Parallel Computing Libraries

Chang Wang
Agenda

- POSIX Threads
- OpenMP
- OpenMPI
- OpenCL
- Grand Central Dispatch
POSIX Threads
Pthreads

• A portable threading library (A set of C language programming types and procedure calls)

• Improve program performance

• It is “lightweight”
Pthreads

• Thread Management
  - creating, detaching, joining, query
• Mutexes
  - dealing with synchronization
• Condition Variables
  - address communications between threads
• Synchronization
  - manage read/write locks and barriers
OpenMP

- An Application Program Interface (API) that may be used to explicitly direct multi-threaded, shared memory parallelism
- Supports C/C++ and Fortran
- Comprised of three primary API components
  - Compiler Directives
  - Runtime Library Routines
  - Environment Variables
OpenMP uses the fork-join model of parallel execution
OpenMP

#pragma omp directive-name [clause...] newline

parallel if (scalar_expression)
for private (list)
sections shared (list)
critical default (shared | none)
num_threads (integer-expression)

.......
OpenMPI

- Message Passing Interface, A standard
- OpenMPI is one implementation
- Standardization
- Portability
- Performance Opportunities
OpenMPI

- **Point to Point Communication Routines**
  - Blocking message passing routines
  - Non-Blocking message passing routines

- **Collective Communication Routines**
  - Communicator is a group of processes (MPI-tasks) may communicate with each other
OpenMPI

tid #0
MPI_Init(...)

tid #1
MPI_Init(...)

tid #2
MPI_Init(...)

tid #3
MPI_Init(...)
OpenMPI

- tid #0
  - MPI_Comm_size(...)
  - MPI_Comm_rank(...)
- tid #1
  - MPI_Comm_size(...)
  - MPI_Comm_rank(...)
- tid #2
  - MPI_Comm_size(...)
  - MPI_Comm_rank(...)
- tid #3
  - MPI_Comm_size(...)
  - MPI_Comm_rank(...)
OpenMPI

tid #0
if (tid == 0) {
    for (...) {
        MPI_Send(...)
    }
}

tid #1
else {
    MPI_Recv(...)
}

tid #2
else {
    MPI_Recv(...)
}

tid #3
else {
    MPI_Recv(...)
}
OpenCL

• Approachable language for accessing heterogeneous computational resources
• Supports parallel execution on single or multiple processors
• Compiler, API, runtime
OpenCL

- **Compute Kernel**
  - Basic unit of executable code - similar to a C function
  - Data-Parallel or Task-Parallel

- **Compute Program**
  - Collection of compute kernels and internal functions

- Applications queue compute kernel execution instance
OpenCL
Grand Central Dispatch
GCD

- A revolutionary approach to multicore computing
- Shifting the responsibility for managing threads and their execution from applications to OS
GCD

- Block Objects
  - An extension to C/C++ and Obj-C
  - easy to define self-contained units of work
  - ban be defined inline - anonymous functions
  - capture read-only copies of local variables (lambda, closure)
GCD

void (^my_block)(void);

typedef void (^blockWithString)(char *);
char *greetings = "Hello";
blockWithString b = ^(char *place) {
    printf("%s %s\n", greetings, place);
};
b("world");
References

• POSIX Threads Programming: https://computing.llnl.gov/tutorials/pthreads/

• OpenMP: http://openmp.org/wp/

• OpenMPI: http://www.open-mpi.org/
