PGAS10 Workshop

Introducing OpenSHMEM

Barbara Chapman, Tony Curtis, University of Houston; Charles Koelbel, Jeffery Kuehn, Oak Ridge National Laboratory; Stephen Poole, Oak Ridge National Laboratory and OSSS, Inc.; Lauren Smith, OSSS, Inc. & DOD
Introducing OpenSHMEM

• Structure of the Talk
  – Overview of SHMEM & PGAS
    • Features
    • A brief history
  – Defining OpenSHMEM
    • Standardization issues
    • Implementation work
    • Community building
Introducing OpenSHMEM

• Introduction to SHMEM and PGAS
  – MPI became de facto standard library for distributed parallel computing
    • Message-passing, send + acknowledge
  – 1-sided communication requires less overhead
    • Shoot first, ask questions later
    • SHMEM is such a library
    • SHared MEMory
Introducing OpenSHMEM

• Introduction to SHMEM and PGAS
  – SHMEM has
    • Point-to-point put & get
    • Broadcast & collect
    • Arithmetical and logical reductions
    • Atomic operations, critical sections and locks
Introducing OpenSHMEM

• Introduction to SHMEM and PGAS
  – C and Fortran interfaces
  – Variables can be allocated with global visibility
    • All processors see a named variable
    • Global Address Space
  – Processors with separate same-named variables
    • Each processor sees the same name, but has a separate copy
    • **Partitioned** Global Address Space
  – Can underpin PGAS languages
    • Chapel, X10, Co-Array Fortran, UPC, Titanium...
Introducing OpenSHMEM

2-sided communication with acknowledgement

1-sided communication without acknowledgement
Introducing OpenSHMEM

- Communication in SHMEM
  - Symmetric variables
    - Accessible from remote processors (put/get)
    - Same name on all processors
    - At same relative address
    - But differing values
  - Fences and Barriers
    - To synchronize previous 1-sided communication
Introducing OpenSHMEM

#include <stdio.h>
#include <mpp/shmem.h>

int main(int argc, char **argv)
{
    int me, npes, right;

    start_pes(0);

    me   = __my_pe();
    npes = __num_pes();

    ...

    right = (me + 1) % npes; /* right neighbor in ring */
    shmem_int_put(dest, source, 1, right);

    ... something interesting goes here ...

    shmem_barrier_all();

    return 0;
}
A portable shmem implementation

```c
x = (int *) shmalloc(sizeof(int));

if (_my_pe() == 0) {
    shmem_int_put(dest = x, src = a, len = 1, pe = 1);
}

// has "a" arrived yet? We don't know...
```

Symmetric memory

Put "a" -> x @ PE 1

Same offset, but at potentially different addresses
Introducing OpenSHMEM

• Taking advantage of hardware for performance
  – Hardware offload frees other resources
  – Remote direct memory access
    • Processor can “put” directly to another processor’s memory without interrupting
  – Atomic, collective, locking and barrier operations can also benefit
  – Can produce substantial performance gains
Introducing OpenSHMEM

• Examples of SHMEM’s features
  – Point-to-point put & get
    • `shmem_long_put(long *dest, long *src, len, pe)`
  – Broadcast & collect
    • `shmem_broadcast64(dest, src, n, root, start, stride, size, sync)`
  – Arithmetical and logical reductions
    • `shmem_long_sum_to_all(...)`
  – Atomic operations, critical sections and locks
    • `shmem_swap(long *dst, long *src, int pe)`
Introducing OpenSHMEM

• A brief history of SHMEM
  – Cray (1993)
    • T3D
      – Ordered communication
    • T3E
      – Communication became unordered, extend API to cope
  – SGI (1997)
    • 64-bit extensions
  – Quadrics (1998)
    • Included SGI extensions
    • Has non-blocking puts and gets
    • On top of QsNet
  – GPSHMEM (2000)
    • On top of ARMCI
  – Other versions include: HP, SiCortex (based on Quadrics API); IBM
Introducing OpenSHMEM

• Standardization & community
  – Various versions of SHMEM diverged
    • Different APIs, usage restrictions
    • So code is not directly portable
    • Simple example:

<table>
<thead>
<tr>
<th>SGI</th>
<th>Quadrics</th>
<th>SiCortex</th>
</tr>
</thead>
<tbody>
<tr>
<td>start_pes(int npes)</td>
<td>start_pes(int npes)</td>
<td>start_pes(int npes) NO-OP</td>
</tr>
<tr>
<td>shmem_init(void)</td>
<td>shmem_init(void)</td>
<td>shmem_init(void)</td>
</tr>
</tbody>
</table>
Introducing OpenSHMEM

• Here’s “Hello World” again on an SGI Altix

```c
#include <stdio.h>
#include <mpp/shmem.h>

int main(int argc, char **argv) {
    int me, npes;
    start_pes(0);

    me = _my_pe();
    npes = _num_pes();

    printf("Hello from node %4d of %4d\n", me, npes);

    return 0;
}
```

Not the same in all SHMEMS
Introducing OpenSHMEM

• Standardization & community
  – We propose a process to standardize and extend SHMEM
    • To be called OpenSHMEM
  – Steve Poole founded “Open Source Software Solutions” (OSSS)
    • A home for OpenSHMEM
    • SGI transferred rights to SHMEM to OSSS
    • SGI has permanent chair
  – Form community to move forward and develop materials
Introducing OpenSHMEM

• Towards OpenSHMEM
  – Take the SGI version as a starting-point
    • SGI (Altix) implementation as reference
      – With editorial rewrites
    • A.k.a. version 1.0
  – Develop new specification as version 2.0
    • Solicit new ideas from community
    • What features should be changed/added?
  – Reference implementations
    • New OpenSHMEM to be written by University of Houston
    • Baseline for future development
    • Start with 1.0 and move toward 2.0
Introducing OpenSHMEM

OpenSHMEM programmer

OpenSHMEM API

Internal Comms API  \[\cdots\]  Internal Symm. Memory API

GASNet / ARMCI / direct drivers...

Network Layer: IB, Quadrics, Myrinet, ...

University of Houston: Implementation Structure
Introducing OpenSHMEM

• Validation & Verification
  – Merge existing test suites
  – Build core tests of correctness
  – And of performance
    • To compare implementations of collective algorithms
    • For tuning of underlying libraries/transports
    • To evaluate adaptive behavior
  – Iowa State working with University of Houston
Introducing OpenSHMEM

• OpenSHMEM outreach and participation
  – OpenSHMEM web site (under construction)
    • Community Wiki
    • Documentation: FAQ, cheatsheet, specification
    • Training material / tutorials
    • Software downloads
      – Source code of OpenSHMEM versions
      – Validation and Verification Suite
      – Sample programs
  – Conferences/workshops
  – Mailing list(s)
Introducing OpenSHMEM

• OpenSHMEM outreach
  – SC10 (New Orleans, November 13-19)
    • Birds of a Feather meeting
      – OpenSHMEM: SHMEM for the PGAS community at large
      – Wednesday, November 17th, 5:30pm – 7:00pm
    • Current exhibition booth presence
      – PGAS (#1233)
      – Oak Ridge National Laboratory (#3325)
      – Gulf Coast Academic Supercomputing (#2401)
      – Cray (#2829)
      – SGI (#3313)
Introducing OpenSHMEM

• To get involved
  – OpenSHMEM mailing list
    • https://email.ornl.gov/mailman/listinfo/openshmem
  – OpenSHMEM web site
    • COMING SOON!
  – SC10 Birds of a Feather
    • http://sc10.supercomputing.org/schedule/event_detail.php?evid=bof159
  – Come talk with the OpenSHMEMers here
    • Lauren Smith, Chuck Koelbel, Tony Curtis