UPC Overview and Update

Greg Fischer
Cray Inc.
gsf@cray.com

CRAY

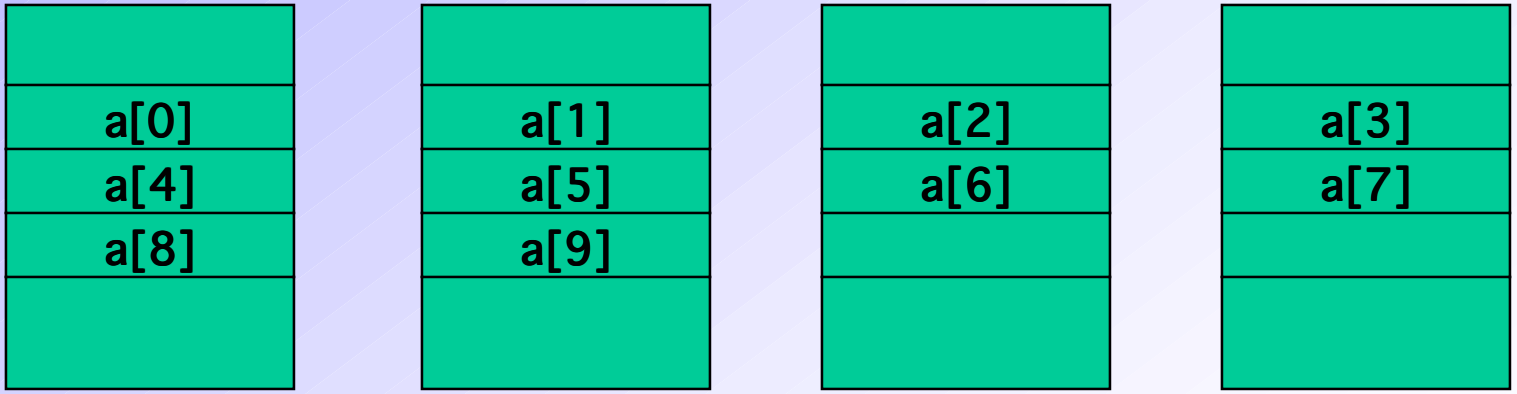
What is UPC?

- Unified Parallel C
- Syntactic extension to Standard C
- Designed by IDA CCS, UC-Berkeley, LANL
- SPMD plus HPF-like data and work distribution features

CRAY

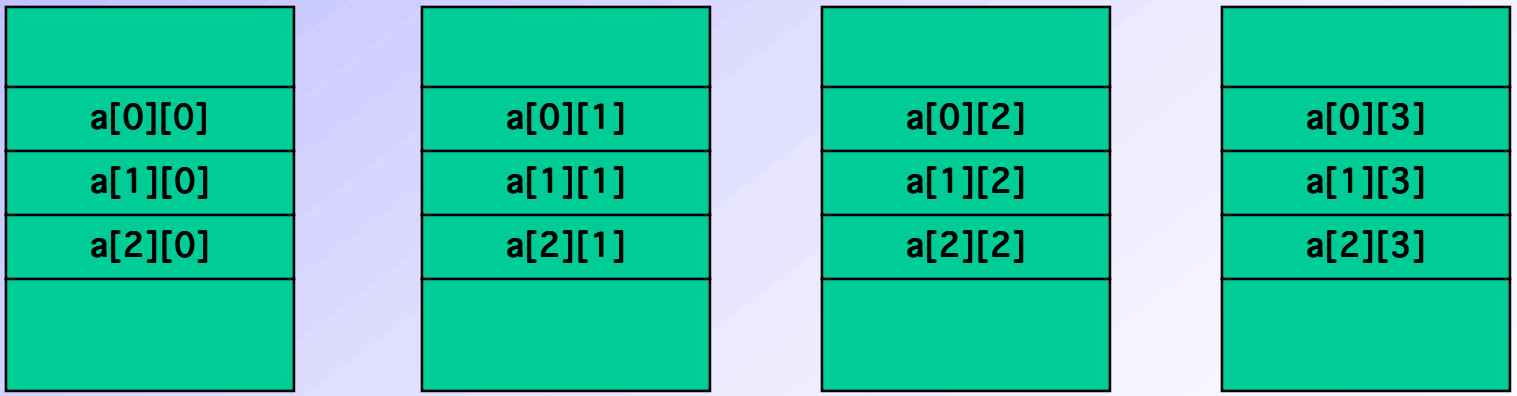
UPC Examples

```
shared double a[10];
...
upc_forall( i=0; i<10; i++; a[i] ) {
    a[i] = 1.0;
}
```



UPC Examples

shared double a[3][THREADS];



Why a Cray UPC Subset?

- Full UPC is “large”; many combination of array shapes and distributions
- Full UPC supports several "styles" of distributed memory programming
- Cray UPC Subset focuses on "Data Passing“ style

Data Passing

- AKA One-sided Communication
- SPMD (Single Program, Multiple Data)
- Replicated data and execution across PEs
- Explicit communication (like MPI)
- Direct reads and writes to other PE's Data
- Library-based (SHMEM)
- Syntax-based (Co-array Fortran, UPC)

CRAY

Halo Exchange: MPI

```
double ai[6][ihp][ip];
...
mpi_send_init( &ai[0][0][0],    ihp*ip, mpi_real, thd[np+1],
                    9905, mpi_comm_world, &mpireq[1] );
mpi_send_init( &ai[1][0][0], 2*ihp*ip, mpi_real, thd[np-1],
                    9905, mpi_comm_world, &mpireq[2] );
mpi_recv_init( &ai[3][0][0],    ihp*ip, mpi_real, thd[np-1],
                    9905, mpi_comm_world, &mpireq[3] );
mpi_recv_init( &ai[4][0][0], 2*ihp*ip, mpi_real, thd[np+1],
                    9905, mpi_comm_world, &mpireq[4] );
mpi_startall( 4, mpireq );
mpi_waitall ( 4, mpireq, mpistat )
```

CRAY

Halo Exchange: SHMEM

```
double ai[6][ihp][ip];
...
shmem_barrier_all();
shmem_get_64( &ai[3][0][0], &ai[0][0][0],    ihp*ip,
                            thd[np-1] );
shmem_get_64( &ai[4][0][0], &ai[1][0][0], 2*ihp*ip,
                            thd[np+1] );
shmem_barrier_all();
```

CRAY

Halo Exchange: Cray UPC Subset
(Rightmost PE Dimension)

```
shared double ai[6][ihp][ip][THREADS];
...
upc_barrier();
for ( j=0; j<ihp; j++ ){
    for ( i=0; i<ip; i++ ){
            ai[3][j][i][MYTHREAD] = ai[0][j][i][thd[np-1]];
            ai[4][j][i][MYTHREAD] = ai[1][j][i][thd[np+1]];
            ai[5][j][i][MYTHREAD] = ai[2][j][i][thd[np+1]];
    }
}
upc_barrier();
```

CRAY

The Cray UPC Subset

- Shared arrays must have cyclic or block distribution.
- Cyclic must have rightmost extent be multiple of THREADS.
- Block must have leftmost extent be multiple of THREADS.
- “upc_forall" not supported (not essential for data passing)

Advantages of Data Passing over Message Passing

- Only one side of communication needs to be programmed, hence less redundancy, more maintainability
- Reduced communication time overhead, hence better scalability (for supporting hardware, such as T3E, SV2)

Advantages of Cray UPC Subset over SHMEM

- Eliminates function calls for communication
- More readable, hence more maintainable
- Transfer size and type implicit, hence more reliable
- Allows compiler to understand, overlap communication, avoid memory-to-memory copy (for supporting hardware, such as T3E, SV2)

Overlapping Communication, Computation

Finite differencing on rectangular grid

```
shared double u[nrow][THREADS];
double new_u[nrow];
...
for ( i=0; i<nrow; i++ ) {
    new_u[i] = new_u[i] + u[i][left] + u[i][right];
}
```

UPC Availability

- SV2 FCS
- Available on Compaq, SGI platforms
- In development at SUN, HP, IBM
- Open source in development, DOE grant, http://www.pmodels.org

Conclusion

- Cray UPC Subset designed to support data passing
- Cray UPC Subset offers a method for improving performance, maintainability and reliability of explicit communication C codes
- UPC available soon on most platforms

Bibliography

- El-Ghazawi, Tarek A., Carlson, William W., Draper, Jesse M., "UPC Language Specifications V1.0“, http://hpc.gmu.edu/~upc/, February, 2001
- Numrich, R.W. and Reid, J.K., "Co-array Fortran for Parallel Programming", technical report RAL-TR-1998-060, Rutherford Appleton Laboratory, Oxon OX11 0QX, UK, August 1998, ftp://matisa.cc.rl.ac.uk/pub/reports/nrRAL98060.ps.gz

CRAY