




SC11

INTRODUCING OPENSHMEM



Barbara Chapman, Tony Curtis, **Swaroop Pophale** University of Houston;
Jeffery Kuehn, Oak Ridge National Laboratory;
Stephen Poole, Oak Ridge National Laboratory and OSSS, Inc.;
Lauren Smith, OSSS, Inc. & DOD



Introducing OpenSHMEM

Outline

- Structure of the Talk
 - Overview of PGAS and SHMEM
 - Features
 - A brief history
 - Defining OpenSHMEM
 - Standardization issues
 - Implementation work
 - Community building

Introducing OpenSHMEM

PGAS and SHMEM

- MPI became de facto standard library for distributed parallel computing
 - Message-passing, send + acknowledge
- PGAS
 - Processors with separate same-named variables
 - Each processor sees the same name, but has a separate copy
 - **Partitioned** Global Address Space
 - CAF, UPC, SHMEM

Introducing OpenSHMEM PGAS

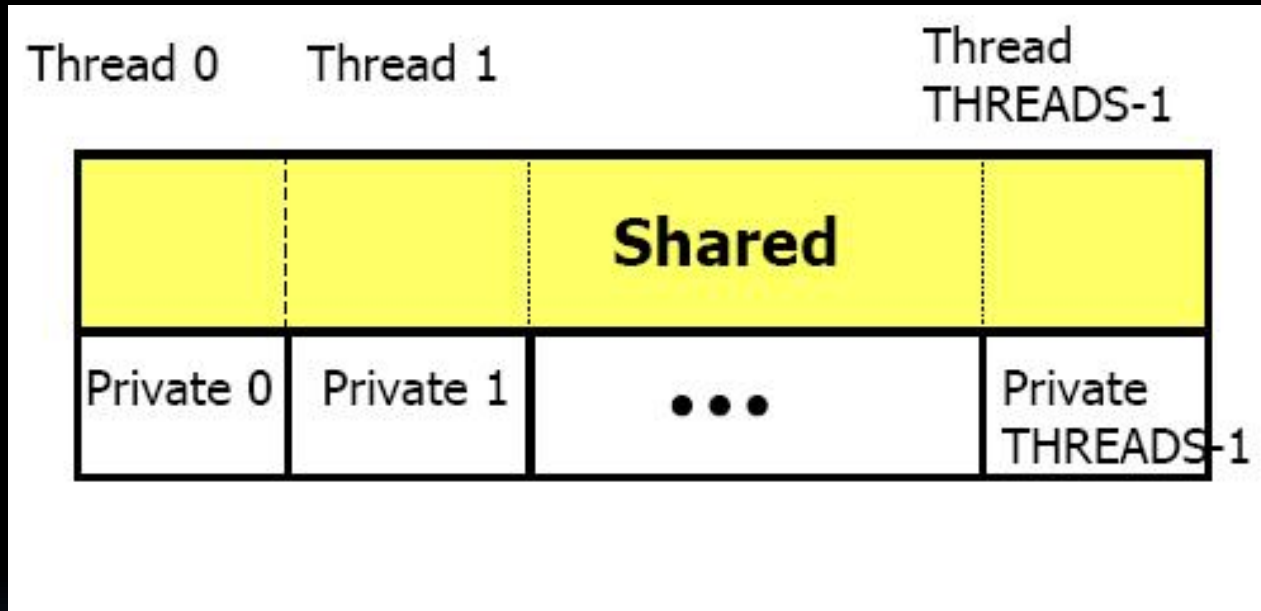


Figure 1. Logical view of a PGAS Execution Model (UPC)


Introducing OpenSHMEM

SHMEM

- 1-sided communication requires less overhead
 - Shoot first, ask questions later
 - SHMEM is such a library
 - **Symmetric Hierarchical MEMory**
- Introduction to SHMEM
 - C and Fortran interfaces
 - Variables can be allocated with global visibility
 - All processors see a named variable
 - Global Address Space



Introducing OpenSHMEM SHMEM Feature Set

- Point-to-point put & get
 - Broadcast & collect
 - Arithmetical and logical reductions
 - Atomic operations
 - Locks
- 



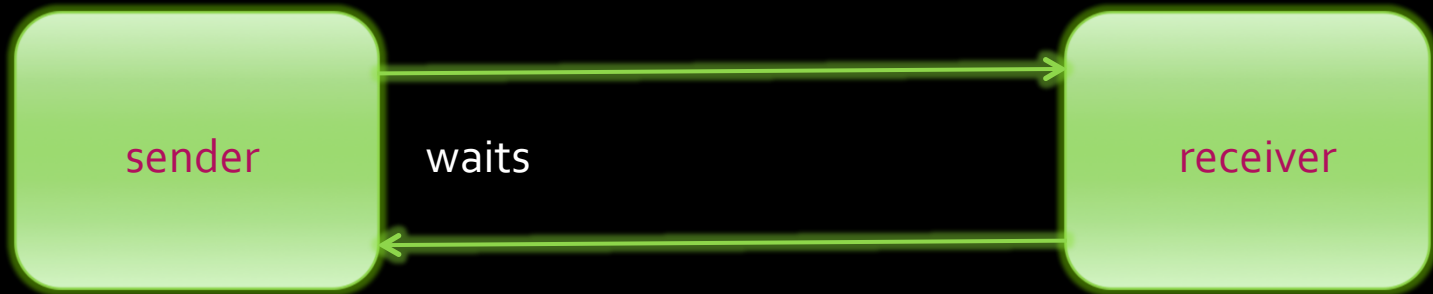
Introducing OpenSHMEM

One-sided communication

- One-sided communication allows one PE to access certain variables of another PE without interrupting the other PE
- SHMEM facilitates one-sided communication through SYMMETRIC variables
- There are two types of Symmetric Variables
 - Globals
 - Dynamically allocated and maintained by the SHMEM library

Introducing OpenSHMEM

One-sided communication



2-sided communication with acknowledgement



1-sided communication without acknowledgement



Introducing OpenSHMEM

One-sided communication

- 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

Symmetric Variables

- 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

Symmetric Variables

- Arrays or variables that exist with the **same size, type, and relative address** on all PEs.
- The following kinds of data objects are symmetric:
 - Globals
 - C/C++: Non-stack variables,
 - Fortran: objects in common blocks or with the **SAVE** attribute
 - dynamically allocated
 - C/C++: shmalloc,
 - Fortran: shpalloc

Introducing OpenSHMEM Symmetric Variables

```
int main (void)
{
  int *x;
  ...
  start_pes(0);
  ...
  x = (int*) shmalloc(sizeof(x));
  ...
  ...
  shmem_barrier_all();
  ...
  shfree(x);
  return 0;
}
```

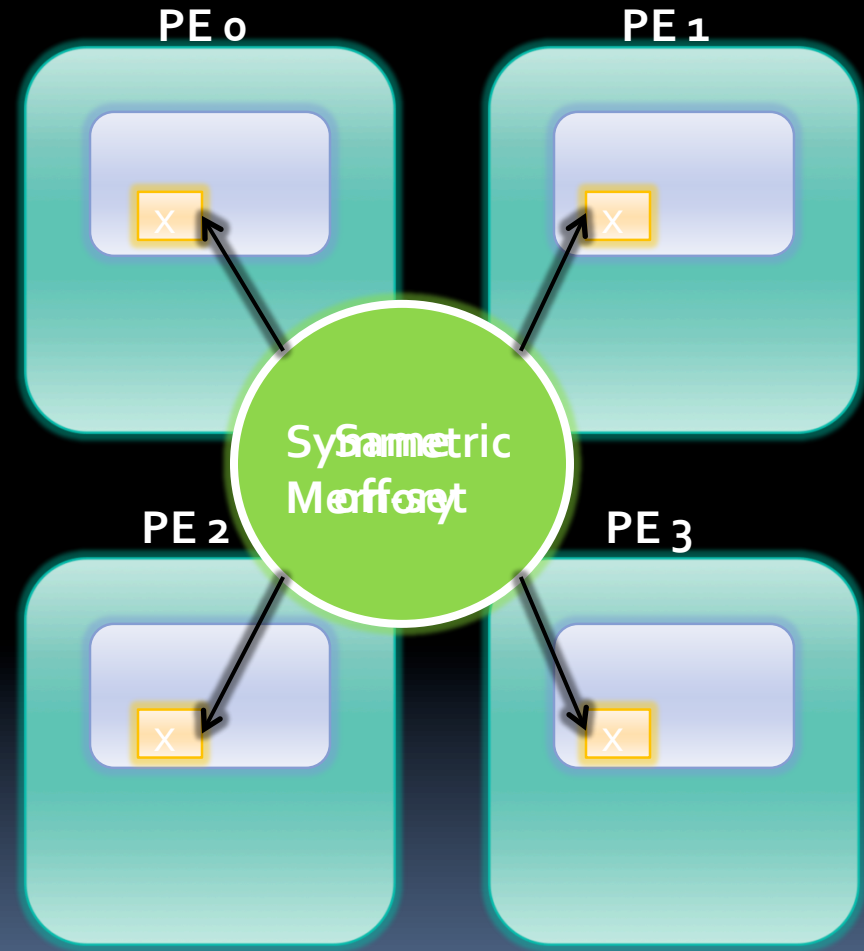
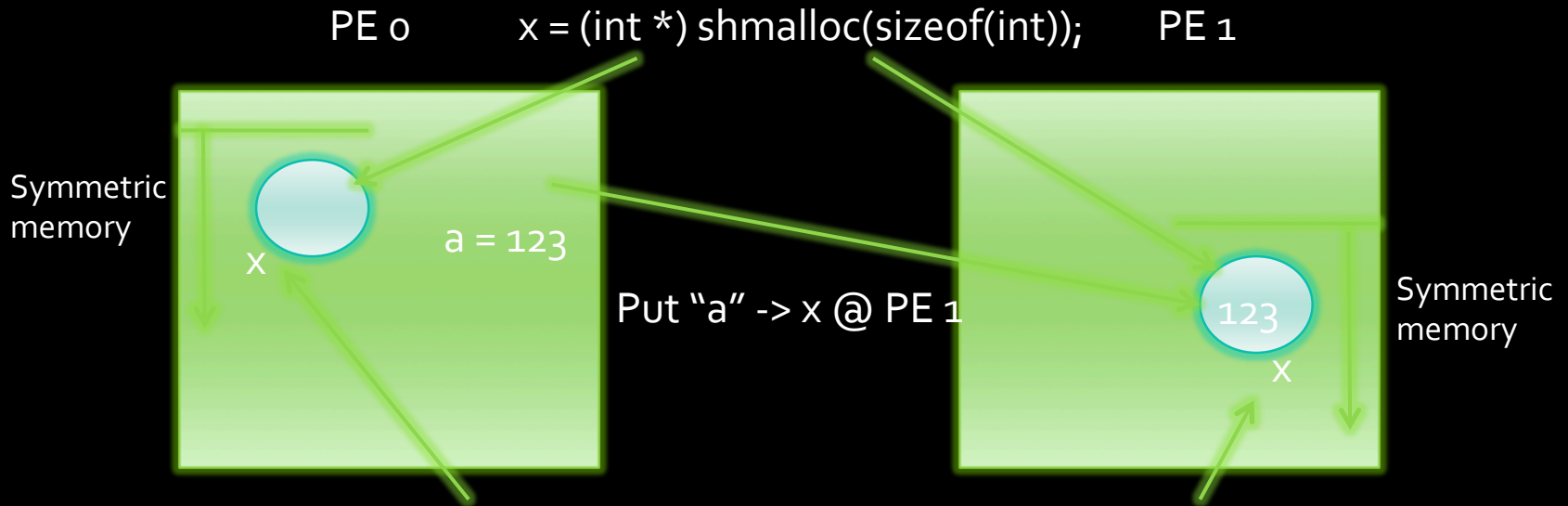


Figure 2. Dynamic allocation of Symmetric Data

Introducing OpenSHMEM Symmetric Variables



Same offset, but at potentially different addresses

```
if (_my_pe() == 0) {  
    shmem_int_put (dest = x, src = a, len = 1, pe = 1);  
}  
// has "a" arrived yet? We don't know...
```

Introducing OpenSHMEM

History of SHMEM

- History

- Cray SHMEM
- SGI SHMEM
- Quadrics SHMEM
- Others
 - HP SHMEM, IBM SHMEM (used internally only)
 - GPOSHMEM

SHMEM was not defined by any one standard

- OpenSHMEM



Open Source Software
Solutions Inc.



Introducing OpenSHMEM

Need for Standardization

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

SGI	Quadrics	SiCortex
<code>start_pes(int npes)</code>	<code>start_pes(int npes)</code> <code>shmem_init(void)</code>	<code>start_pes(int npes)</code> NO-OP <code>shmem_init(void)</code>

Introducing OpenSHMEM

Need for Standardization

- Here's "Hello World" again on an SGI Altix

```
#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

Towards OpenSHMEM

- Standardization & community
 - 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

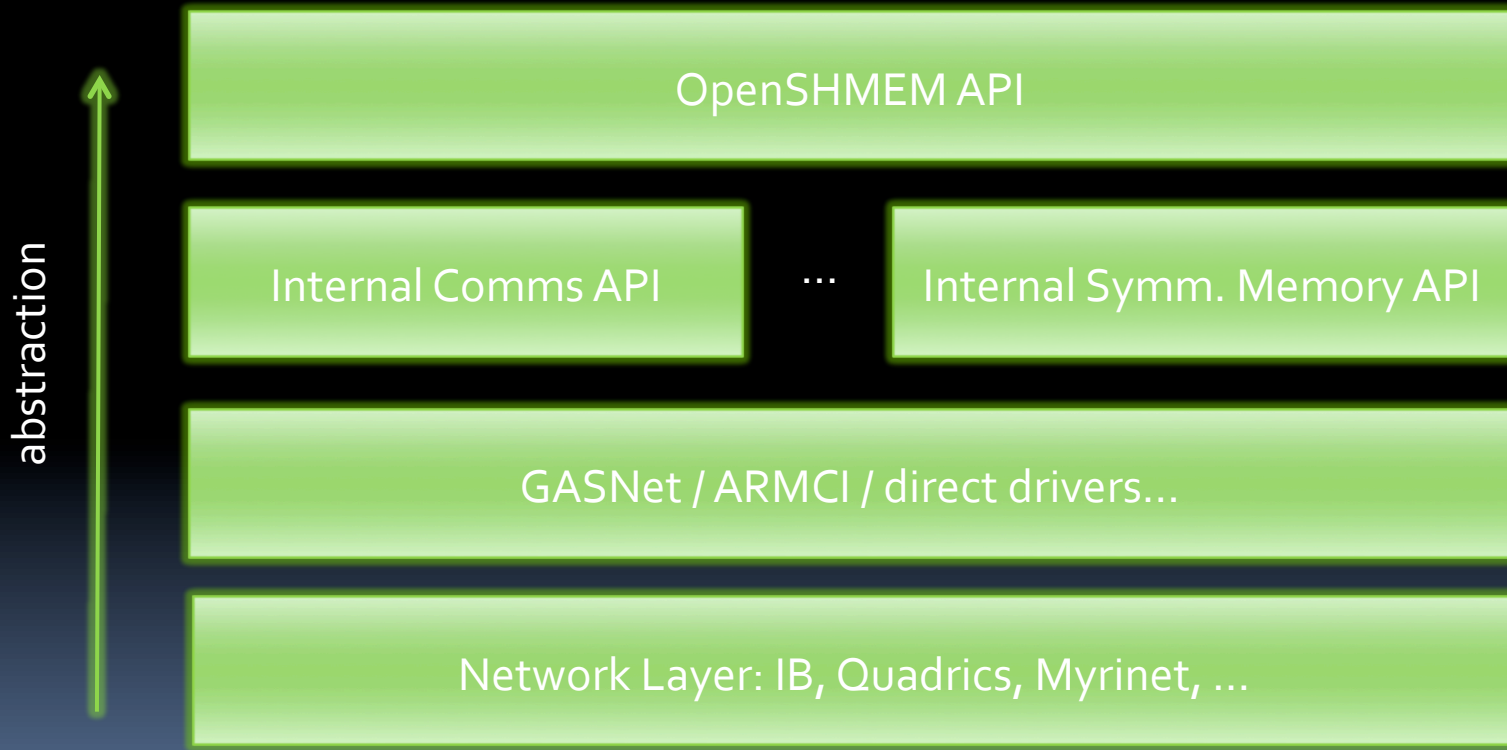
Towards OpenSHMEM

- SGI version as a starting-point
 - SGI (Altix) implementation as reference
 - With editorial rewrites
 - OpenSHMEM Specification 1.0 (released !)
- Develop new specification as version 2.0
 - Solicit new ideas from community
 - What features should be changed/added?
- Reference implementations
 - New OpenSHMEM written by University of Houston
 - Baseline for future development
 - 1.0 version complete and moving towards 2.0

Introducing OpenSHMEM Reference Implementation



OpenSHMEM programmer



University of Houston: Implementation Structure



Introducing OpenSHMEM V&V

- Validation & Verification
 - Build core tests of correctness
 - Created by University of Houston
 - And of performance
 - To compare implementations of collective algorithms
 - For tuning of underlying libraries/ transports

Introducing OpenSHMEM

Outreach and Participation

- OpenSHMEM web site (LIVE!)
 - <http://www.openshmem.org/>
 - Community Wiki
 - Documentation: FAQ, cheatsheet, specification
 - Training material / tutorials
 - Software downloads
 - Source code of OpenSHMEM versions
 - Validation and Verification Suite
 - Sample programs
- Conferences/workshops
 - PGAS₁₀, SC₁₀, ICS₁₁, PGAS₁₁, SC₁₁
- Mailing list

Introducing OpenSHMEM At SC11

- HERE and NOW
 - SC11 (Seattle, November 12-18)
 - Birds of a Feather meeting
 - OpenSHMEM
 - Wednesday, November 16th, 5:30pm – 7:00pm (TCC 203)
 - Current exhibition booth presence
 - PGAS (#124)
 - Oak Ridge National Laboratory (#1831)
 - Gulf Coast Academic Supercomputing (#3009)
 - SGI (#1841)

Introducing OpenSHMEM

Get Involved

- OpenSHMEM mailing list for discussions and contributions can be joined at <https://email.ornl.gov/mailman/listinfo/openshmem>
- OpenSHMEM web site
 - <http://www.openshmem.org/>
- SC11 Birds of a Feather
 - Wednesday, November 16th, 5:30pm – 7:00pm (TCC 203)
- Come talk with the OpenSHMEMers here
 - Lauren Smith, Tony Curtis, Swaroop Pophale