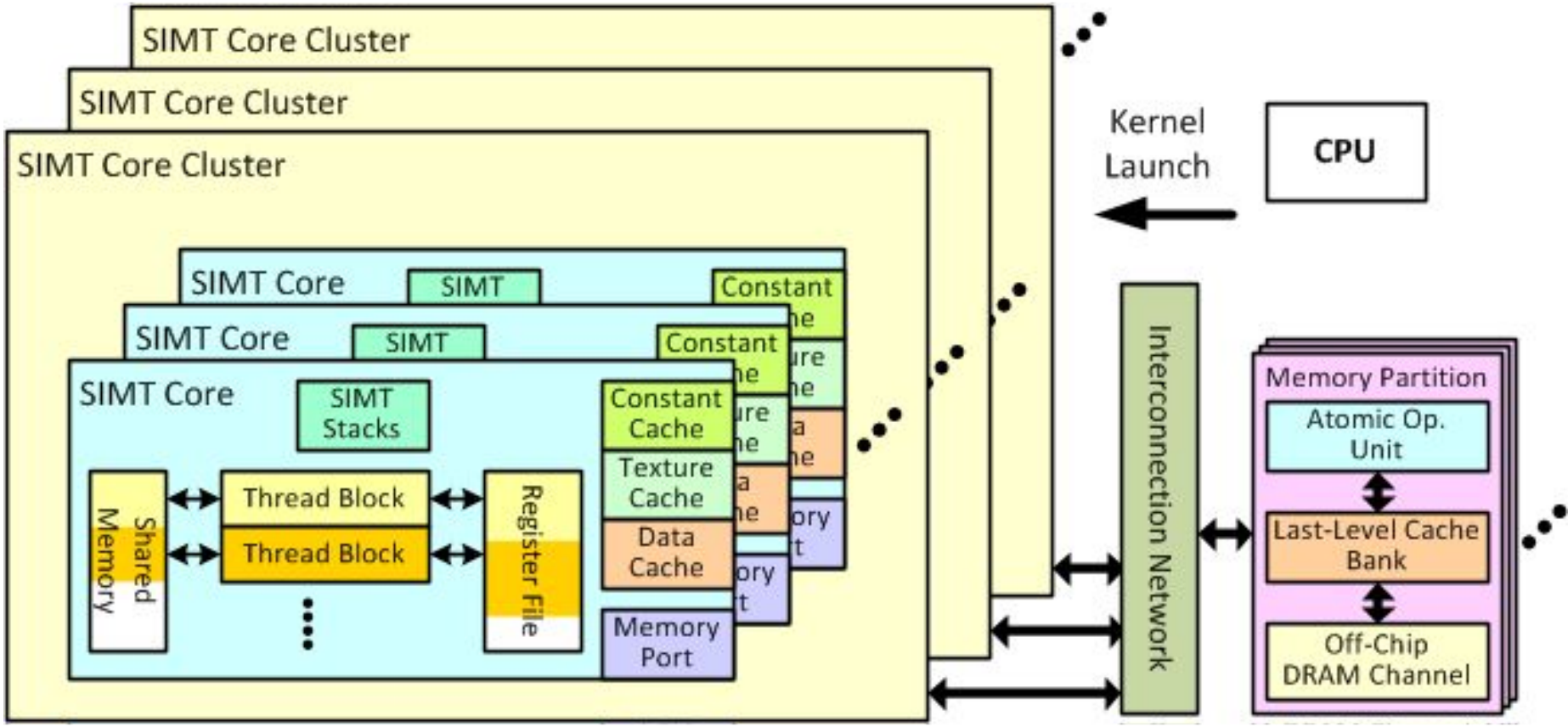
Characterizing Performance loss from mapping general purpose applications onto GPU architecture

Archit Gupta, Sohum Datta

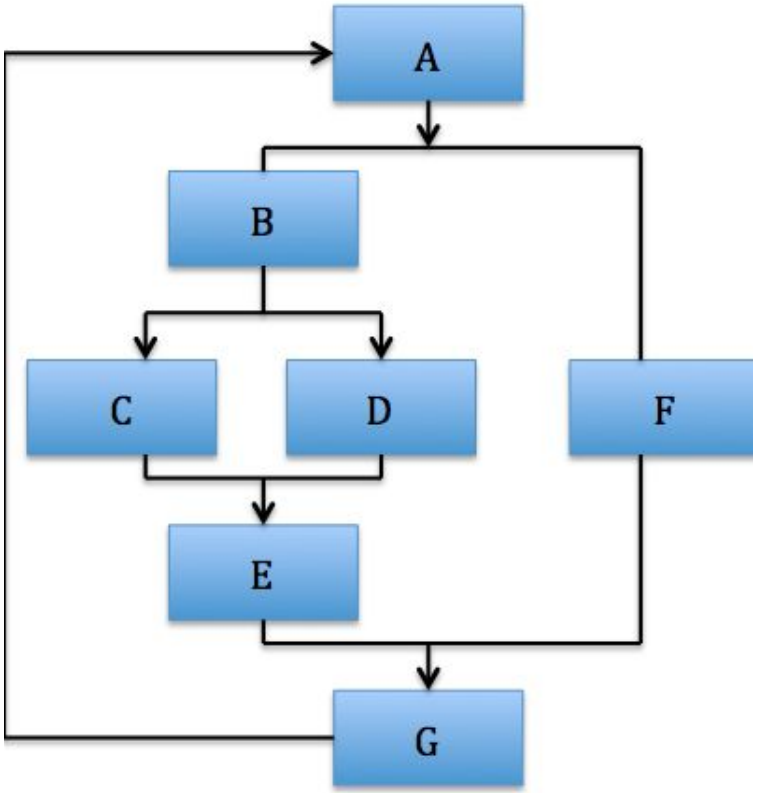
CONTROL DIVERGENCE

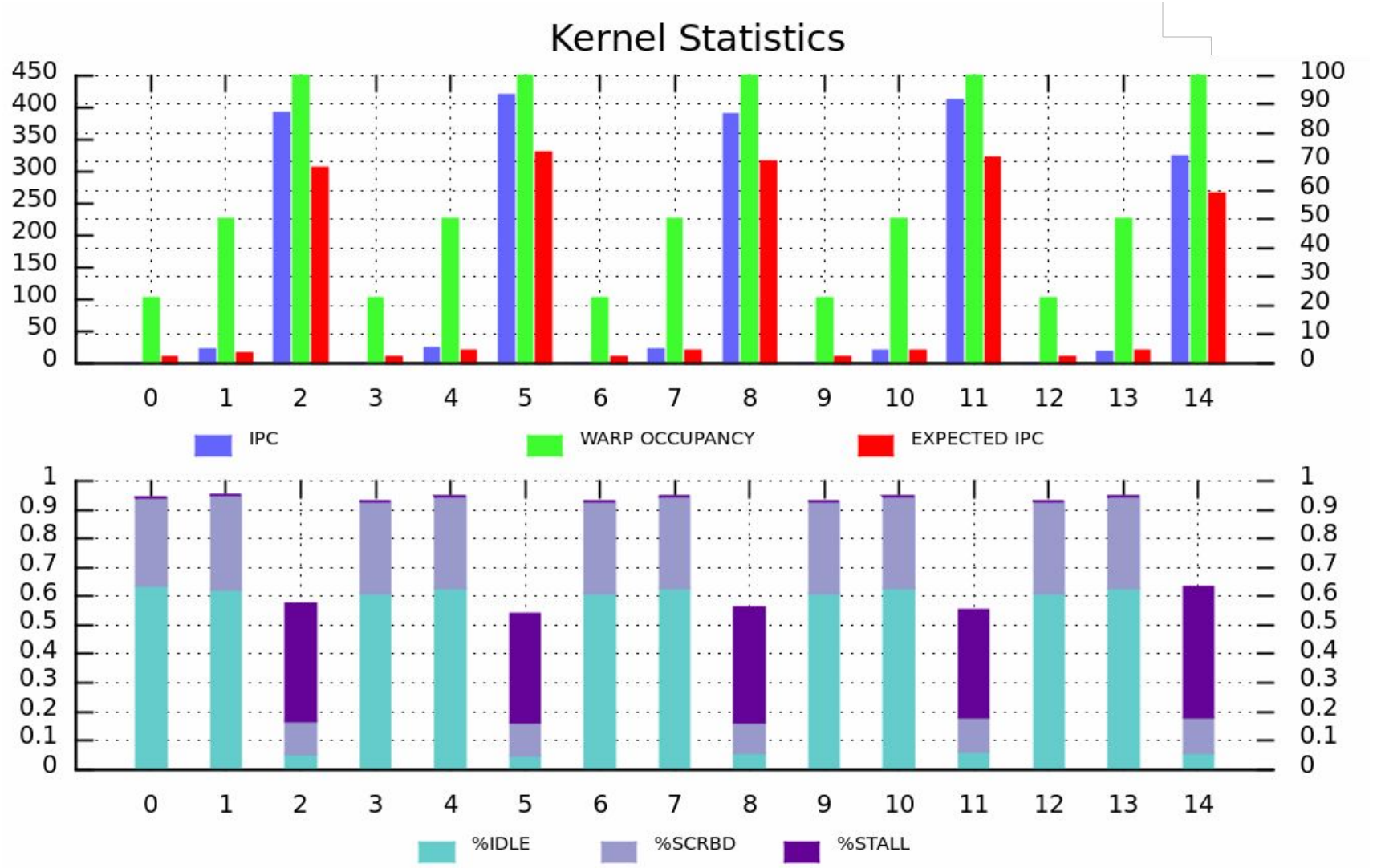
GPU ARCHITECTURE



**Architecture of GPGPU-SIM (developed by Tor Aamodt at University of British Columbia)

- SIMD architecture - Originates from vector supercomputers
  - Mainly oriented towards graphics processing where a single set of operations need to be applied on a massive set of data. Eg: Adjusting screen brightness
- Some of the characteristics of a modern day GPU
  - Bare minimum execution pipelines (in-order, w/o branch prediction, lock-step execution of multiple threads etc)
  - Large number of lightweight cores (SMs) with large register files, capable of switching context every cycle
- Lock-step execution of thread groups (called warps) leads to control hazards, referred to as branch divergence





Branch divergence consumes a significant portion of the GPU execution time

**Analyzing GPU workloads for control dependent inefficiencies in the GPGPU architecture - Nehal Bhandari, Archit Gupta

CHARACTERIZING BRANCHES

INTRINSIC AND EXTRINSIC BRANCHES

```
__global__ void
BFS_kernel( ... )
{
  if(threadIdx.x < NUM_BIN){
    q.reset(threadIdx.x, blockDim);
  }
  __syncthreads();
  tid = blockIdx.x*THREADS_PER_BLOCK
                ... + threadIdx.x;
  if( tid < no_of_nodes)
  {
    visit_node(q1[tid], threadIdx.x
        & MOD_OP, q, overflow,
        g_color, g_cost, gray_shade);
  }
  __syncthreads();
  if(threadIdx.x == 0){
    ...
    shift = atomicAdd(tail,tot_sum);
  }
  __syncthreads();
}
```

Which branches are intrinsic and which are extrinsic?

PARBOIL benchmark suite - Breadth First Search

Branches can be *intrinsic* to an algorithm

Intrinsic Branches → Extrinsic Branches → Performance Impact

CPU codes gives us the intrinsic branches

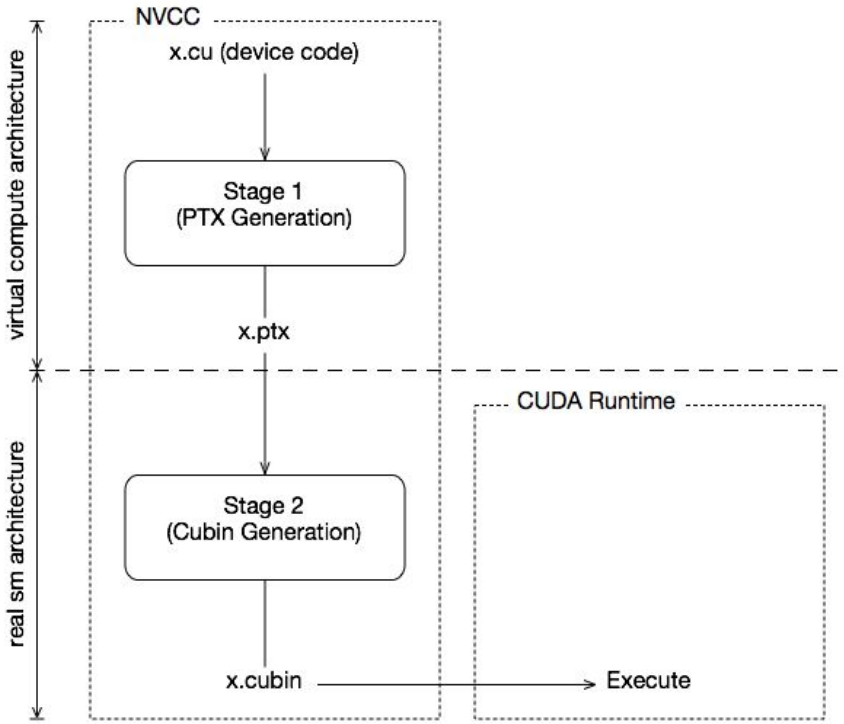
Intrinsic branches in CPU code is related to the CUDA code to point out the intrinsic control. Any other branches/controls present in the CUDA code are extrinsic.

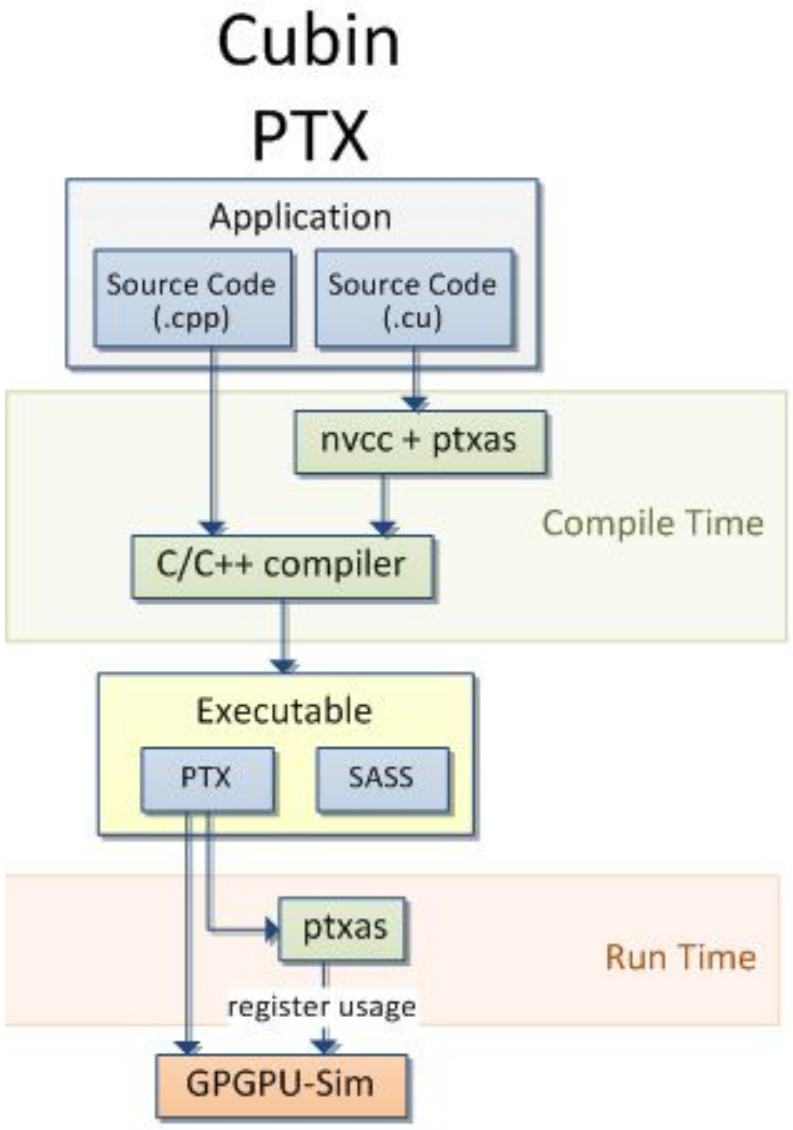
Locate the extrinsic branches in the GPU code

Some branches are the result of mapping an algorithm onto a GPU, called extrinsic branches. They arise because a general purpose program is run an a GPU.

Map branches in simulated binaries to High-level code

Extrinsic branches in CUDA code is traced to compiled code and is instrumented to measure performance loss due to their divergence.





**NVIDIA CUDA Compiler NVCC v7.5 Documentation (http://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc)
****CUDA-sim Functional Simulation Engine, GPGPU-sim v3.x