GCC LTO BOF Agenda

GCC Summit, 2010

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs
- We need to switch WHOPR and LTO (for testing at least)

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs
- We need to switch WHOPR and LTO (for testing at least)
- `-flinker-plugin` should probably become default

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs
- We need to switch WHOPR and LTO (for testing at least)
- `-flinker-plugin` should probably become default
- We probably should default to slim files

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs
- We need to switch WHOPR and LTO (for testing at least)
- `-flinker-plugin` should probably become default
- We probably should default to slim files
- Get LTO section versioning right so mismatches are reported nicely

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs
- We need to switch WHOPR and LTO (for testing at least)
- `-flinker-plugin` should probably become default
- We probably should default to slim files
- Get LTO section versioning right so mismatches are reported nicely
- Get command line options encoding independent of the autogenerated enum

Command line/user experience issues

- `-fwhopr` should become `-flto`.
- What defaults?
  - 32 partitions
  - Auto-detect parallel make, otherwise single thread compilation
  - Disable WHOPR for very small programs
- We need to switch WHOPR and LTO (for testing at least)
- `-flinker-plugin` should probably become default
- We probably should default to slim files
- Get LTO section versioning right so mismatches are reported nicely
- Get command line options encoding independent of the autogenerated enum
- LTO plugin should not hold stderr output

Bug reporting problems

- “Building OOo with -flto makes GCC to segfault”

Bug reporting problems

- “Building OOo with -flto makes GCC to segfault”
- It is very non-trivial to find the minimal set of source files
  - Collect2 won’t even start LTO when some symbols are missing
    (unless `-r -nostdlib` is used)
  - Whole program assumption makes minimizing hard — incomplete testcases tends to be fully optimized out
  - Some Mozilla PRs took me a day to hand reduce
  - Resulting testcases are still huge and takes ages to delta

Bug reporting problems

- “Building OOo with -flto makes GCC to segfault”
- It is very non-trivial to find the minimal set of source files
  - Collect2 won’t even start LTO when some symbols are missing
    (unless `-r -nostdlib` is used)
  - Whole program assumption makes minimizing hard — incomplete testcases tends to be fully optimized out
  - Some Mozilla PRs took me a day to hand reduce
  - Resulting testcases are still huge and takes ages to delta
- Object formats are unsafe and can’t be shipped

LTO streaming issues

- GIMPLE + types definition
  - unbloat on-disk format
  - GIMPLE front-end for sane readable representation

LTO streaming issues

- GIMPLE + types definition
  - unbloat on-disk format
  - GIMPLE front-end for sane readable representation
- Define section formats, tag with major/minor versions, attempt to keep LTO stable across function sections

LTO streaming issues

- GIMPLE + types definition
  - unbloat on-disk format
  - GIMPLE front-end for sane readable representation
- Define section formats, tag with major/minor versions, attempt to keep LTO stable across function sections
- Make LTO/WHOPR tolerant to missing IPA pass summaries

LTO streaming issues

- GIMPLE + types definition
  - unbloat on-disk format
  - GIMPLE front-end for sane readable representation
- Define section formats, tag with major/minor versions, attempt to keep LTO stable across function sections
- Make LTO/WHOPR tolerant to missing IPA pass summaries
- Implement dumping tools

LTO streaming issues

- GIMPLE + types definition
  - unbloat on-disk format
  - GIMPLE front-end for sane readable representation
- Define section formats, tag with major/minor versions, attempt to keep LTO stable across function sections
- Make LTO/WHOPR tolerant to missing IPA pass summaries
- Implement dumping tools
- Slim object files

Compiler option mismatches

- `-g` + `-g0` LTO $\rightarrow$ crash

Compiler option mismatches

- `-g` + `-g0` LTO $\rightarrow$ crash
- Categorize option as
  - processed by FE or Early optimization
  - global (used by IPA passes only)
  - processed by late compilation only We could use option attributes here...

Compiler option mismatches

- `-g` + `-g0` LTO $\rightarrow$ crash
- Categorize option as
  - processed by FE or Early optimization
  - global (used by IPA passes only)
  - processed by late compilation only We could use option attributes here...
- Push as many of global flags into the IL as possible

Compiler option mismatches

- `-g` + `-g0` LTO $\rightarrow$ crash
- Categorize option as
  - processed by FE or Early optimization
  - global (used by IPA passes only)
  - processed by late compilation only We could use option attributes here...
- Push as many of global flags into the IL as possible
- Should early optimizations be independent of `-Os`/`-O2` difference?

Debug info issues

- Streaming early debug info?
- Reducing memory footprint of types/decls by understanding what really is important

Missing parts of IPA infrastructure

- scalable points-to
- datastructure layout opts + type escape
- may edges in the callgraph
- devirtualization is important and broken
- function cloning pass
- inlining heuristics can always be improved
- Merging of identical functions can be implemented at IPA level, too.
- More aggressive privatization of functions/vars?

Incremental WHOPR

- WPA stage ships every function with summaries and optimization decision
- Functions hat did not change might be re-used
- WPA still needs to get faster

LTO plugin issues

- Currently Gold seems to behave funny about `PREVAILED` wrt `PREVAILED_IRONLY`
- Plugin interface disallows
- Linker plugin **should** be the default
- For sane support of slim LTO objects we need plugin+`collect2` to claim them even without `-flto`
- Plugin+`collect2` needs to understand the difference of fat and slim objects

Real symbol tables

- Symbol table entry
  - Identifier pointer
  - Linkage info (like in ELF symbol table)
  - Optional declaration pointer
- Inheritance tree with
  - cgraph node
  - cgraph node with body
  - varpool node
  - aliases and alternate entry points (thunks)

Profile instrumentation at linktime only

- Would be sweet to not force user to recompile, only relink
- Will allow easy indirect call removal across unit boundaries
- Problems with WHOPR
- need to integrate gcov style CFG summaries + solver $\rightarrow$ `libgcovslver`?
- How much can we borrow from LIPO?