

XcalableACC

Abstract

XcalableACC (XACC) is a PGAS language for accelerated clusters, which is a directive-based language extension of C and Fortran.

- High productivity by directives
- High performance by direct communication between accelerators

Components

- XcalableMP (XMP) for distributed-memory parallelism

XcalableMP XMP is a directive-based language extension of C and Fortran for distributed-memory parallel systems

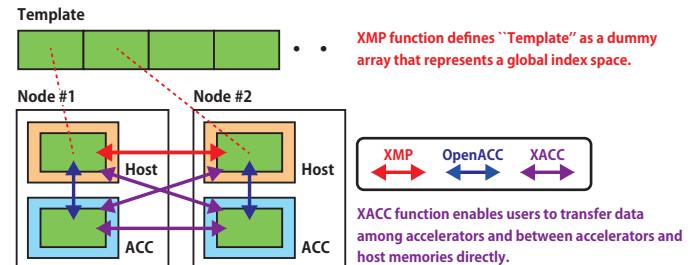
- OpenACC for offloading works for accelerators

OpenACC
DIRECTIVES FOR ACCELERATORS

OpenACC is another directive-based language extension for heterogeneous CPU/Accelerator systems

- XACC extensions for communication of data on accelerators

Memory Model



Omni XcalableACC Compiler

- <http://omni-compiler.org>
- Open-source software under LGPL
- As a reference implementation developed by RIKEN AICS and Center for Computational Sciences in University of Tsukuba



Evaluation

Productivity

- Comparison of the source lines of code (SLOC)
 - XACC v.s. OpenACC + MPI as a typical programming
 - SLOCs of XACC are smaller than those of OpenACC + MPI
- Example : HIMENO Benchmark
 - Stencil application
 - Evaluates the performance of incompressible fluid analysis code
 - Programmer only adds XMP and OpenACC directives into the sequential Himeno benchmark

```

static float p[MIMAX][MJMAX][MKMAX];
#pragma xmp template t(0:MKMAX, 0:JMAX, 0:MIMAX)
#pragma xmp nodes n(NDV, NDX)
#pragma xmp distribute t(*, block, block) onto
#pragma xmp align [k][j][i] with t(i, j, k) :: p, ..
#pragma xmp shadow p[1:1][1:1][0]
...
#pragma acc data copy(p, ...)
{
    for(n=0; n<nn; ++n)
    ...
    #pragma xmp loop (k,j,i) on t(k,j,i)
    #pragma acc parallel loop collapse(2) reduction(:gosa) ...
        for(i=1; i<imax-1; ++i){
            for(j=1; j<jmax-1; ++j){
                #pragma acc loop vector reduction(:gosa) private(s0, ss)
                    for(k=1; k<kmax-1; ++k){
                        s0 = p[i+1][j][k] * ...
                        ss = ...
                        gosa += ss * ss;
                    }
                }
            #pragma xmp loop (k,j,i) on t(k,j,i)
            #pragma acc parallel loop collapse(2) ...
                for(i=1; i<imax-1; ++i){
                    for(j=1; j<jmax-1; ++j){
                        #pragma acc loop vector
                            for(k=1; k<kmax-1; ++k){
                                p[i][j][k] = wrk2[i][j][k];
                            }
                        }
                    ...
                    #pragma xmp reflect (p) acc
                    ...
                /* end n loop */
            /* end of acc data */
        }
    }
}

```

Define XMP distributed arrays

Copy XMP distributed arrays to accelerator

Parallelize loop
OpenACC directive parallelizes the loop statement parallelized by XMP directive

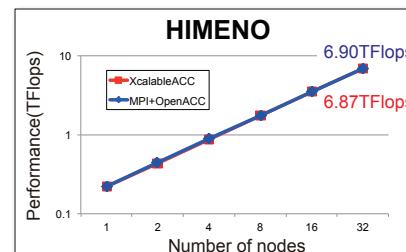
Exchange halo region
When "acc" clause is specified in XMP communication directive, data on accelerators is transferred.

Source Lines of Codes

	HIMENO	NPB-CG	HPL	STREAM
XcalableACC (a, b)	198(34.9)	609(48.20)	437(54.6)	90(8.3)
MPI + OpenACC (b)	328(13)	772(24)	—	—
XcalableMP(a)*1	—	—	343(54)	78(8)

(a) Number of XMP directives
(b) Number of OpenACC directives
*1 Only CPU is used in performance evaluation

Performance on HA-PACS/TCA

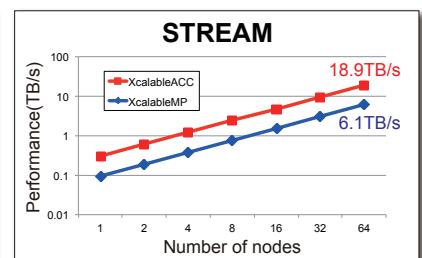
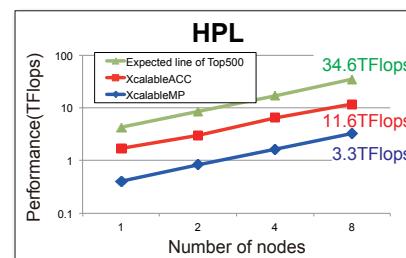
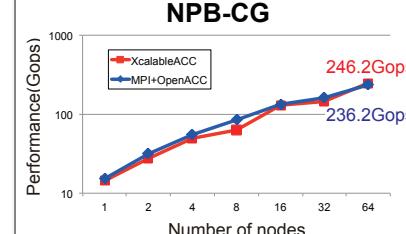


HA-PACS/TCA cluster system



<http://ccs.tsukuba.ac.jp/eng/research-activities/projects/ha-pacs/>

- Ivy Bridge E5-2680v2, 10Cores x 2Sockets
- DDR3 128GB (59.7GB/s x 2, NUMA)
- NVIDIA K20X (D.P. 1.31TFlops) x 4 GPUs
- GDDR5 6GB (250GB/s)
- InfiniBand 4xQDR x 2rails, 8GB/s
- MVAPICH2-GDR 2.1a, gcc-4.4.7, etc...



Reference

- Tetsuya Odajima, et al. ``Hybrid Communication with TCA and InfiniBand on A Parallel Programming Language XcalableACC for GPU Clusters," Workshop Series on Heterogeneous and Unconventional Cluster Architectures and Applications (HUCAA), IL, USA, Sep., 2015.
- Masahiro Nakao, et al. ``XcalableACC: Extension of XcalableMP PGAS Language using OpenACC for Accelerator Clusters," Workshop on accelerator programming using directives (WACCPD), New Orleans, LA, USA, Nov., 2014.