

SIMD: agindu bera hainbat alditan datu askoren gainean

Bektore-prozesadoreak:

- Bektore aginduak modu segmentatuan
- Bektore agindu batek bektoreko osagai guztiak prozesatzen ditu
- Arazoak: bus kopurua, unitate funtzional kopurua, bektore-erregistroen tamaina/kopurua eta kontrol dependentziak (begizta desagertu da)

Prozesadore grafikoak:

- Agindu estandarrak matrize moduan egikaritutako datu multzo bati

Koprozesadoreak prozesadoreari lotutako gailuak dira, azelaradoreak

Eragiketa jakin bat eraginkortasun handiz prozesadoreak eskatzean (n dimentsioko bektoreekin)

VL->bektore luzera | VS->bektorearen pausua

Datu-dependentsiak konpontzeko Id->Ir zirkuitulaburrak (kateaketa/chaining)

Baliabideak okupatuta badaude N ziklo galduko dira

Memoria busak

- Bakarra badago memoria aginduak zain geldituko dira bukatu arte
- Gehiago badira lehen libratzen dena erabiliko dute
- k bus eta m memoria agindu izanda exekuzio-denbora: $m/k * N$ izango da

$$T_{BE} = fT_B + (1-f)T_E$$

$$K_{BE} = K_B / (K_B - f(K_B - 1))$$

Kode bat bektorialki exekutatzeko aginduen jatorrizko ordena aldatu behar da datu-dependentsiak errespetatuz

Begizta bat bektorizatu ahal izateko aginduen arteko dependentsiek ezin dute ziklorik osatu grafoan (fisioaren bitartez agian konpon daitezke)

Antidependentsiak dituzten zikloetan irakurketa>idazketa ordena errespetatu

Irteera-dependentsiak dituzten zikloetan idazketa>idazketa ordena errespetatu

Begizten ordena trukatu daiteke, baldin eta dependentsien distantzia-bektore berrietan 0 ez den lehen elementua positiboa bada

Hobe da bektore luze gutxi prozesatzea txiki asko baino

Bektoreko osagai batzuk bakarrik prozesatu behar badira maskarak erabiltzen dira

Pausu aldakorrek bektoreak prozesatzeko indize-bektoreak erabiltzen dira

Segmentazioaren mugak gainditzeko nukleo eta UF gehiago

GP-GPUek kalkulu bektorialean ez bezela makina lengoia estandarra erabiltzen dute

Ehunka UF simple matrizeen kalkulua gauzatzeko

CPU normal bat hainbat GPUekin konbinatzen da, programan zehaztuz zein den GPUaren atala

RAM partekatua (ahalik eta handiena)

SM multiprozesadore bakoitzean:

- Hainbat core (SP edo CUDA)
- UF bereziak
- Memoria partekatua L1 cachea
- Erregistro asko

Prozesadore eskalar bakoitzean (SP):

- UF bat osoetarako eta beste bat koma higikorrerako
- Eragiketa logikoak, konparazioak...
- Jauziak kudeatzeko unitatea

CUDAko hariak oso arinak dira

Errendimendu altua lortzeko paralelismo maila altuak

Memoria hierarkiaren goi mailetako atzipenak gero eta garestiagoak