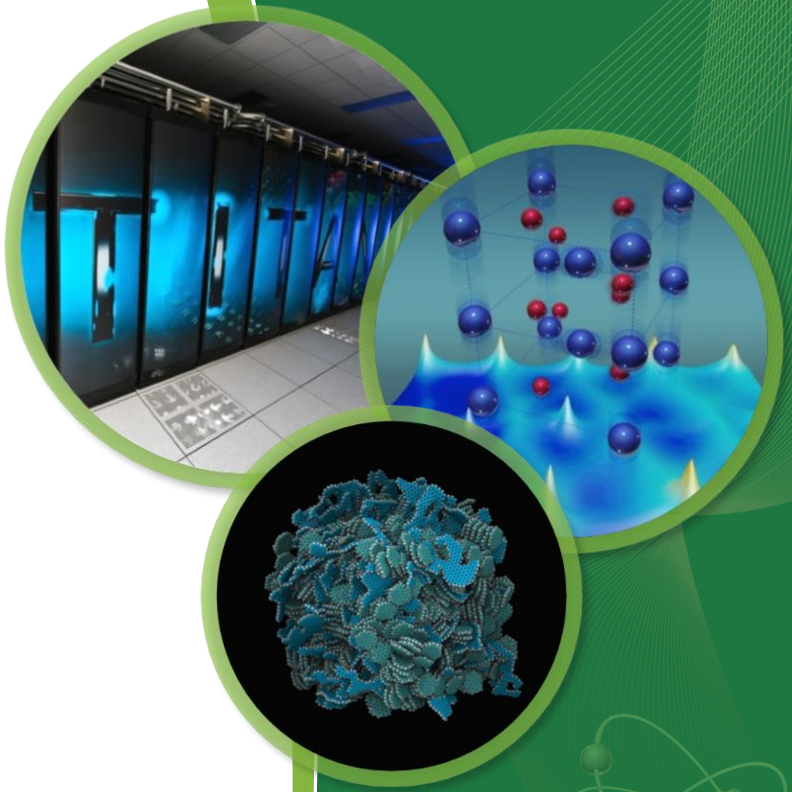
Clacc: OpenACC Support for Clang/LLVM

Joel E. Denny

Seyong Lee

Jeffrey S. Vetter



Presented to

EuroLLVM 2018
Bristol
17 April 2018

Future Technologies Group ornl

http://ft.ornl.gov vetter@computer.org

Oak Ridge National Laboratory

Background | Extreme Heterogeneous Computing

- Problem
  - Heterogeneous/manycore processors becoming de facto architectures in HPC
  - Architectures diverse in functionality, performance, programmability, scalability
  - Architectures once required dedicated/proprietary programming models (CUDA, VHDL)
  - Key problem: lack of functional portability
  - Rapidly evolving architectures often lacked robust programming ecosystems
- Solution
  - Directive-based, accelerator programming models attempt to provide functionally portable programming solutions for heterogeneous computing
  - Provide very high-level abstractions over complexity of underlying architectures and low-level programming languages like CUDA and OpenCL
  - Examples: OpenMP and OpenACC

System attributes	NERSC Now	OLCF Now	ALCF Now	NERSC Upgrade	OLCF Upgrade	ALCF Upgrades	
Planned Installation	Edison	TITAN	MIRA	Cori 2016	Summit 2017-2018	Theta 2016	Aurora 2018-2019
System peak (PF)	2.6	27	10	> 30	150	>8.5	180
Peak Power (MW)	2	9	4.8	< 3.7	10	1.7	13
Total system memory	357 TB	710TB	768TB	~1 PB DDR4 + High Bandwidth Memory (HBM)+1.5PB persistent memory	> 1.74 PB DDR4 + HBM + ~~2.8~~ ~~3.7~~ ~7 PB persistent memory	>480 TB DDR4 + High Bandwidth Memory (HBM)	> 7 PB High Bandwidth On-Package Memory Local Memory and Persistent Memory
Node performance (TF)	0.460	1.452	0.204	> 3	> 40	> 3	> 17 times Mira
Node processors	Intel Ivy Bridge	AMD Opteron Nvidia Kepler	64-bit PowerPC A2	Intel Knights Landing many core CPUs Intel Haswell CPU in data partition	Multiple IBM Power9 CPUs & multiple Nvidia Voltas GPUS	Intel Knights Landing Xeon Phi many core CPUs	Knights Hill Xeon Phi many core CPUs
System size (nodes)	5,600 nodes	18,688 nodes	49,152	9,300 nodes 1,900 nodes in data partition	~3,500 nodes	>2,500 nodes	>50,000 nodes
System Interconnect	Aries	Gemini	5D Torus	Aries	Dual Rail EDR-IB	Aries	2nd Generation Intel Omni-Path Architecture
File System	7.6 PB 168 GB/s, Lustre®	32 PB 1 TB/s, Lustre®	26 PB 300 GB/s GPFS™	28 PB 744 GB/s Lustre®	120 PB 1 TB/s GPFS™	10PB, 210 GB/s Lustre initial	150 PB 1 TB/s Lustre®

OAK RIDGE National Laboratory

Background | What is OpenACC?

- https://www.openacc.org/
- Launched in 2010 as a portable programming model for heterogeneous accelerators
- Consists of compiler directives, library routines, and environment variables
- Programmer provides hints, or "directives", identifying areas of code to accelerate
- Aimed at incremental development of accelerator code

```
// example from https:///www.openacc.org
#pragma acc data copy(A) create(Anew)
while ( error > tol  &&  iter  <  iter_max )  {
  error = 0.0;
#pragma acc kernels {
#pragma acc loop independent collapse(2)
  for (  int  j = 1; j < n-1;  j++ )  {
    for (  int  i = 1; i < m-1; i++ )  {
       Anew [j] [i] = 0.25
         * ( A [j] [i+1] + A [j] [i-1] +
             A [j-1] [i] + A [j+1] [i]);
       error = max ( error, fabs (Anew [j] [i]
         - A [j] [i]));
      }
    }
  }
}
```


OAK RIDGE National Laboratory

Status | OpenACC Compilers

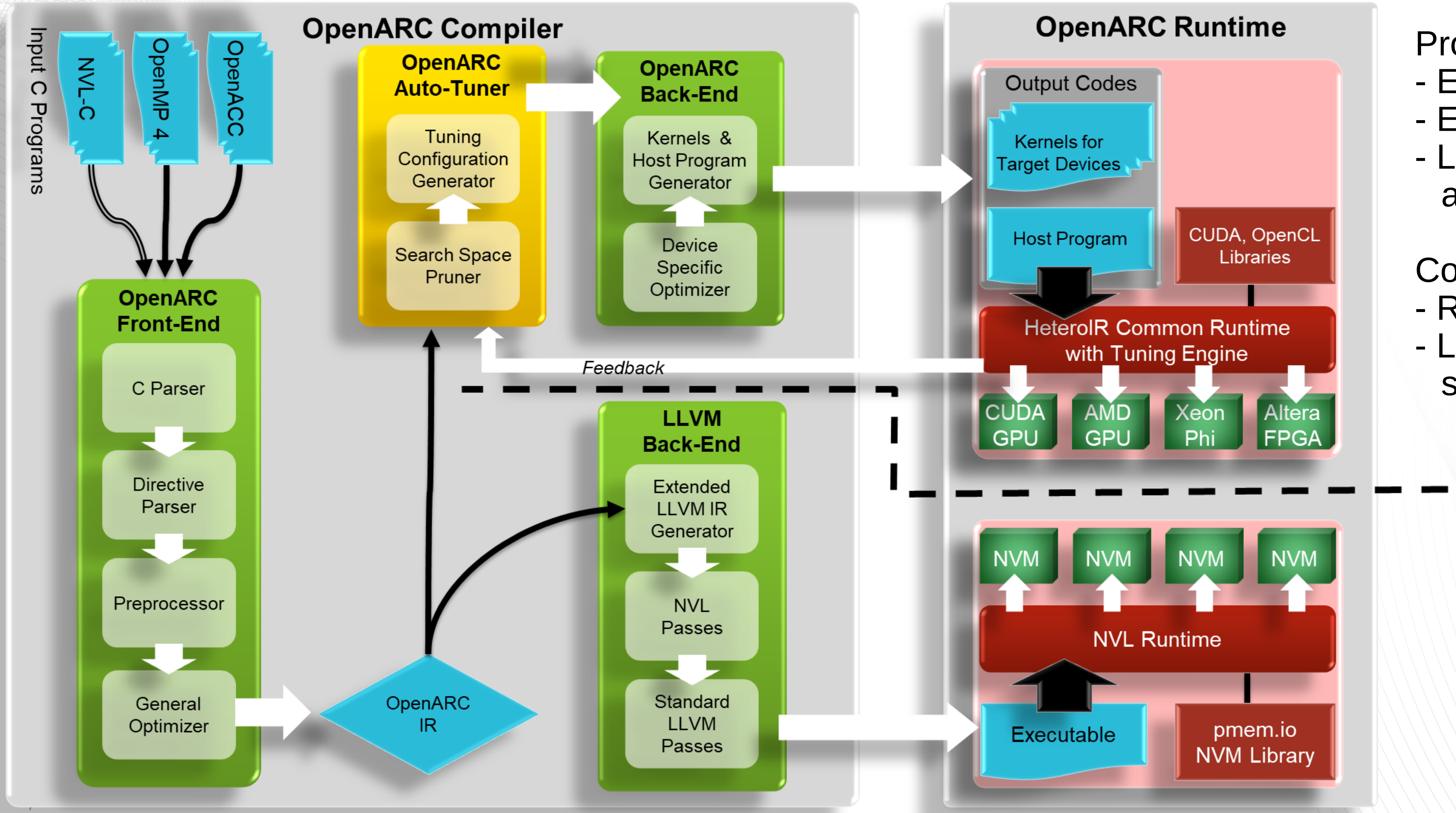
- Commercial
  - PGI, Cray
  - National Supercomputing Center in Wuxi
- Open Source
  - GCC 7 (initial support for 2.5)
- Academic
  - OpenARC (ORNL)
  - Omni compiler project (RIKEN, Univ. Of Tsukuba)
  - OpenUH (University of Houston, Stony Brook University)
  - ROSEACC (LLNL, University of Delaware)

Compiler Versions Used
GNU 6.0.0-20160415
GNU 6.3-20170303
PGI 16.10
PGI 17.3
More recent results are under development.

Architecture	PGI pass rate	GNU pass rate
K20	175/177	112/177
K80	175/177	113/177
Ivy Bridge	171/177	154/177
Bulldozer	172/177	157/177

Friedline K., Chandrasekaran S., Lopez M.G., Hernandez O. (2017) OpenACC 2.5 Validation Testsuite Targeting Multiple Architectures. In: Kunkel J., Yokota R., Taufer M., Shalf J. (eds) High Performance Computing. ISC High Performance 2017. Lecture Notes in Computer Science, vol 10524. Springer, Cham

Prior Experiences with OpenARC

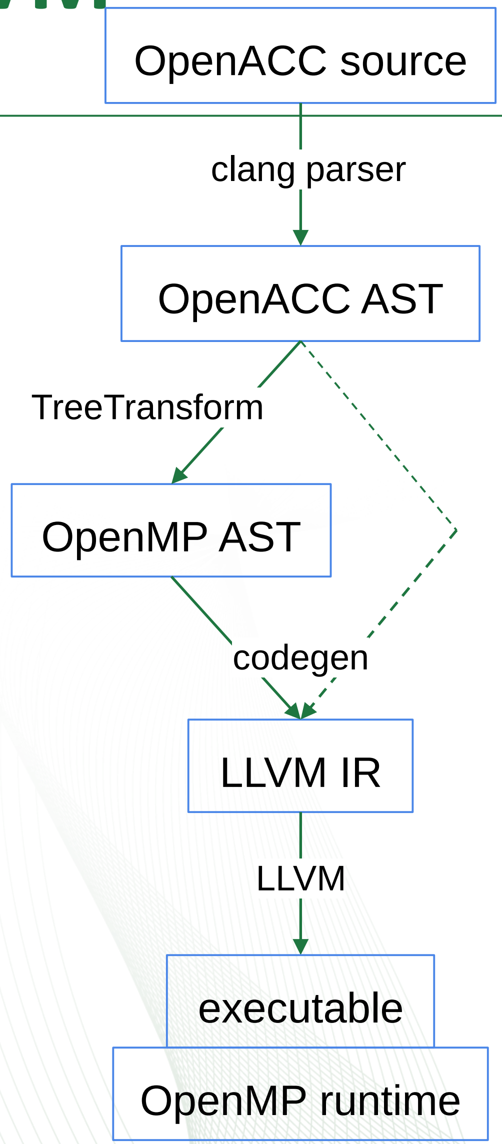


Pros:
- Easy src-to-src
- Easy transformations
- Leverage backend if available (e.g., CUDA)

Cons:
- Research compiler
- Limited language support

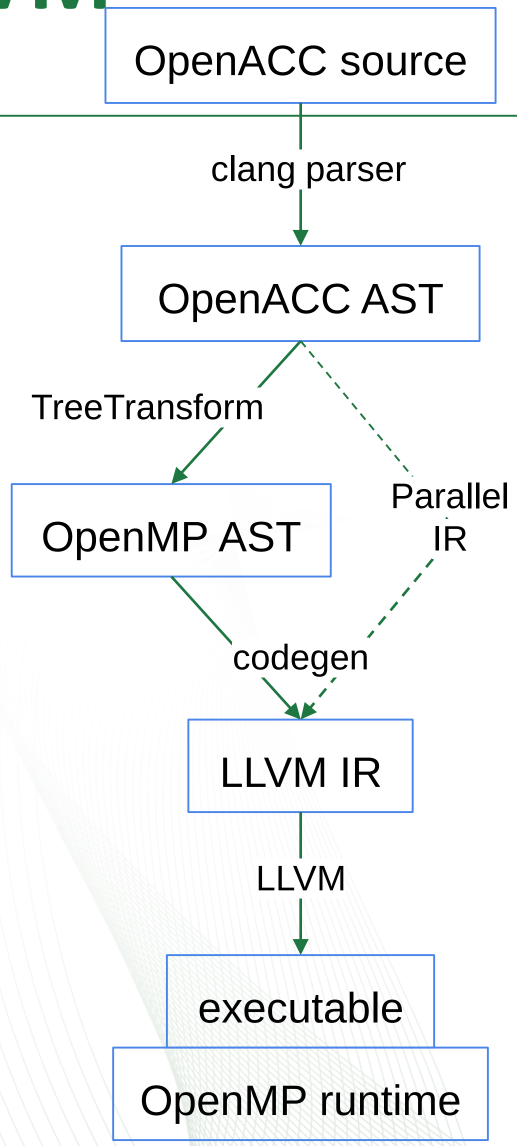
Clacc | OpenACC Support for Clang/LLVM

- Goals
  - Contribute production-quality OpenACC compiler support to clang/LLVM
  - Enable construction of source-level OpenACC tools built on clang
    - Pretty printers, analyzers, lint tools, debugger extensions, editor extensions, etc.
- Design
  - Key: translate (lower) OpenACC to OpenMP
    - Builds on clang's existing OpenMP compiler/runtime support
    - OpenACC is descriptive, OpenMP is prescriptive
      - Directive mapping is not one-to-one: analysis required
  - AST transformation
    - OpenACC AST needed for second goal
    - Maximize OpenMP implementation reuse: AST transformation
    - Clang AST is immutable: use TreeTransform to create modified copy
  - Began design discussions within clang community last year
- Funded by Exascale Computing Project under ST PROTEAS



Clacc | OpenACC Support for Clang/LLVM

- Design alternative: target future LLVM IR parallel extensions
  - For lowering to OpenMP: use LLVM IR analyses
  - Conflicts with source-to-source feature (feedback to AST?)
- Status
  - Early prototyping phase
    - Prescriptive interpretation of OpenACC in C for correctness
    - Design still evolving: continue discussion with clang devs
    - Upstreaming other clang/LLVM improvements as encountered
  - Clang/LLVM OpenMP offloading implementation under active upstreaming/development
  - Possible feature: automated translation from OpenACC source to OpenMP source
    - Permanent migration to OpenMP
    - Aid in understanding compiler decisions (imagine editor extension)
    - OpenMP source-level tools applied to OpenACC



Comments, questions, code → vetter@computer.org