



GCC Update

OpenACC Summit 2021

GCC Update

OpenACC Summit 2021

Pre-recorded, because... any day now...



GCC/OpenACC

GCC Branches

- GCC 11: current stable release series
 - OpenACC 2.6 support
 - Code offloading to AMD (GCN) and Nvidia (nvptx) GPUs
- og11 development branch (public)
 - GCC 11 release branch plus new features:
 - Backlog inherited from og10 branch
 - Ongoing development
 - Provides early access to GCC 12+ features
- GCC 12: next major release due spring 2022
 - Ongoing development



GCC/OpenACC

Last Year: Development Focus

- OpenACC 'kernels' work
 - Compiler "magic":
 - Array access delinearization
 - Scalar data privatization
 - Analyze 'loop' constructs with 'auto' clause, decide 'seq' vs. 'independent'
 - See talk at LPC¹, GNU Tools Track: *OpenACC "kernels" improvements*
 - <https://linuxplumbersconf.org/event/11/contributions/998/> (Frederik Harwath)
 - Work done on separate development branch; currently integrating into og11 branch
- Upstreaming (and revision) of existing og10 development branch work into GCC mainline



¹ *Linux Plumbers Conference 2021*, <https://linuxplumbersconf.org/>, virtual, week of 2021-09-20

GCC/OpenACC Last Year: Upstreaming



Integrated into GCC mainline (for GCC 12):

- Decompose OpenACC 'kernels' constructs into parts, a sequence of compute constructs
- OpenACC worker parallelism for AMD GPUs
 - Execution state changes (neutering/broadcasting) as a GCC middle end transformation
 - Different approach from nvptx where it all happens in the back end
- Interactive debugging (GDB; ROCGDB) for GCN offloading
 - See demo at LPC¹, GNU Tools Track: *Debugging offloaded kernels on AMD GPUs*
 - <https://linuxplumbersconf.org/event/11/contributions/997/> (Andrew Stubbs)
 - See also at LPC¹, GNU Tools Track: *DWARF extensions for optimized SIMT/SIMD (GPU) debugging*
 - <https://linuxplumbersconf.org/event/11/contributions/1012/> (AMD: Tony Tye, Zoran Zaric)



GCC/OpenACC

Last Year: Upstreaming, continued



Integrated into GCC mainline (for GCC 12), continued:

- Bug fixing (such as OpenACC specification adherence), for example:
 - Data privatization/sharing at the gang level: use GCN LDS, nvptx `'.shared'` memory
 - OpenACC/Fortran: strided array sections and components of derived-type arrays
 - OpenACC `'async'` correctness
 - Keep up with breaking changes of LLVM/HSA (GCN), PTX/CUDA (nvptx)
 - The usual miscellanea
- Optimizations: middle end as well as GCN, nvptx back ends
- Diagnostics: `'-Wopenacc-parallelism'` to diagnose potentially suboptimal choices of OpenACC parallelism



GCC/OpenACC

Next Steps

- Further revision of existing og11 branch changes for integration into GCC mainline
 - Still for GCC 12 until cut-off date (winter)
- Same for current OpenACC 'kernels' work
- OpenACC 2.7 features (not yet scheduled)
- Other items as prioritized by customers
- ... and: continue working with OpenACC Technical Committee



Thanks!

Contact: <thomas_schwinge@mentor.com>

Disclaimer

© Siemens 2021

Subject to changes and errors. The information given in this document only contains general descriptions and/or performance features which may not always specifically reflect those described, or which may undergo modification in the course of further development of the products. The requested performance features are binding only when they are expressly agreed upon in the concluded contract.

All product designations may be trademarks or other rights of Siemens AG, its affiliated companies or other companies whose use by third parties for their own purposes could violate the rights of the respective owner.