Linking LTO and Make

John Ravi
jjravi@ncsu.edu

Google Summer of Code
Outline

• GCC and Link-time optimization
• GNU Make Jobserver
• GCC Jobserver Integration
• Final Steps
GCC LTO Parallelism (WHOPR)

file1.c → file1.o
file2.c → file2.o
file3.c → file3.o

ld / gold → a.out
LTO plug-in
Whole Program Analysis
Local opt.
Local opt.
Local opt.
GNU Make Jobserver

- `make -j N`, spawns at most $N$ concurrent jobs
- Uses $N$ tokens
GCC LTO Modifications

- Added libcody
- Set an environment variable to utilize jobserver
- LTO Compilation requests are sent to the jobserver
GNU Make Modifications

• Added libcody (ported some make internals to C++)
• Connects to GCC and can spawn LTO compilations
Final steps

- Enable and verify performance with parallelism
- Backport libcody to support C
Thank You

gcc:devel/lto-offload
github.com/gcc-mirror/gcc

Nathan Sidwell’s communication library:
github.com/urnathan/libcody

make:lto-hack
github.com/jjravi/make