

LLVM平行宇宙计划：LLVM与openEuler协同进展与展望

赵川峰：openEuler TC委员、Compiler SIG maintainer

目录

- LLVM平行宇宙计划介绍

- LLVM的优势及劣势

 - LLVM的优势

 - LLVM的劣势

- LLVM与openEuler协同

 - 嵌入式场景

 - 云场景

 - 服务器场景

- 下一步工作展望

LLVM平行宇宙计划介绍

LLVM平行宇宙计划—缘起

Apple公司推动LLVM成熟

- 2000年LLVM起源于UIUC, 2003年10月首次发布LLVM v1.0
- 2005年Chris Lattner加入Apple, **参加了LLVM/GCC等项目**
- 2007年C/C++/Objective-C前端Clang 成为开源项目
- 2009年**Clang/LLVM编译器在Mac OS X 10.6(Snow Leopard)商用**
- 2010年开始设计Swift语言, 基于LLVM实现
- 2014年Apple正式采用Swift开发iOS和macOS应用
- 至今Apple**所有的操作系统** iOS/macOS/tvOS/watchOS 均由Clang/LLVM构建

LLVM平行宇宙计划—缘起

Google公司拥抱LLVM

- 2010年Google的RenderScript项目基于LLVM实现
- 2011年贡献了AddressSanitizer代码, 两年后LSAN、TSAN等
- 2014年**所有的Host Builds采用Clang/LLVM构建成功**
- 2016年Android Userland 缺省采用了Clang/LLVM构建(Google Pixel)
- 2017年由Clang/LLVM**构建的Linux内核首次商用**(Google Pixel 2)
- 2018年Linux内核的LTO和CFI在Google Pixel 3上商用, NDK放弃了GCC
- 2020年Android系统放弃了GCC工具链
- 至今ChromeOS和Google数据中心都使用了**由Clang/LLVM构建的内核**
 - Facebook/Meta的数据中心也这样做的

LLVM平行宇宙计划—缘起

开源OS推动LLVM默认构建

OpenMandriva:

- 2014年, OpenMandriva社区开始切换到LLVM构建
- 2019年, 宣称OpenMandriva是**第一个Clang/LLVM构建的Linux发行版**

Chimera Linux:

- 2021年, Chimera Linux启动, **完全基于LLVM技术栈构建, 应用freeBSD工具**
- 2023年, Alpha release
- 2024年, 升级到6.6 LTS kernel

LLVM平行宇宙计划—缘起

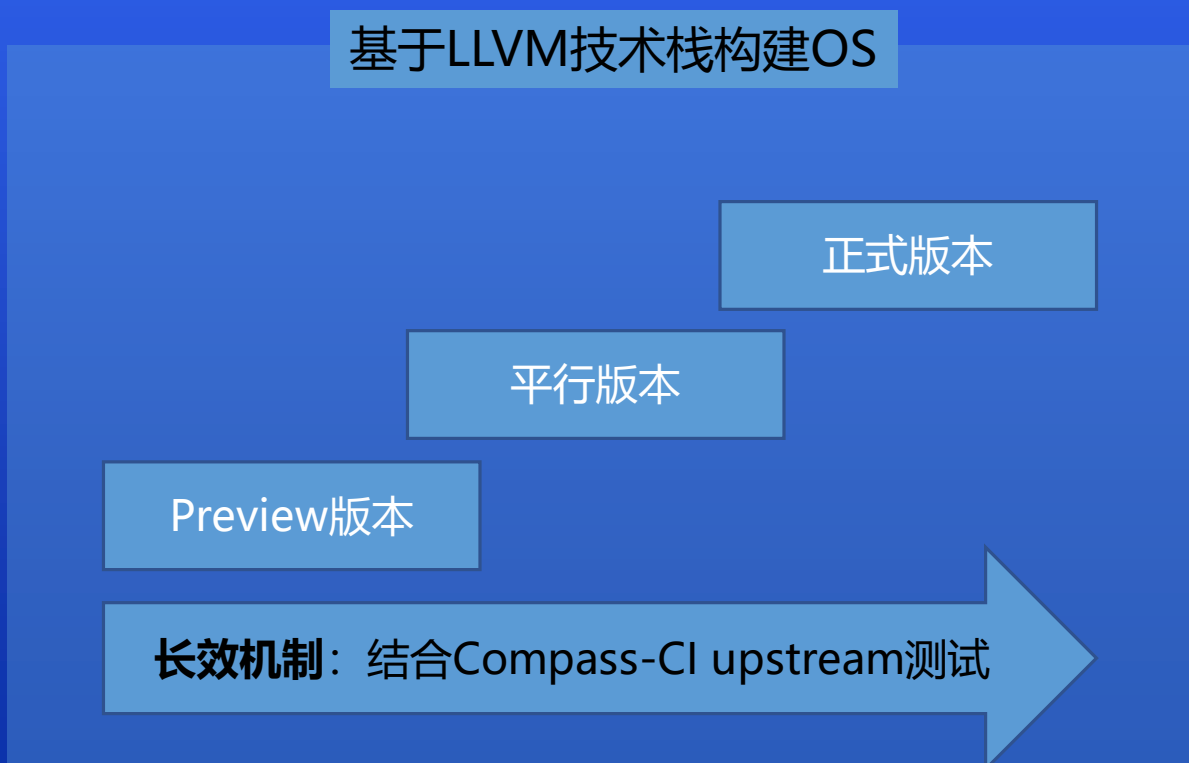
编译器开发生态向LLVM技术栈转移：大量基于LLVM实现的编译器

- **AMD** Optimizing C/C++ and Fortran Compilers (AOCC)
- **AMD-Xilinx** Nanotube Compiler for Xilinx FPGA
- **Apple** Swift Compiler
- **Arm** C/C++ Compiler for Linux, Arm Fortran Compiler
- **Huawei** Bisheng C/C++/Fortran Compilers
- **IBM** Open XL C/C++/Fortran for AIX
- **Intel** oneAPI DPC++/C++ Compiler, Fortran Compiler
- **NVIDIA**'s CUDA Compiler (NVCC)
- **Qualcomm** Snapdragon LLVM Compiler
- **Rust** Compiler (内核开发语言之一)
- **Kotlin** Native Compiler
- **Mojo** Compiler

LLVM平行宇宙计划—openEuler拥抱LLVM

版本与长效机制结合，社区化协作推进工作开展

2023年，软件所与华为编译器团队共同发起“LLVM平行宇宙计划”



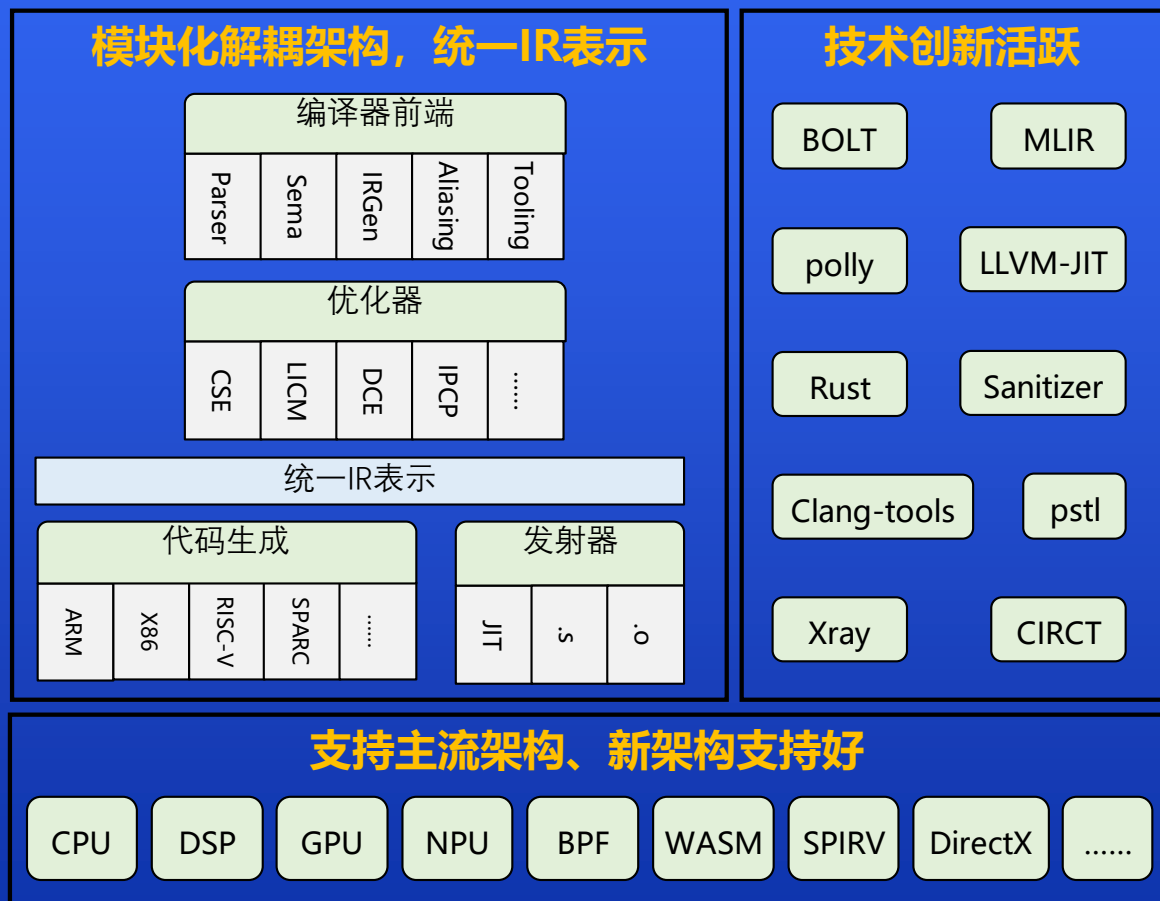
关键进展：

- 社区基础设施支持软件包选择GCC/LLVM构建
- RISC-V平台发布openEuler 24.03 LLVM preview版本
- 嵌入式场景发布openEuler 24.03 LTS LLVM 正式版本

LLVM的优势及劣势

LLVM的优势

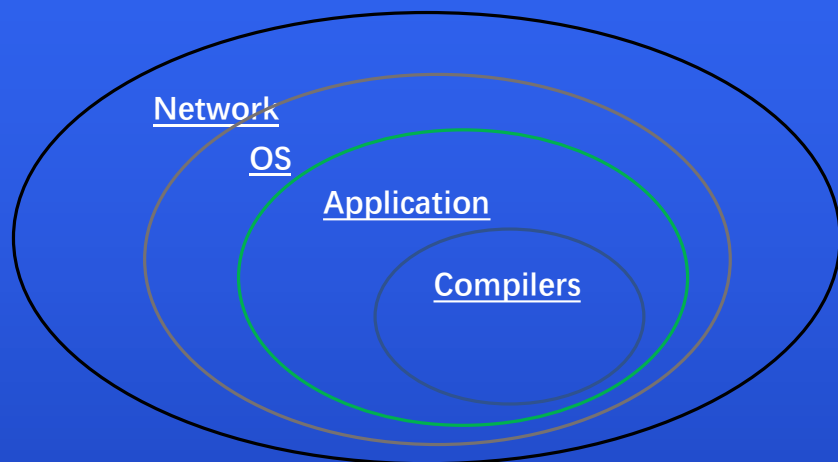
良好的架构及大公司的支持，更多的新架构新语言支持，更活跃的技术创新，形成一个更好的社区生态



- **模块化解耦架构，统一IR表示：**各功能模块以库方式组织，可分可合；统一IR保证各模块的重用性。
- **新架构支持好：**GPU、NPU、BPF、WASM等，为应用软件增加平台支持。
- **技术创新活跃：**基于LLVM基础设施开发多种工具/库，提升应用性能、安全及开发效率。

LLVM的优势

支持编译器安全方面更多特性



- ❑ 网络安全 (Firewalls, VPC, Security Groups etc.)
- ❑ 操作系统安全 (File System, Permissions etc.)
- ❑ 应用程序安全 (Auth0, API Token etc.)
- ❑ 编译器安全 (Stack Protection, PIE, CFI etc.)

调查：60% ~ 70% 操作系统和浏览器的安全问题来自于内存安全

解决内存安全两大途径：

1、安全加固：C/C++语言

- LLVM与GCC都支持：
 - ASAN, UBSAN, TSAN, HWASAN, KASAN, KCSAN
- LLVM特有支持：
 - MSAN, KMSAN、CFI、KCFI、SafeStack、GWP-Asan

2、安全语言：Rust语言

- LLVM是Rust语言的官方后端编译器
 - Rust特性支持完善
 - 支持Rust与C/C++混编LTO
 - 获取更好性能
 - 支持依赖LTO的特性，如CFI
- 例：Firefox、dora-rs

LLVM的优势

技术生态与开发者生态繁荣

arXiv

Advanced Search

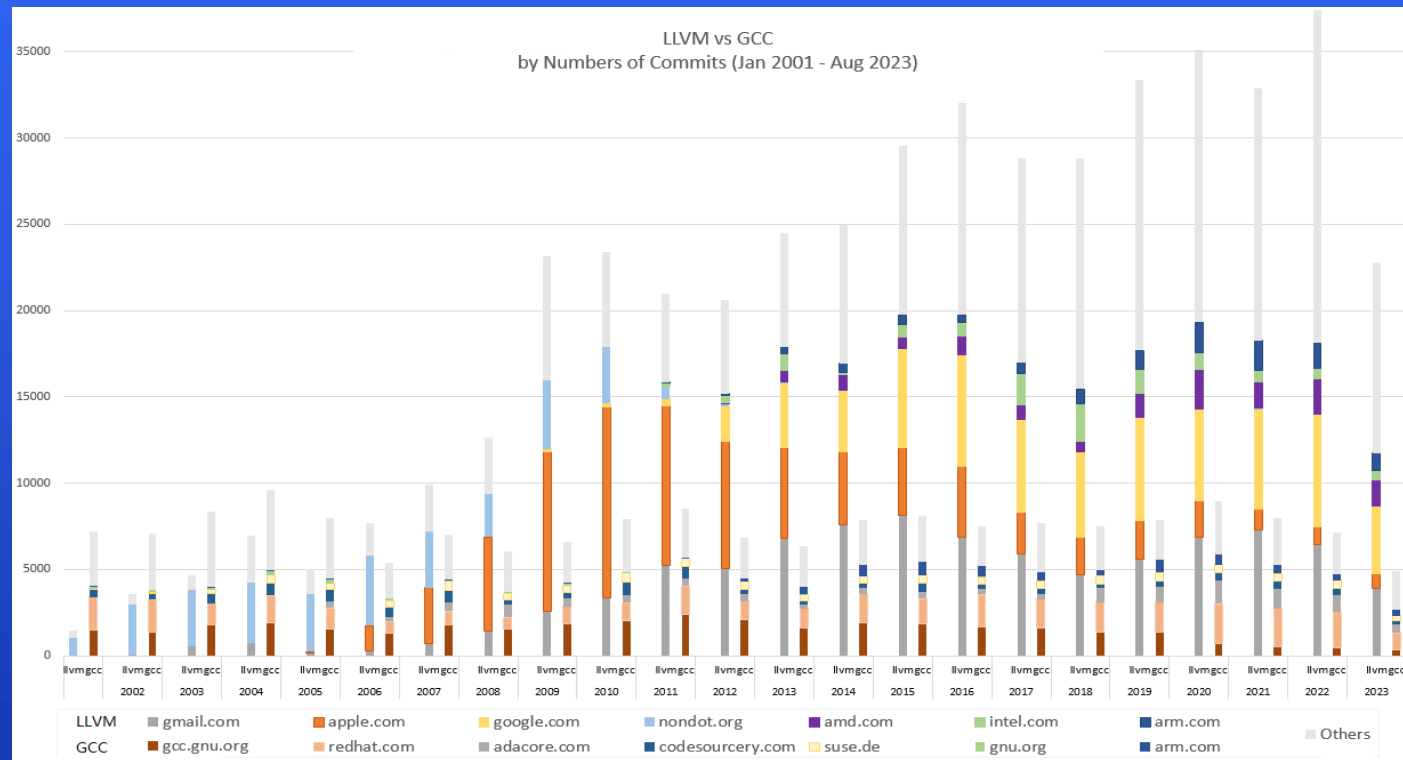
Search term(s)
"LLVM" Abstract

Add another term+ Search

Subject
All classifications will be included by default.
☒ Computer Science (cs) ☐ Physics all
☐ Economics (econ) ☐ Quantitative Biology (q-bio)
☐ Electrical Engineering and Systems Science (eess) ☐ Quantitative Finance (q-fin)
☐ Mathematics (math) ☐ Statistics (stat)

☒ Include cross-listed papers ☐ Exclude cross-listed papers

Date
☐ All dates
☒ Past 12 months
☐ Specific year
☐ Date range



最近一年Paper数量:

LLVM相关**64篇**; GCC相关**8篇**

社区commits数量及TOP贡献组织:

2009年之后LLVM社区增长明显, **互联网&芯片公司贡献较多**

LLVM的劣势

- LLVM对较过时的某些语言、某些平台支持的少
 - 如Pascal、Ada、D语言等、H8/300、PDP-11平台等
- LLVM对语言标准遵从度高，但对一些legacy特性、不标准特性与GCC不一致
 - LLVM对一些常用的特性、GNU扩展不断支持；
 - 软件适配，开发者持续贡献；
- LLVM社区尚没有LTS版本
 - Compiler SIG在openEuler社区维护LTS版本

LLVM与openEuler协同进展

嵌入式场景

嵌入式场景相对封闭的特性，特别适合作为新技术发挥价值的战场

嵌入式场景的诉求与挑战：

- **代码体积**：成本、应用特性、OTA
- **程序安全**：安全加固相对少，程序本身安全重要
 - 内核、浏览器、关键应用
- **交叉编译**：编译器交叉构建相对复杂

关键价值：

- 全系统thinLTO+Oz，**降低镜像体积10%+**；
- 提供Clang+LLVM全栈SDK，**对C++/Rust混编应用支持友好**；
- 提供强大的、轻量化的**Sanitizer**功能；

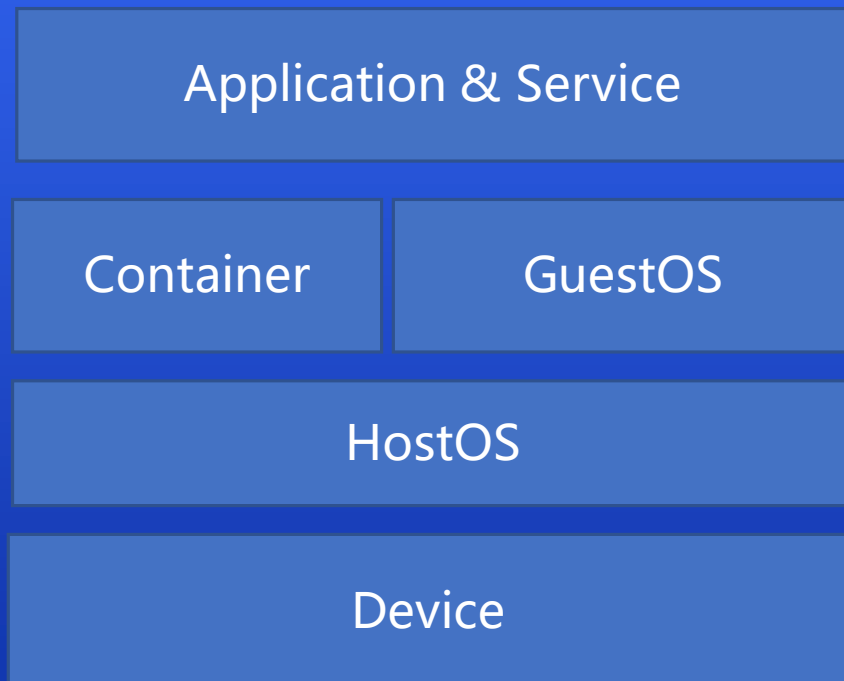
目前进展：

- 530发布正式版本，**下载地址**
 - 支持Aarch64平台，标准镜像（200+包）

云场景

云场景操作系统和应用的运行环境相对封闭，是应用LLVM全栈全链路优化的首选场景

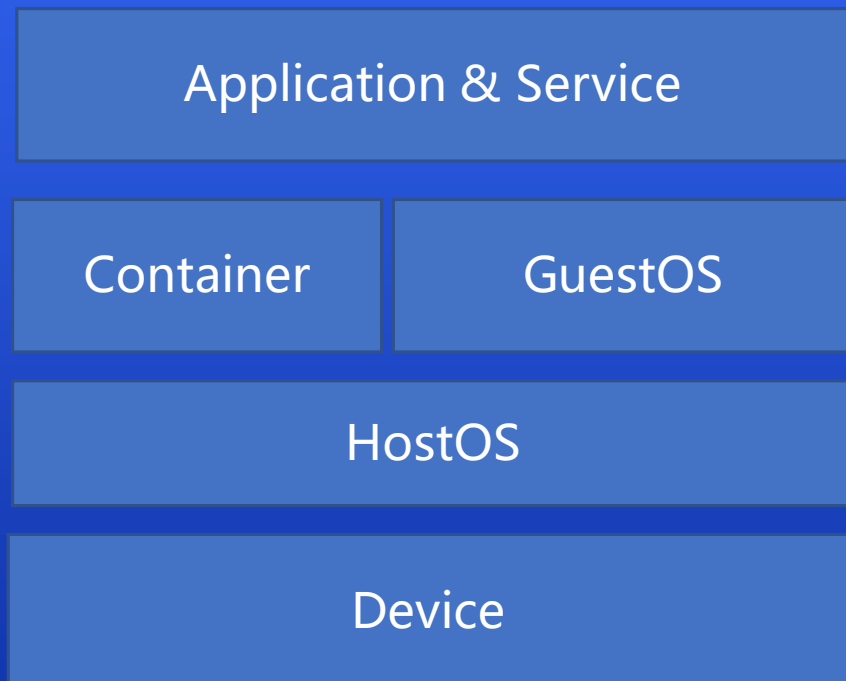
如何通过编译优化进一步压榨性能？



云场景

云场景操作系统和应用的运行环境相对封闭，是应用LLVM全栈全链路优化的首选场景

如何通过编译优化进一步压榨性能？



全栈

应用: mysql、nginx等
库: openssl、jemalloc等
内核: kernel

全链路

编译时

PGO/CSPGO

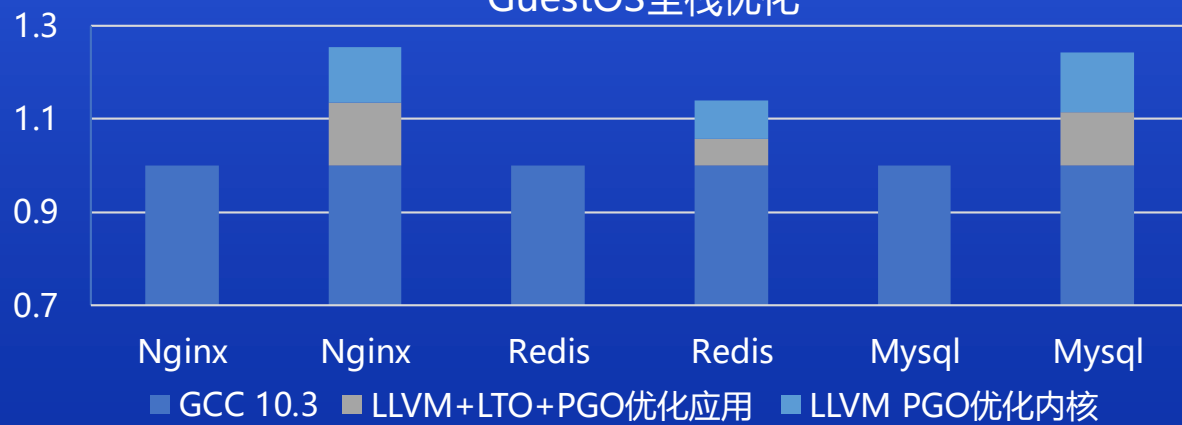
链接时

LTO/ThinLTO

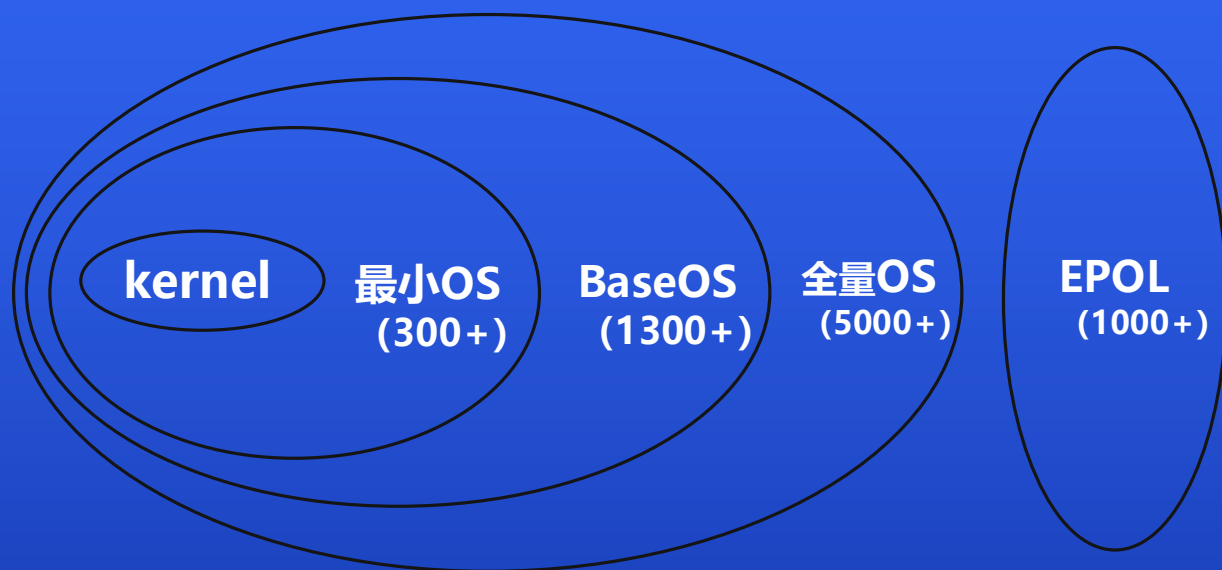
链接后

BOLT

GuestOS全栈优化



服务器场景



循序渐进，通过版本验证

目前进展：

验证：

- Kernel验证：1W+测试用例
- 外围包验证：1000+外围包验证
- 性能：RISCV版本，性能略微提升

目前没有发布LLVM本身的有效问题

版本发布：

- 530发布RISCV平台review版本
- KarsierOS：第一款基于 LLVM 平行宇宙成果的商业发行版（from苦芽科技）

下一步工作展望

下一步工作展望

加强LLVM编译器LTS构建；持续迭代LLVM平行宇宙openEuler版本

加强社区LLVM编译器能力

- 优体验：
 - 主+多版本支持，24.09支持LLVM 18
 - 持续Bugfix、质量加固（LTS）
- 高性能：
 - 持续优化kernel、反馈优化增强
- 易创新：
 - Autotuner支持，编译器参数细粒度调优
 - AI4Compiler构建，AI Enhanced 编译器优化。

持续迭代LLVM平行宇宙openEuler版本

- 嵌入式：
 - X86平台嵌入式版本及LLVM全栈SDK
 - ThinLTO+Oz，进一步降低代码体积
- RISCV平台：
 - 扩大软件包支持范围；正式版本流程
- 其他平台：
 - 逐步扩大到其他平台支持

THANKS