Porting PLASMA_dgemm to OmpSs

Sicong Zhuang

February 27, 2015

1 Global changes

- *plasma_context_self* (control/context.c: 136). PLASMA allows multiple contexts, each of which is associated with the master thread id of that context. In OmpSs

  ```
  #pragma omp taskwait
  ```

  might change the thread id which renders this mechanism unusable. As a workaround, only one context is assumed through the entire execution.

```
136 plasma_context_t *plasma_context_self() {
137 #if 0
138         int i;
139 
140         // For each entry
141         for (i = 0; i < CONTEXTS_MAX; i++) {
142                 // If id matches
143                 if (pthread_equal(context_map[i].thread_id, pthread_self())) {
144                         return context_map[i].context;
145                 }
146         }
147         return NULL;
148 #endif
149         return context_map[0].context;
150 }
```

- *PLASMA_Finalize* (control/control.c: 320). Since we are not using PLASMA nor QUARK to handle the threads, the various barriers, synchronizations in this function could cause deadlocks, hence removed along with the context reset.

```
333 #if 0
334         /* Terminate the dynamic scheduler */
335         plasma_dynamic_sync();
336 
337         /* Free quark structures */
338         QUARK_Free(plasma->quark);
339 
340         /* Set termination action */
341         pthread_mutex_lock(&plasma->action_mutex);
342         plasma->action = PLASMA_ACT_FINALIZE;
343         pthread_mutex_unlock(&plasma->action_mutex);
344         pthread_cond_broadcast(&plasma->action_condt);
345 
346         /* Barrier and clear action */
347         plasma_barrier(plasma);
348         plasma->action = PLASMA_ACT_STAND_BY;
349 
```

```
        // Join threads
        for (core = 1; core < plasma->world_size; core++) {
                status = pthread_join(plasma->thread_id[core], &exitcodep);
                if (status != 0) {
                        plasma_fatal_error("PLASMA_Finalize", "pthread_join() failed");
                        return status;
                }
        }
        plasma_barrier_finalize(plasma);
        plasma_barrier_bw_finalize(plasma);
#endif
```

```
#if 0
        status = plasma_context_remove(plasma, pthread_self());
        if (status != PLASMA_SUCCESS) {
                plasma_fatal_error("PLASMA_Finalize", "plasma_context_remove() failed");
                return status;
        }
#endif
```

2 dgemm/pdgemm

- *PLASMA_dgemm* (compute/dgemm.c: 17)

```
#pragma omp taskwait
//        plasma_dynamic_sync();
```

- *plasma_pdgemm_quark*(compute/pdgemm.c: 150) Replace the QUARK_CORE_dgemm call with CORE_dgemm (OmpSs task)

```
CORE_dgemm(transA, transB, tempmm, tempnn, tempkn,
        alpha, A(m, k), ldam, B(k ,n), ldbk, zbeta, C(m, n), ldcm);
//QUARK_CORE_dgemm(
//     plasma->quark, &task_flags,
//     transA, transB,
//     tempmm, tempnn, tempkn, A.mb,
//     alpha, A(m, k), ldam,  /* lda * Z */
//     B(k, n), ldbk,  /* ldb * Y */
//     zbeta, C(m, n), ldcm); /* ldc * Y */
```

- Adding OmpSs task (include/core_dblas.h: 62)

```
#pragma omp task in(A[0;M*K-1], B[0;N*K-1]) inout(C[0;M*N-1])
void CORE_dgemm(PLASMA_enum transA, PLASMA_enum transB,
                int M, int N, int K,
                double alpha, const double *A, int LDA,
                                          const double *B, int LDB,
                double beta,        double *C, int LDC);
```

3 Makefile changes

- The final PLASMA static library should be linked with nanox libraries.

  `LDFLAGS += -L/path/to/nanox -lnanox-ss -lnanox-omp -lnanox-c -lnanox`

- Makefile in compute/ (compute/Makefile)

```
dgemm.o : dgemm.c
        mcc --ompss $(CFLAGS) $(INC) -c $< -o $@

pdgemm.o : pdgemm.c
        mcc --ompss $(CFLAGS) $(INC) -c $< -o $@
```

- Makefile in core_blas (core_blas/Makefile)

```
core_dgemm.o : core_dgemm.c
        mcc --ompss $(CFLAGS) $(INC) -c $< -o $@
```

4 Out-Of-Place Format Conversion (optional)

The porting of format conversions *plasma_pdlapack_to_tile_quark* and *plasma_pdtile_to_lapack_quark* is the same as the dgemm.

- *plasma_pdlapack_to_tile_quark*(control/pdtile.c: 88) replace QUARK_CORE_dlacpy with CORE_dlacpy

- *plasma_pdtile_to_lapack_quark*(control/pdtile.c: 187) replace QUARK_CORE_dlacpy with CORE_dlacpy

- Adding OmpSs task pragma to include/core_dblas.h: 187)

  `#pragma omp task inout(A[0;M*N-1], B[0;M*N-1])`

- Makefile in core_blas (core_blas/Makefile)

```
core_dlacpy.o : core_dlacpy.c
        mcc --ompss $(CFLAGS) $(INC) -c $< -o $@
```

- Makefile in control/ (control/Makefile)

```
pdtile.o : pdtile.c
	mcc --ompss $(CFLAGS) $(INC) -c $< -o $@
```