11 Nov 2026

Agenda:
Andreas:

- Target independent usages of LBR and whether these should be changed to `brstack`, such as profile headers with the `no_lbr` label. How does this change affect backward compatibility? There are NFC changes that are currently in review ([NFC][BOLT] Clean up brstack wording).

Alexandros:

- BTI hardening in clustered relaxation: https://github.com/llvm/llvm-project/pull/234122
- Enable clustered relaxation by default: https://github.com/llvm/llvm-project/pull/234932

Asher:

- Perhaps worth nudging the infrastructure team on the subdomain side.Documentation has been moved by Andreas, but http://bolt.llvm.org/ still points to https://bugs.llvm.org/. I assume this is due to missing Apache configuration. Discord message in the infrastructure channel got lost among others.

Kostas:

- PowerPC latest progress and metrics
- PowerPC64_BOLT_status__11_Nov_2026

Amir:

- China-friendly time, or separate meeting(s)?

Notes
Amir:

- Rafael on recharge

Pavel

- Chat with Peter Hosek: migrating server, out of disk. BOLT subdomain pending infra migration. No ETA

Amir:

- **Moving the meetings to 7AM PT, 3PM London, 6PM Moscow, 11PM Shanghai**

Alexandros:

- Experimental -> clustered relaxation renaming, +alias
- Enabling by default
- BTI patch: Gergely looking from BTI angle
    - Compact code model as base, local branches relaxed further. If bti inserted later, may invalidate earlier relaxation. Bti is needed for indirect calls in landing pads. Use “ladders” - chained trampolines
    - Yongkang: is ladder used only for bti or default?
    - Alexandros: using ladders now by default (?)
    - Small binary: 1-2 clusters in compact code model, max 1 hop
    - Adrp stubs in compact code model? Probably not
    - Yongkang to take a look

Andreas

- No_lbr

- Amir:
    - Context specific: brstack or branch/basic sample
    - Use aliases for backwards compat, produce new by default
    - Migration to drop lbr completely

Kostas

- PowerPC update presentation
    - Llvm llc as target
    - Indirect call bctrl
    - Profile?
    - Hypotheses for why the code is slower, heatmap, boltdiff
    - Alexey: try to increase threshold for jump tables (to increase coverage of rewritten code)
      Pavel: GCC is more aggressive with jump tables (lower threshold)
      Alexey: -march specific basic block alignment - as perf improvement opportunity

Pavel

- Linker optimizations corrupting/invalidating relocations for BOLT.
    - Expect cooperating toolchain
    - Keep track of options/optimizations to disable, no-relax would be one of those
- AI in BOLT
    - Full rewriting in agentic loop. Review is the bottleneck.
    - Alexey: AI code is harder to understand/maintain, should reduce complexity. Post llvm21 migration will focus on full rewriting. Complexity due to NextAvailableAddress, should only care about it during emission.
    - AI-generated issues. Sometimes hallucinated issues/root causes. DWARF rewriting issue as prime example - still useful due to data sharing limitations
        - Improve debug info via validation

Alexey

- Constants in assembly code, attribution problem. Don’t blow up heuristics.
- Pavel: assembly in chromium. Skip assembly functions
- Yongkang: chromium support?
    - Skip-funcs and use-old-text, may rewrite

Oct 14, 2026

Agenda:
Adam:

- I'd like to present four design questions arising from recent changes to AArch64 branch relaxation in BOLT: adopting the experimental path, using the full code layout, and correctly handling tail calls and skipped functions.

Pavel:

- Linker optimisations and what BOLT should rely on to handle such transformations correctly: retained relocations, instruction patterns, additional metadata. Potentially related approach of reconstructing references when relocations are unavailable
- Leveraging AI in BOLT: AI-assisted bug triage, reproducer generation, fixes, including DWARF rewriting issues and work on full binary rewriting mode. Implications of the broader LLVM RFC on AI-generated communication

---

Notes
Paschalis:

- Update about leaving ARM
- Continue with LLVM, but not BOLT
- Maintainer role: to Pavel and Alexandros

Adam:

- LongJmp issues:
    - Relaxation path is superior, promote from experimental status
    - All inter-func branches are tail calls
    - x16/17 clobber
    - Skipped functions
- Presentation

Rafael:

- +1 on switching experimental relax to default
- Keep existing longjmp as fallback, remove later
- BOLT heuristics depend on binaries/options
    - Chromium: lite off, >128MB code
    - Seeing <50MB of hot code in prod
    - Lite has two benefits: skips cold code, less corner cases
- Practical recommendations
    - Use lite mode
    - Ffunction-sections
    - Avoid section syms

Pavel

- Using chromium in no-lite as test vehicle to discover bugs
- Assembly functions are there for a reason: hot/critical functions

Rafael

- Skipping all assembly functions is safe from perf perspective (layout)
- Replaying decisions of layout in relaxation is fragile if logic drifts

Adam

- Still a better approximation

Alexandros

- Care about all knobs (hugify,hot-funcs-at-end,...)
- List all important options that aren’t supported by exp-relax

Sep 9, 2026

Notes:
Amir:

- Let's use the agenda doc (this doc) for upcoming meetings

Pavel:

- Subdomain and BOLT large tests repo: pending Peter Hosek's action

From Alexey:

- Full binary rewriting v2: works on clang, breaks database workloads - needs binary reproducers

Alexandros:

- Call/branch relaxation: Rafael tested, new updates from Sep 8, 2026
- Rafael: Maksim created experimental relaxation as fast path, concerns about performance if clusters are cut incorrectly

Rafael:

- Never shrink adrp+add back – make it strict again
- Adam's idea is to use LongJmp pass
- Reference Mold paper about thunk creation: mold: A Massively Parallel Linker

Pavel

- Patches for hugify: refactoring, hugify-all-text
- Amir to review
- Alexey: hugify-all-text current version may not be correct as-is.

Amir:

- hugify-all-text might not be the right place in BOLT, better place is linker or kernel
- Kernel support: 1) reduces runtime overhead, 2) works for mthp, 3) reduces startup latency esp for Android use case

Andreas:

- Compressed debug sections: to ping Alexander, Amir will take another look

Rafael:

- Problem about reviewer availability (PTO or moved on)

Alexey:

- Rodata loads simplification
- Maksim Rachinsky proposes changes to the order of relocations, BOLT sorts them - Rafael to review
- RISCV has several relocations with same offset - motivation

Rafael:

- Nominated Craig Topper as RISCV backend owner

Amir:

- Join RISCV sync up call to propose RISCV ownership