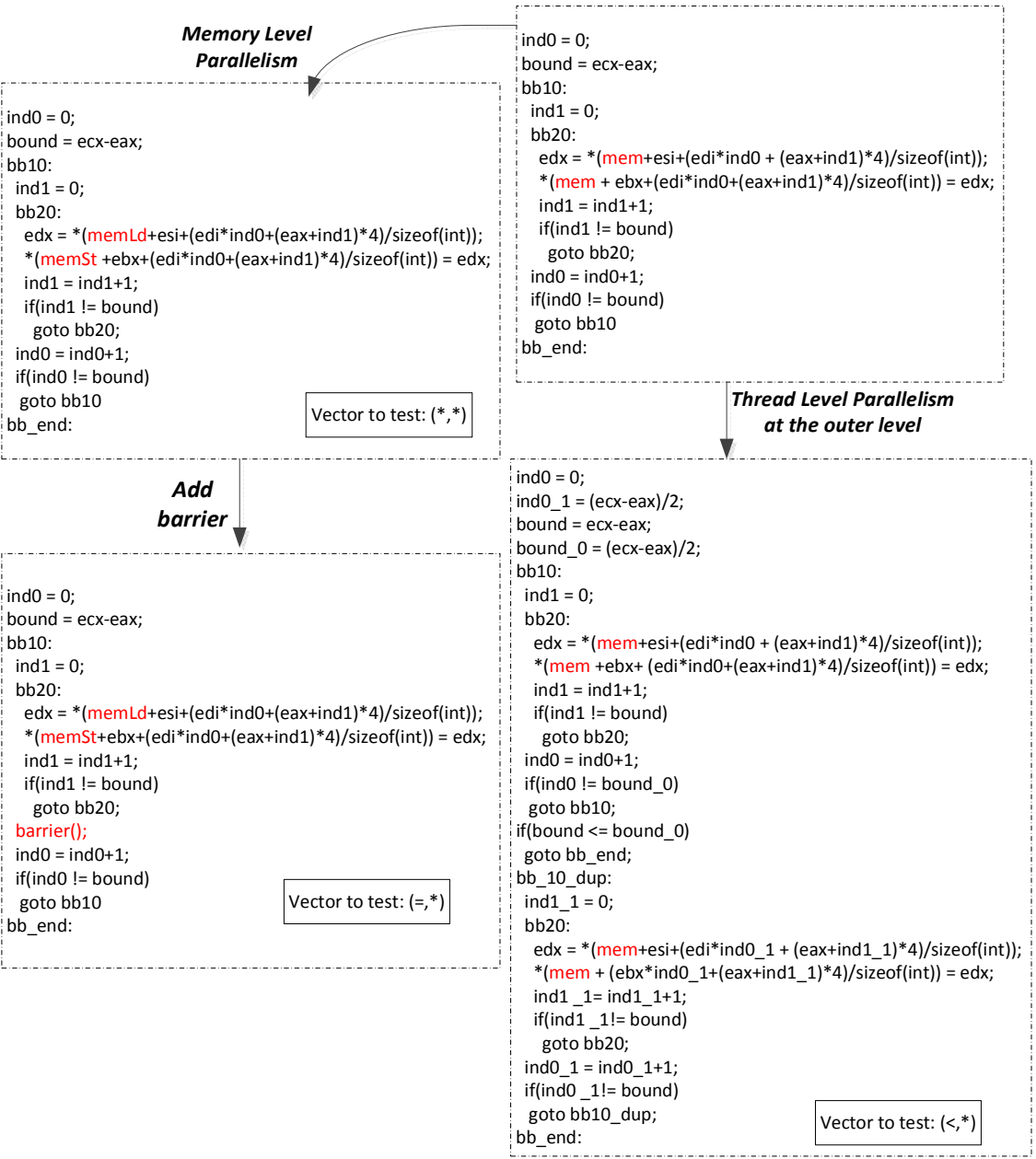
Memory Level Parallelism
ind0 = 0;
bound = ecx-eax;
bb10:
ind1 = 0;
bb20:
edx = *(mem+esi+(edi*ind0 + (eax+ind1)*4)/sizeof(int));
*(mem + ebx+(edi*ind0+(eax+ind1)*4)/sizeof(int)) = edx;
ind1 = ind1+1;
if(ind1 != bound)
goto bb20;
ind0 = ind0+1;
if(ind0 != bound)
goto bb10
bb_end:
ind0 = 0;
bound = ecx-eax;
bb10:
ind1 = 0;
bb20:
edx = *(memLd+esi+(edi*ind0+(eax+ind1)*4)/sizeof(int));
*(memSt +ebx+(edi*ind0+(eax+ind1)*4)/sizeof(int)) = edx;
ind1 = ind1+1;
if(ind1 != bound)
goto bb20;
ind0 = ind0+1;
if(ind0 != bound)
goto bb10
bb_end:
Vector to test: (*,*)
Thread Level Parallelism at the outer level
Add barrier
ind0 = 0;
bound = ecx-eax;
bb10:
ind1 = 0;
bb20:
edx = *(memLd+esi+(edi*ind0+(eax+ind1)*4)/sizeof(int));
*(memSt+ebx+(edi*ind0+(eax+ind1)*4)/sizeof(int)) = edx;
ind1 = ind1+1;
if(ind1 != bound)
goto bb20;
barrier();
ind0 = ind0+1;
if(ind0 != bound)
goto bb10
bb_end:
Vector to test: (=,*)
ind0 = 0;
ind0_1 = (ecx-eax)/2;
bound = ecx-eax;
bound_0 = (ecx-eax)/2;
bb10:
ind1 = 0;
bb20:
edx = *(mem+esi+(edi*ind0 + (eax+ind1)*4)/sizeof(int));
*(mem +ebx+ (edi*ind0+(eax+ind1)*4)/sizeof(int)) = edx;
ind1 = ind1+1;
if(ind1 != bound)
goto bb20;
ind0 = ind0+1;
if(ind0 != bound_0)
goto bb10;
if(bound <= bound_0)
goto bb_end;
bb_10_dup:
ind1_1 = 0;
bb20:
edx = *(mem+esi+(edi*ind0_1 + (eax+ind1_1)*4)/sizeof(int));
*(mem + (ebx*ind0_1+(eax+ind1_1)*4)/sizeof(int)) = edx;
ind1 _1= ind1_1+1;
if(ind1 _1!= bound)
goto bb20;
ind0_1 = ind0_1+1;
if(ind0 _1!= bound)
goto bb10_dup;
bb_end:
Vector to test: (<,*)