What about MPI + LLVM

A case for dynamic recompilation and runtime specialization
Parallel MPI Communication – Ping Pong

Schneider, Kjolstad, TH: MPI Datatype Processing using Runtime Compilation, EuroMPI’13
Parallel MPI Communication – Reality

Datatypes are interpreted (OMG!)

We implement them with LLVM JIT

10% of ping-pong performance

Schneider, Kjolstad, TH: MPI Datatype Processing using Runtime Compilation, EuroMPI’13
Vector Packing Performance (LLVM 3.1 –O3)

In-cache measurement on AMD Interlagos CPU (Blue Waters)

HVector(2,1,6144) of Vector(8,8,32) of Contig(6) of MPI_FLOAT

This datatype is used by the Quantum-Chromodynamics code MILC [1]

Irregular Packing Performance (LLVM 3.1 –O3)

Hindexed DDT with random displacements

Schneider, Kjolstad, TH: MPI Datatype Processing using Runtime Compilation, EuroMPI’13
Our Observations

- **How easy was it to use LLVM?**
  - It was ok (very easy to get started with something)

- **What made it hard to use?**
  - We got caught in the transition to MC-JIT
  - Would be nice if LLVM could come as a library as default option
  - Issues with static linking on Cray (not LLVM’s fault)

- **What worked well?**
  - Very easy to get started! Initial pass after ½ day.

- **What needs to be improved?**
  - Fortran support (dragonegg generates messy IR)
  - Alias analysis
  - Performance of the passes (for JITing)
  - For Christmas: superoptimization and abstract interpretation
Backup Slides
LLVM-JIT DDTs vs. manual packing (and Cray)!

Schneider, Kjolstad, TH: MPI Datatype Processing using Runtime Compilation, EuroMPI’13
Performance Study: MILC

- Packing faster, but commit is now slower

Schneider, Kjolstad, TH: MPI Datatype Processing using Runtime Compilation, EuroMPI’13
Performance Study: MILC

Most datatypes have to be reused 180-5000 times

Schneider, Kjolstad, TH: MPI Datatype Processing using Runtime Compilation, EuroMPI’13