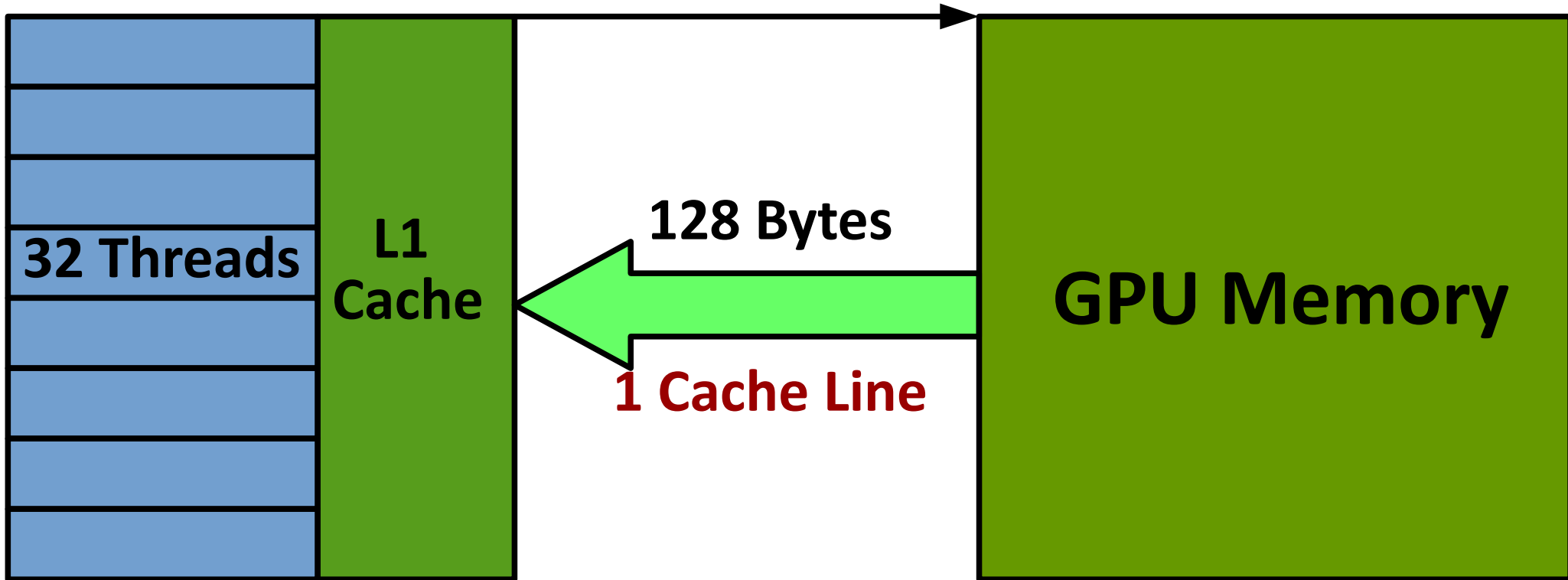


Optimize Memory Accesses

GPU Memory Transactions

Coalesced Access

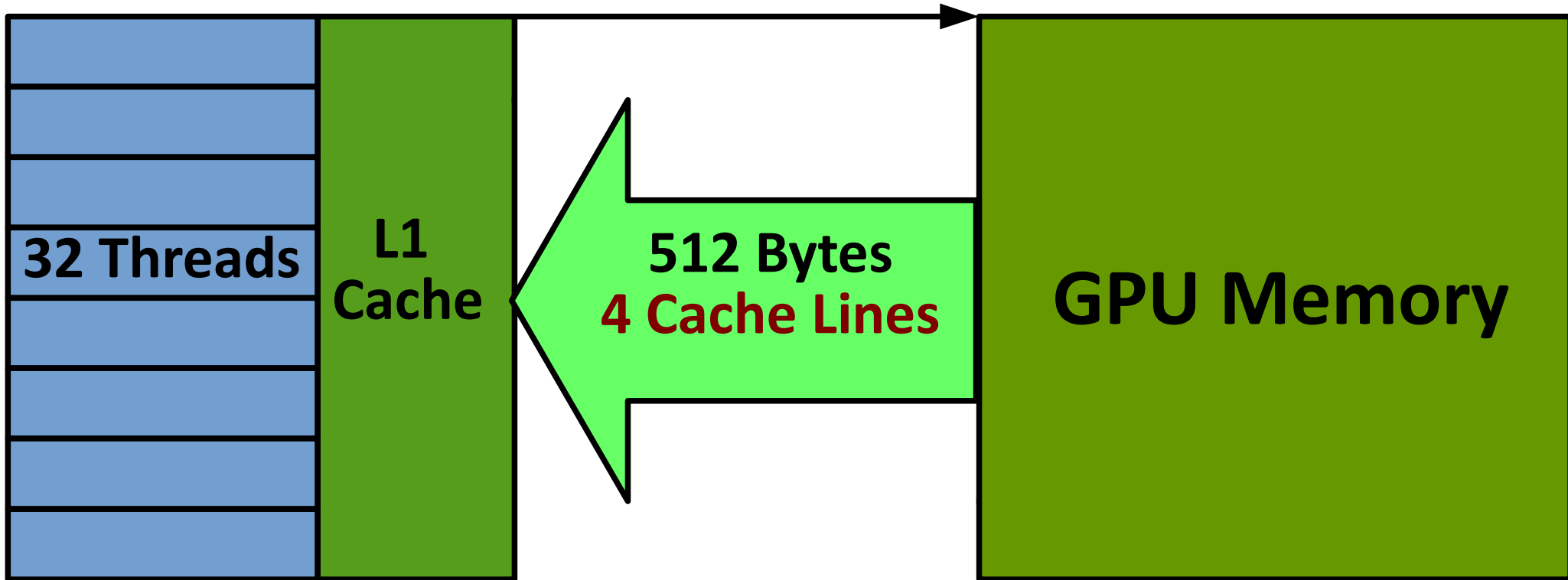
GPU Core 1 Load Request = 4 Bytes per Thread



GPU Memory Transactions

UnCoalesced Access

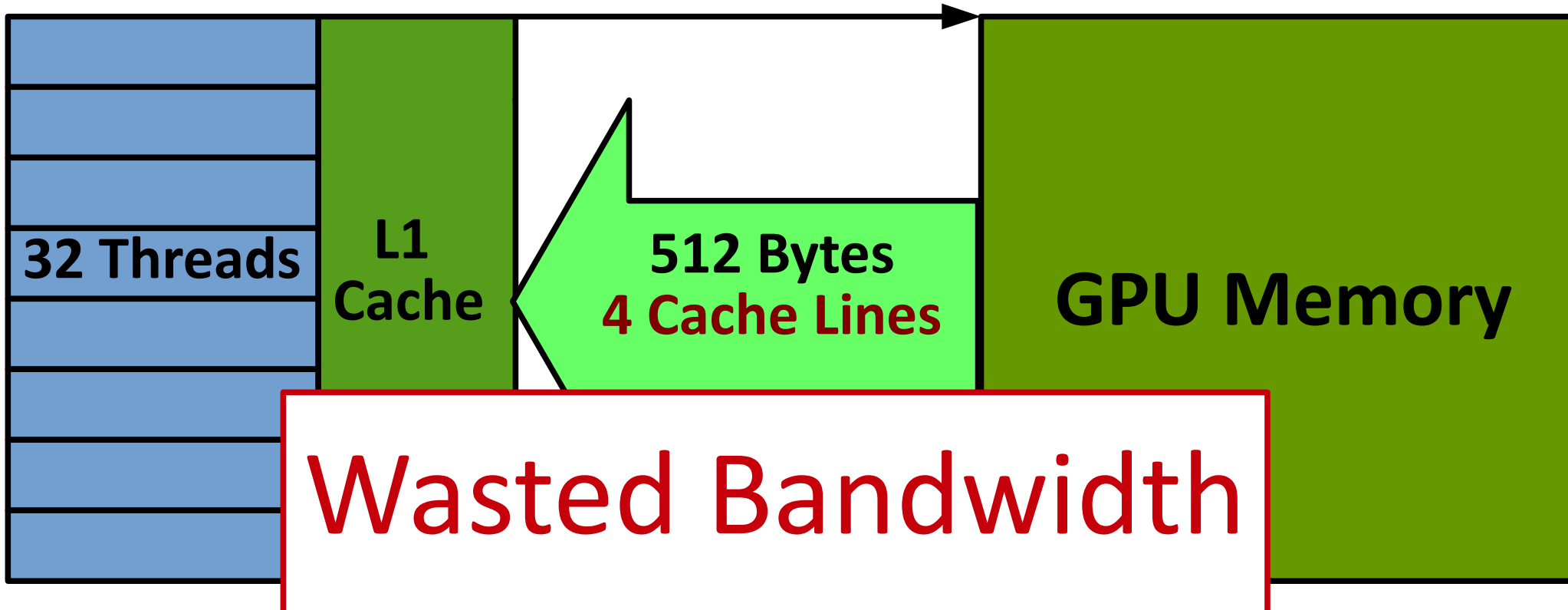
GPU Core 1 Load Request = 4 Bytes per Thread



GPU Memory Transactions

UnCoalesced Access

GPU Core 1 Load Request = 4 Bytes per Thread



SymEngine

Statically Detect Suboptimal Accesses to Memory

SymEngine

Statically Detect Suboptimal Accesses to Memory

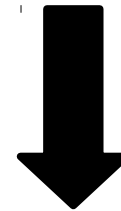
OpenCL Kernel

```
int threadID = get_global_id(0);
sX = x[threadID];
sY = y[threadID];
sZ = z[threadID];
sQr = Qr[threadID];
sQi = Qi[threadID];

for (int kIndex = 0; (kIndex < KERNEL_ELEMS_PER_GRID); kIndex ++,
kGlobalIndex ++) {
    float expArg = PIx2 * (ck[kIndex].Kx * sX + ck[kIndex].Ky * sY +
ck[kIndex].Kz * sZ);
    sQr += ck[kIndex].PhiMag * cos(expArg);
    sQi += ck[kIndex].PhiMag * sin(expArg);
}

Qr[threadID] = sQr;
Qi[threadID] = sQi;
```

Resolve Address



**Compute
Number of Transactions**

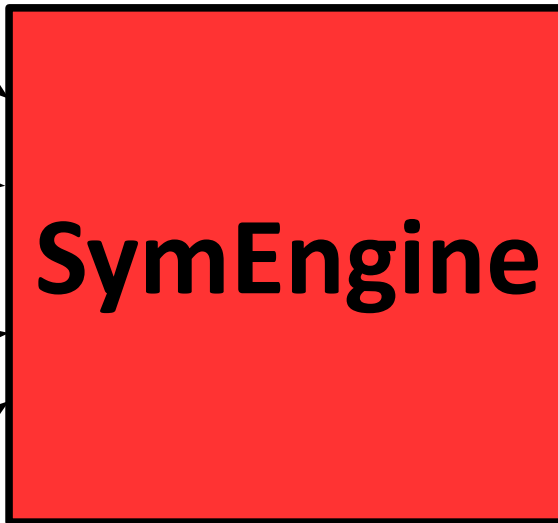
Symbolic Execution

OpenCL Code

Warp-Id

Number of
Threads

Input Values

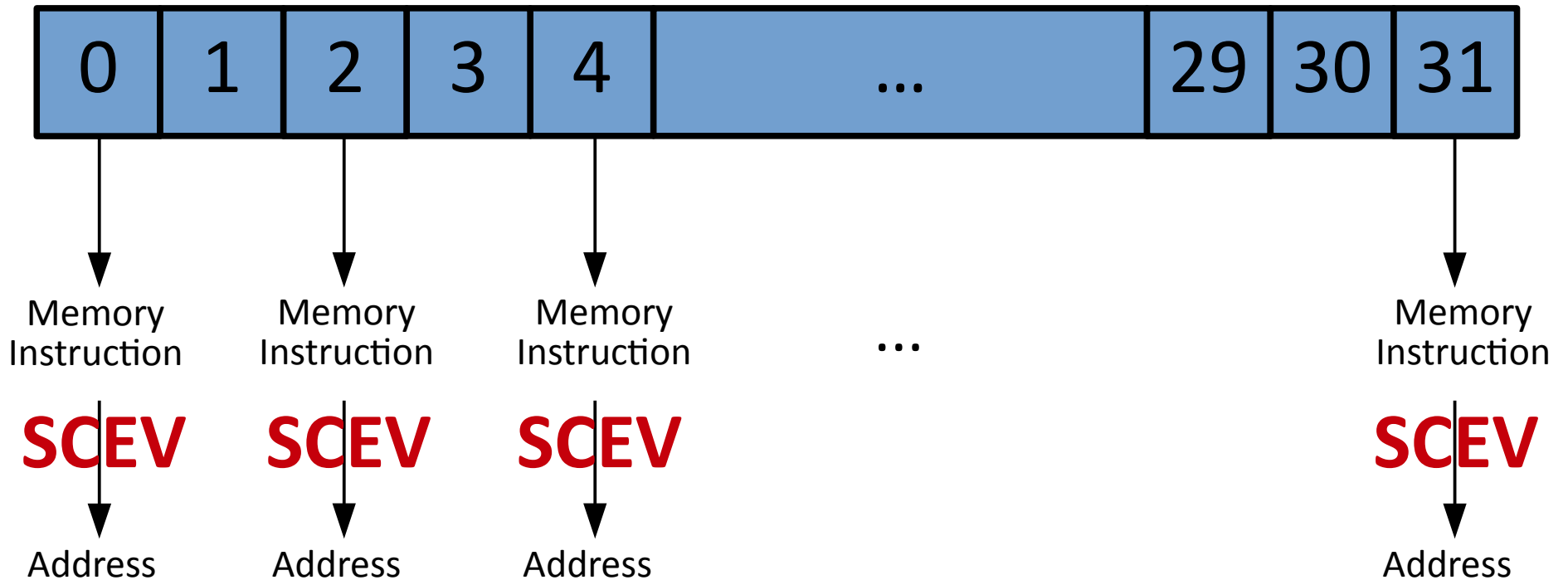


SymEngine

Hardware
Memory
Transactions

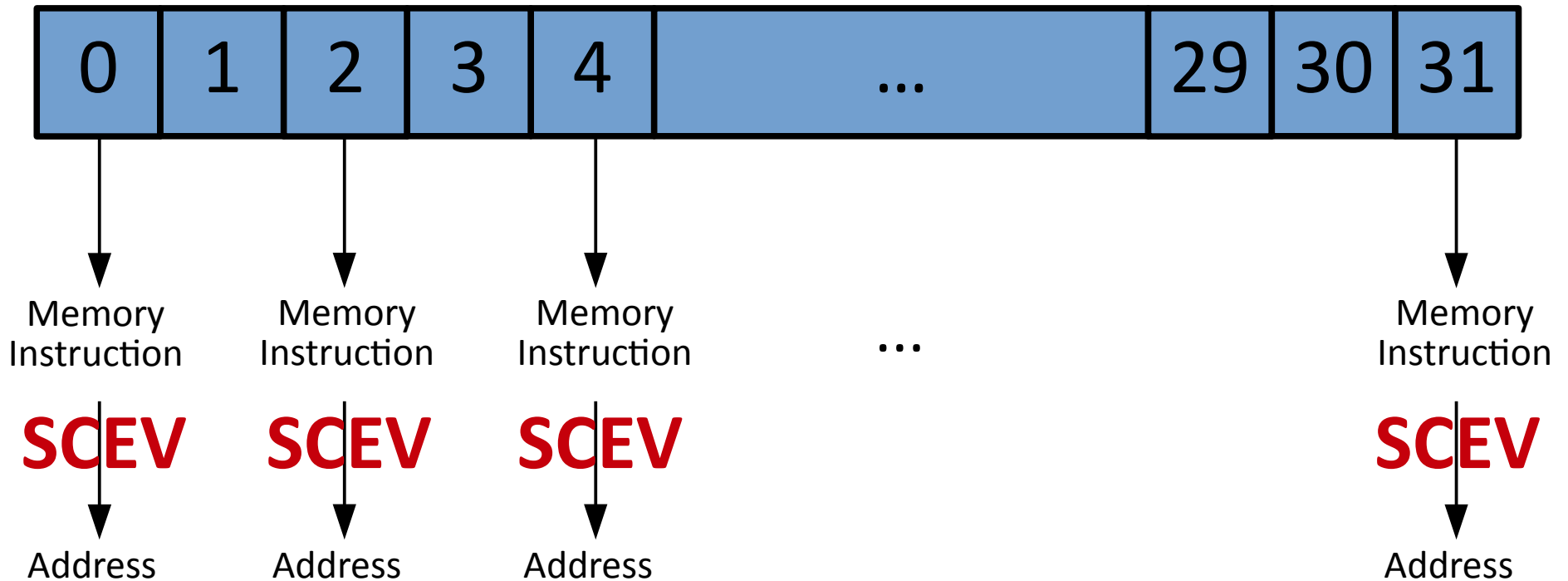
Symbolic Execution

Threads in a Warp



Symbolic Execution

Threads in a Warp



Number of Cache
lines touched

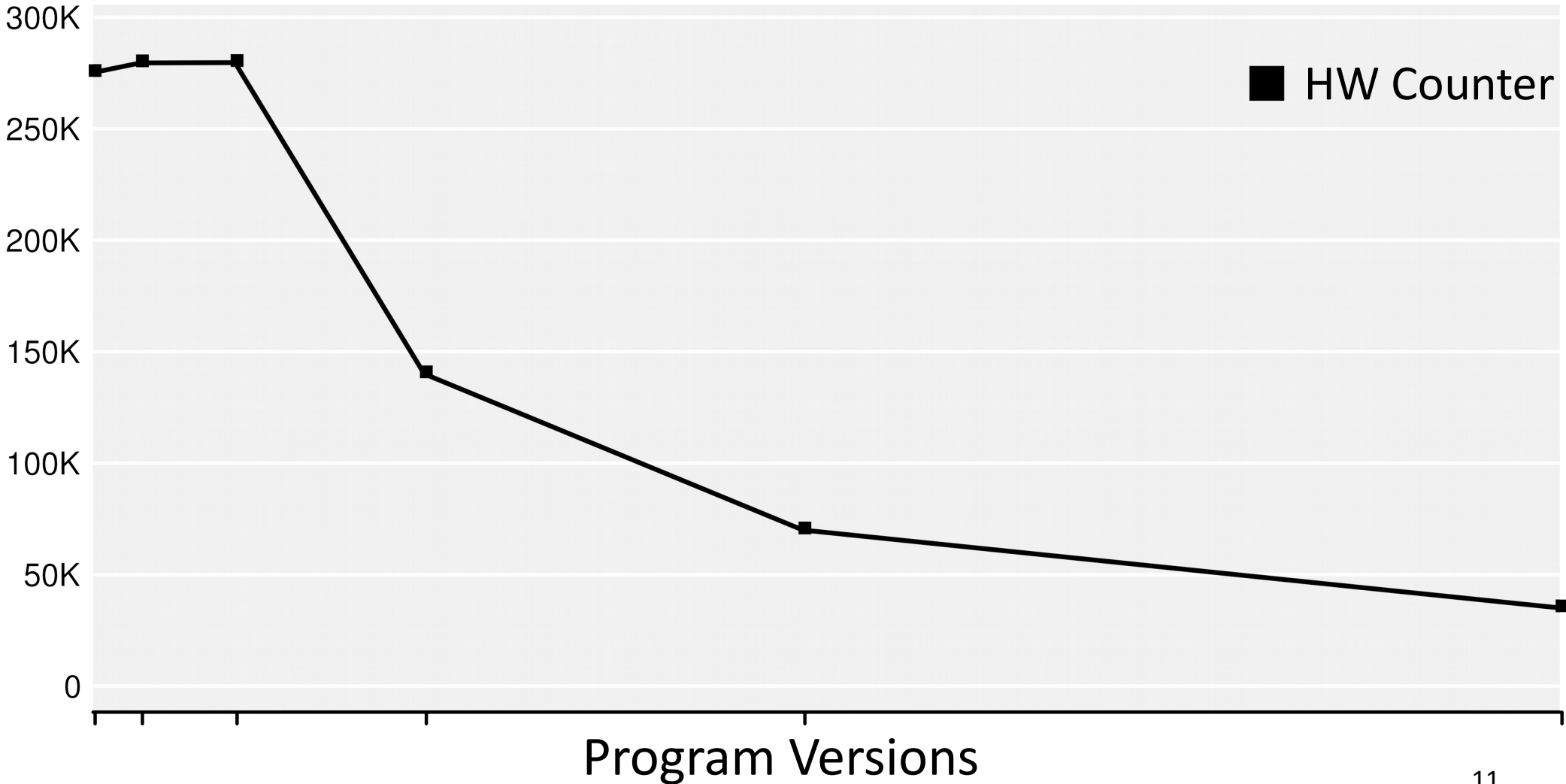


Transaction
Number

Validation – Nvidia GTX480

Against Hardware Performance counters

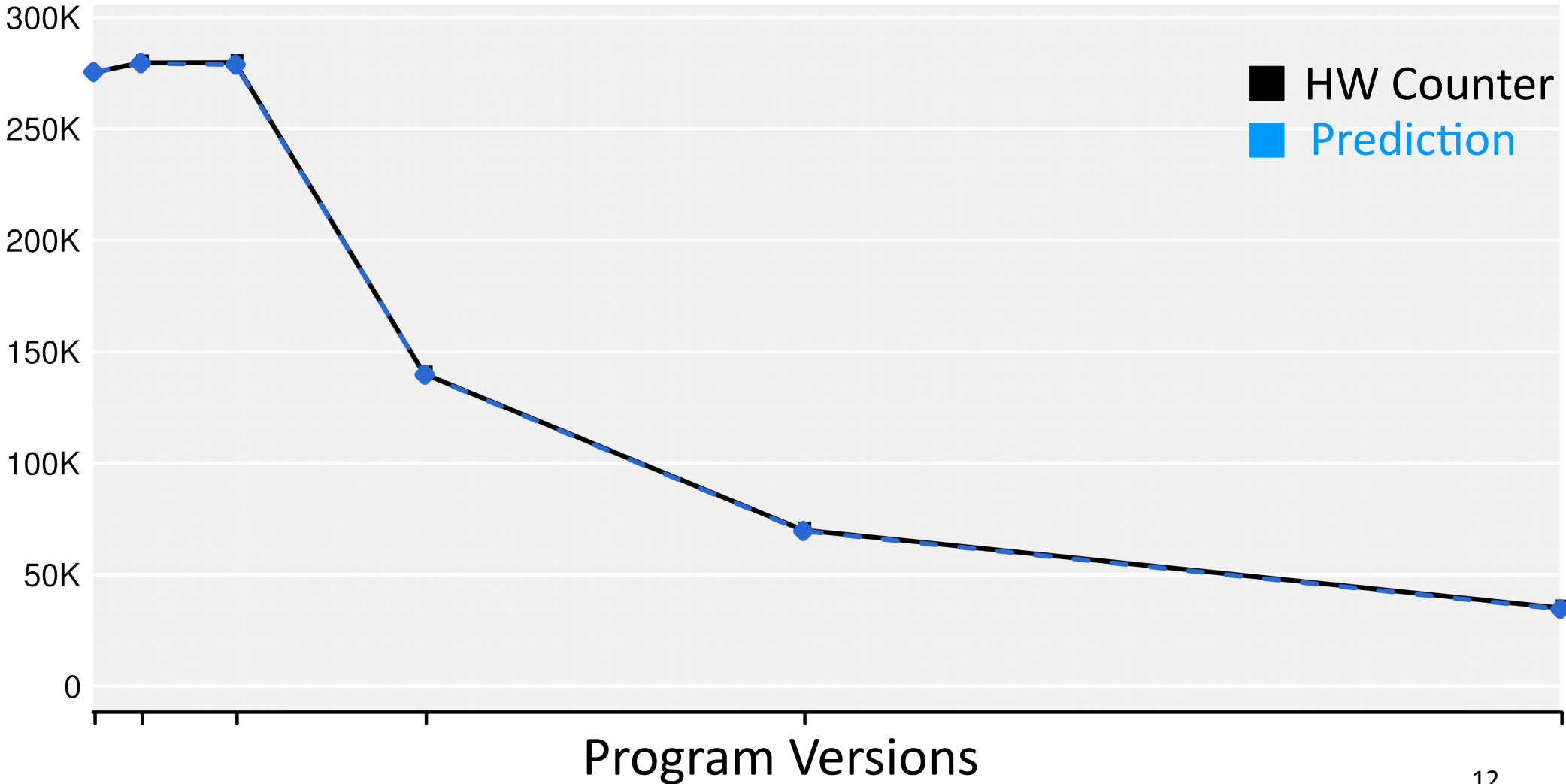
Total HW Transactions for Black-Scholes



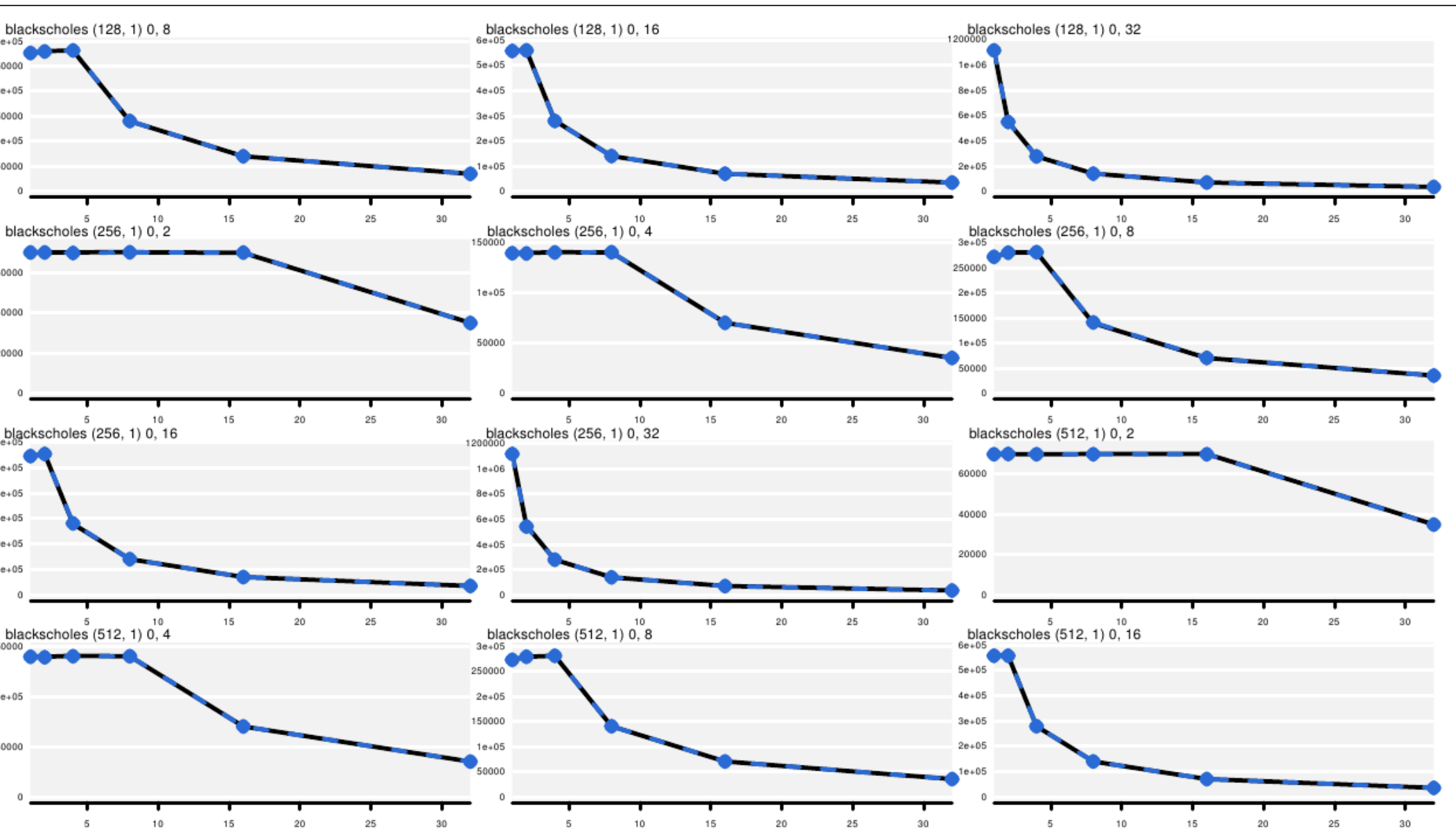
Validation – Nvidia GTX480

Against Hardware Performance counters

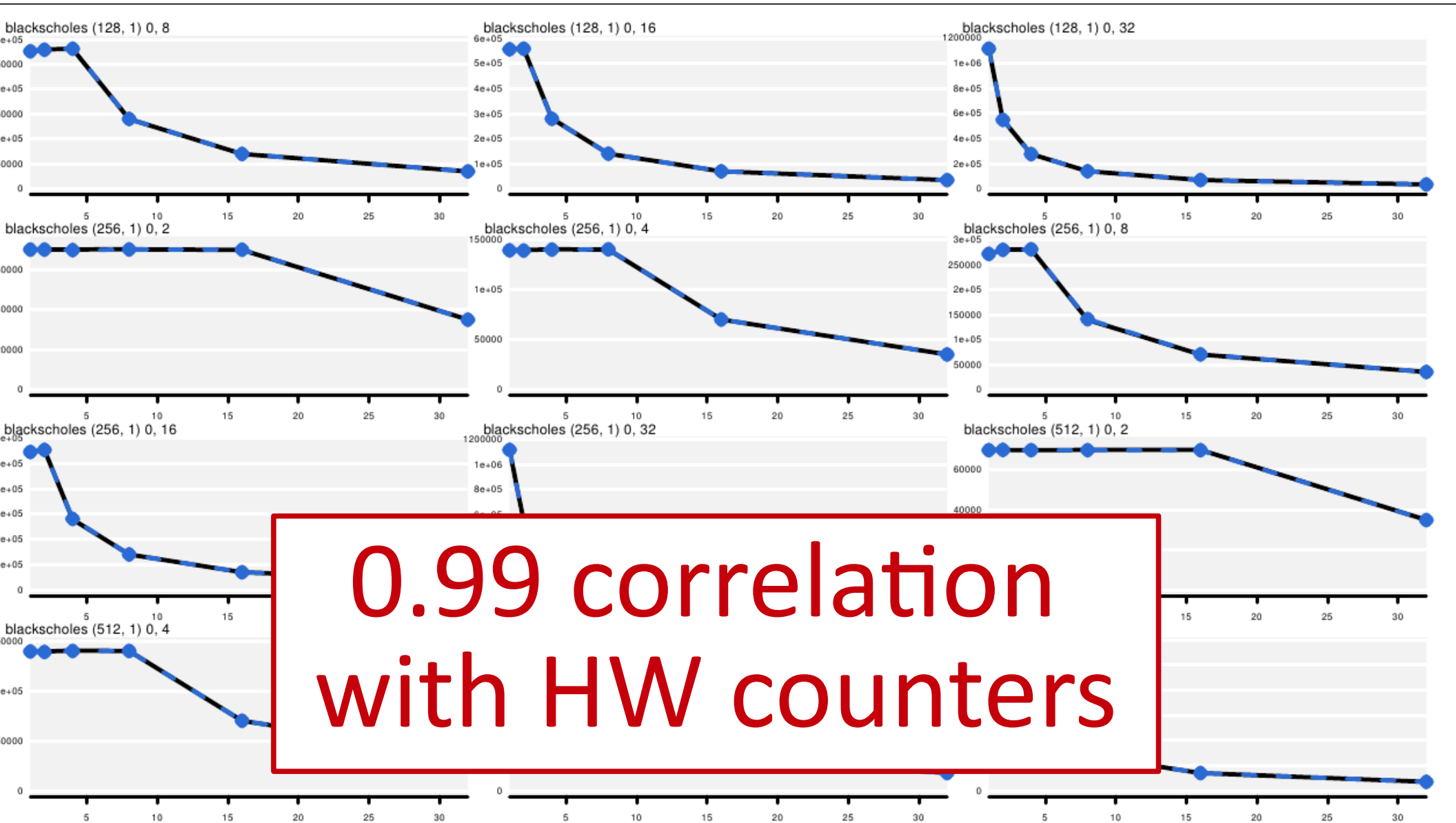
Total HW Transactions for Black-Scholes



Validation – Nvidia GTX480



Validation – Nvidia GTX480



It's on GitHub!

<http://github.com/HariSeldon/SymEngine>