Many Body Algorithms

Andrey Asadchev

Iowa State University

May 30, 2012

Outline

- MP2
- CC

Basic facts

- Cartesian shells $((L+1)^2 + L + 1)/2$
- Spherical shells $2(L+1)+1$
- Occupied/Active: 20-200
- Virtual: 200-5000
- Cartesian vs Spherical: 25% difference is 50% in storage/computations
- AO basis size alone doesn't determine complexity

Scalable/ flexible algorithm

- small memory footprint, few hundred MB per core
- flexible storage transparent to an algorithm, memory/FS
- large number of cores, multilevel parallelism
- ability to utilize extra memory
- NO^2 cap

Hardware

'

- Cray:
  Many cores (2k), 2GB per core, 32 cores per node
  Parallel FS (Lustre)
- BG:
  Many cores, 500MB per core, 4 cores per node
  Parallel FS (GPFS)
- Parallel FS:
  File split over multiple storage nodes in chunks of N bytes
  high-bandwidth, large space 10s of TB

Small MP2, 6 cores

```
active: 76
virtual: 1653
atomic: 1995
Array::HDF5: mp2.v(qsij) { 2902458368, 5 }, 116098 MB parallel=0
OpenMP threads: 6
memory (per thread):581.742 MB

eri, transformations 1+2 (master thread):
    eri: 03:22:32.518858
    trans 1: 04:00:22.352548
    trans 2: 00:28:23.818739
    I/O: 00:00:15.049650, 1213 MB/s
    time: 07:56:42.380269
    memory: 3328.75 MB
```

Large MP2, 512 cores

```
active: 222
virtual: 3300
atomic: 4080

Array::HDF5: mp2.v(qsij) { 3229614080, 128 }, 3.30712e+06 MB parallel=1
OpenMP threads: 32
eri, transformations 1+2 (master thread):
    eri: 00:45:06.848040
    trans 1: 00:47:48.136083
    trans 2: 01:11:31.128841
    I/O: 00:14:37.070417, 15.2066 MB/s
    time: 03:31:12.340953
    memory: 24638 MB

transformations 3+4 (master thread):
    trans 3+4: 01:59:18.939502
    I/O: 01:05:44.724088, 104.808 MB/s
    memory: 32766.5 MB
```

Last 2 Transformations

```
V(i,j,q,s)
for s in S {
    t <- V(:,:,:,s)
    transform t
    t -> V(:,:,:,s)
}

// naive
for q in Q {
    t <- V(:,:,q,:) // noncontiguous, extremely slow
}

// smarter
for q,s in Q,S {
    transpose V(:,:,q,s)
}

better to generate the data to avoid the above patterns
```

Small CCSD(T), 6 cores

Cafe: 37 active, 195 virtual, 260 AO
CCSD(T):
SD: 3 h per iteration
SD: 2 h per iteration with GPU
(T): 66 h

Larger CCSD(T), 512 cores

Nose candy:
NUMBER OF CARTESIAN GAUSSIAN BASIS FUNCTIONS = 430
TOTAL NUMBER OF MOS = 409
NUMBER OF OCCUPIED MOS = 78
NUMBER OF FROZEN CORE MOS = 21
NUMBER OF FROZEN VIRTUAL MOS = 0
SD: 8 mins
(T): 1:09 h

Larger CCSD(T), 1024 cores

PCC-15:
NUMBER OF CARTESIAN GAUSSIAN BASIS FUNCTIONS = 975
TOTAL NUMBER OF MOS = 828
NUMBER OF OCCUPIED MOS = 45
NUMBER OF FROZEN CORE MOS = 15
SD: 14 mins
(T): 2:12 h

Even larger CCSD(T)

PCC-15:
NUMBER OF CARTESIAN GAUSSIAN BASIS FUNCTIONS = 1260
TOTAL NUMBER OF MOS = 1005
NUMBER OF OCCUPIED MOS = 45
NUMBER OF FROZEN CORE MOS = 15
SD: 24/40 mins (1024/512)
(T): 4:44 h

Largest CCSD(T)

Taxol:
NUMBER OF CARTESIAN GAUSSIAN BASIS FUNCTIONS = 660
TOTAL NUMBER OF MOS = 660
NUMBER OF OCCUPIED MOS = 226
NUMBER OF FROZEN CORE MOS = 62
SD: 2:05/4:13 h (1024/512)
(T): $\tilde{4}0$ h

Look Blocking

```
for k = 0:M,N {
  for j = 0:M,N {
    for i = 0:M,N {
      Load A(*,*,i:i+N) // innermost load
      ...
    }
  }
}
```

- $N = 1$: $M * (M * M)$ loads
- $N = 5$: $M * (M * M)/5^2$ loads

Aknowledgements

- Dr. M.$.Gordon