

- ▶ SciDAC package integration
 - ▶ QMP
Fully functional QMP backend is written and being used on BG/[L,P] and recently clusters.
 - ▶ QIO
QIO backend for gauge and propagator files are written (by E. Scholz) and being used extensively. SciDAC propagator format is the canonical format for CPS.
 - ▶ QLA? QMT?
So far, the platforms CPS has been used heavily (QCDOC, BG/[LP]), didn't give big motivation to look into these, as we have BAGEL-generated linear algebra routines and, at least so far, multi-threading hasn't helped.
We are using more clusters now, so we're happy to look into these more seriously (Oliver?)

- ▶ CPS on BG/P CPS on BG/{L,P} is largely based on CPS on QCDOC
QCDOC assemblies can be used on BG. Usually only $\sim 30\%$ slower, because most routines are memory bandwidth bounded.
 - ▶ Wilson/DWF dslash \rightarrow BAGEL by Peter Boyle *
 - ▶ P4 dirac, Gauge/Fermion force ... \rightarrow mostly QCDOC, some Double Hummer assembly
 - ▶ Change communication to QMP *

Using multicore

- ▶ Likely help, especially for smaller local volume ($< 4^4$)
- ▶ BAGEL already supports POSIX threads
- ▶ Rest of CPS?
Most of performance-critical routines are multidimensional (4-8) parallel transports \rightarrow maps well to modest number of cores.
- ▶ Planning to put in (depending on BAGEL performance).
- ▶ Tried OpenMP on BG/P, didn't see a performance gain..

P4 Inverter, 512node BG/P, $8^3(\times 4)$

Volume	$32^3 \times 8$	$48^3 \times 8$
VN	132	280
SMP / 4	102	239

- ▶ Should try QMT.

Further extension of QMP?

Changing starting address(es) of existing message handles.

- ▶ Parallel transports often requires communications with same pattern (length, stride, # of blocks) with different starting addresses.

Currently, only way to do this is to generate new handles! → significant impact on performance for small volumes.

- ▶ Example: P4 dirac operator

512 node BG/L partition ($8^3 \times 2$), local volume 4^4

Total: 434.3 usec

computation: 149.5 usec

QMP setup : 85 usec

QMP_start : 66 usec

QMP_wait: 100 usec

Destroying and re-creating message handles is taking 20% of total time!! Will be worse for BG/P.

- ▶ On MPI, Can be implemented without adding overhead to QMP_start()/QMP_wait(), QMP_declare_msgmem() (or QMP_declare_{send,receive}())
- ▶ New MPI implementation of strided transfers made this even simpler.
- ▶ Binding

```
QMP_status_t QMP_change_address  
    (QMP_msgmem_t m, void *new_base);
```

For arraied handles, new_base becomes the starting address for the first block. Shifts all start address by the same amount. Fails for 'merged' handles. OK to fail.

- ▶ checked into Jlab SVN repository **What is needed to merge this to the main branch?**

Wish list?

- ▶ Concurrency control in parallel I/O
Helps in controlling stress to the file system.
Proved to be very helpful on QCDOC.
On BG/P, could eliminate the need to write in multifile format and have to split/unsplit.
- ▶ Alternate implementation of Serial I/O.
CPS does barrel shifting in Serial I/O node. Proven to be very robust and efficient on mesh machines.
- ▶ In general, need reorganization of SciDAC packages to support multiple implementation less awkward?