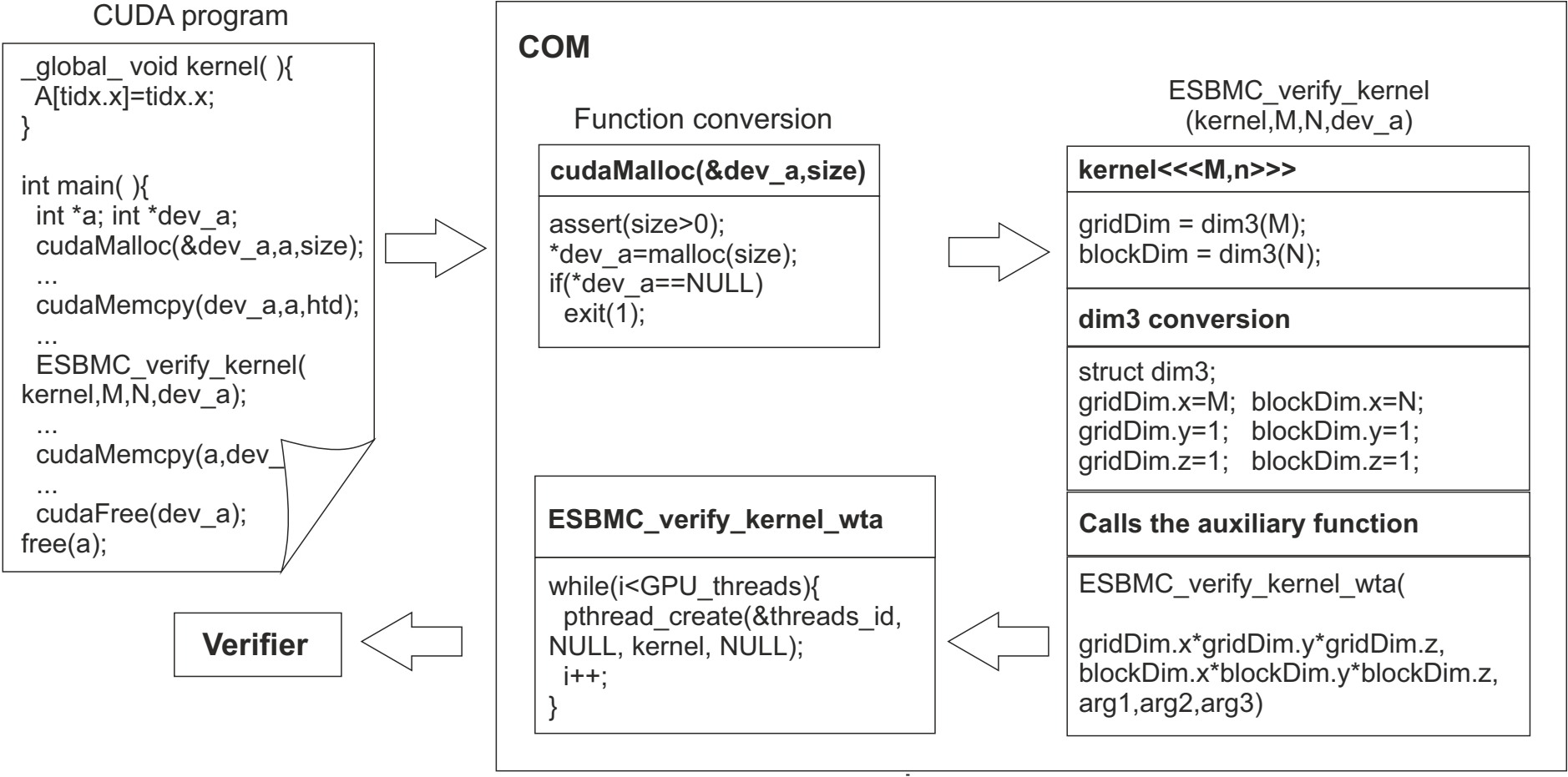
CUDA program
global void kernel( ){
A[tidx.x]=tidx.x;
}
int main( ){
int *a; int *dev_a;
cudaMalloc(&dev_a,a,size);
...
cudaMemcpy(dev_a,a,htd);
...
ESBMC_verify_kernel(
kernel,M,N,dev_a);
...
cudaMemcpy(a,dev_
...
cudaFree(dev_a);
free(a);
Verifier
COM
Function conversion
cudaMalloc(&dev_a,size)
assert(size>0);
*dev_a=malloc(size);
if(*dev_a==NULL)
exit(1);
ESBMC_verify_kernel
(kernel,M,N,dev_a)
kernel<<<M,n>>>
gridDim = dim3(M);
blockDim = dim3(N);
dim3 conversion
struct dim3;
gridDim.x=M; blockDim.x=N;
gridDim.y=1; blockDim.y=1;
gridDim.z=1; blockDim.z=1;
Calls the auxiliary function
ESBMC_verify_kernel_wta(
gridDim.x*gridDim.y*gridDim.z,
blockDim.x*blockDim.y*blockDim.z,
arg1,arg2,arg3)
ESBMC_verify_kernel_wta
while(i<GPU_threads){
pthread_create(&threads_id,
NULL, kernel, NULL);
i++;
}