

openEuler LLVM 平行宇宙计划介绍与进展

2024-04-27

目 录

01

背景

02

LLVM 平行宇宙计划 介绍及进展

03

后续规划

背景：以 Clang 构建内核的努力

- 困难
 - Linux Kernel 代码存在围绕 GCC 特有机制 (GCC'isms) 的实现
 - 对移植到其他编译器产生了阻碍
 - LLVM 编译器套件特性支持未有 GCC 全面
 - 曾缺少 Kernel 构建所需扩展、机制支持

背景：以 Clang 构建内核的努力

- 历程
 - 2010 年之前：Clang 的 C/C++ 功能实现逐步发展完善
 - 2010 年：成功以 Clang 构建 2.6.36 内核并在裸机以及虚拟机上运行
 - 2012~2016 年：知名厂商有意用于 Arm 架构，建立推动 LLVMLinux 计划
 - 2017 年：相关基础补丁主线化接近尾声；成功实现 LLD 链接

背景：以 Clang 构建内核的努力

- 历程
 - 2018 年：**ClangBuiltLinux** 项目建立，Clang 构建 Kernel 落地厂商应用
 - 2019 年：Clang 9.0 可构建同期 x86_64/aarch64 架构 5.3 主线 Kernel
 - 2021 年：Clang LTO 构建合入 5.12 Kernel 主线；内建汇编器覆盖更多架构
 - 2021 年以降：Kernel 支持逐渐成熟完善，riscv64、loongarch64 等新架构可用

背景：以 Clang 构建发行版/衍生版的历程

- 历程
 - 2009 年中 FreeBSD 开发者开始评估使用 LLVM/Clang
 - 2012 年初 Debian 开发者成功以 Clang 构建出其时多数 Debian 软件包，提议解耦 GCC
 - 2012 年中 FreeBSD 宣布将在 10.0 版本改用 LLVM/Clang

背景：以 Clang 构建发行版/衍生版的历程

- 历程
 - 2016 年 Android 用户态部分可全部通过 Clang 编译
 - 2019 年中 OpenMandriva 切换默认编译器为 Clang
 - 2021 年 Chimera Linux 创立，以 Clang 作为默认编译器
 - 2022 年至今 更多主要发行版评估并推进同时支持 GCC/LLVM 构建

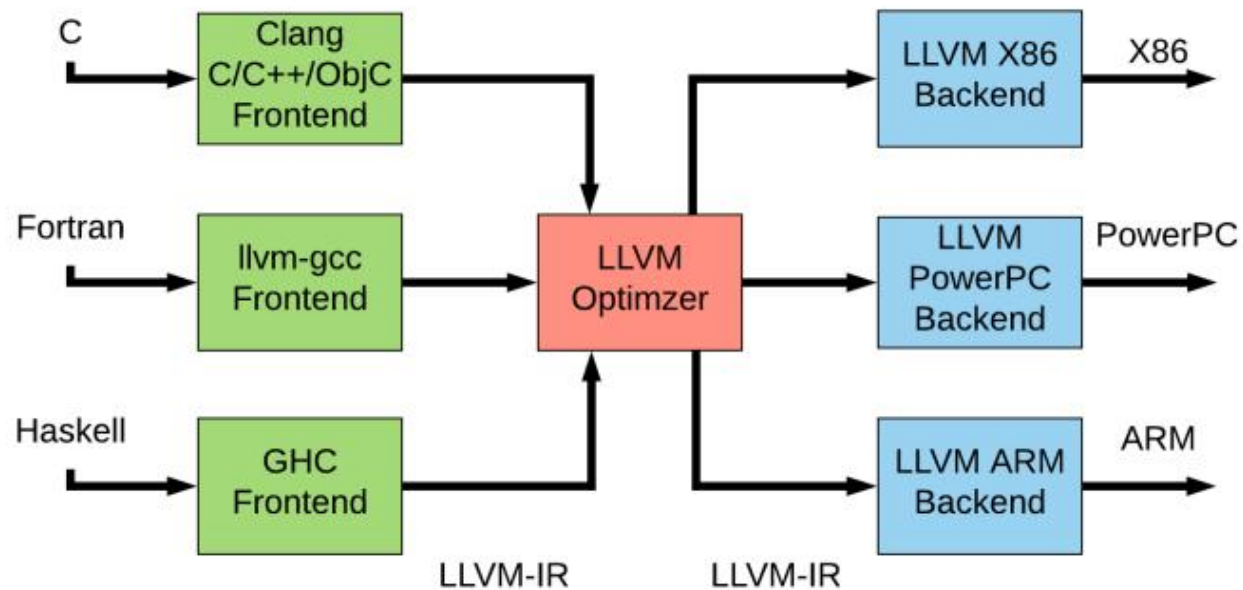
II LLVM 平行宇宙计划 介绍及进展

「平行宇宙计划」是什么

- openEuler 社区的一个探索性项目
 - 社区演进提案 oEEP003：[LLVM平行宇宙计划--基于LLVM技术栈构建oE软件包](#)
- 尝试使用 LLVM 编译器套件替换 GCC 套件构建 Linux 发行版与软件包
- 独立推进，与社区版本平行
 - 故得名平行宇宙

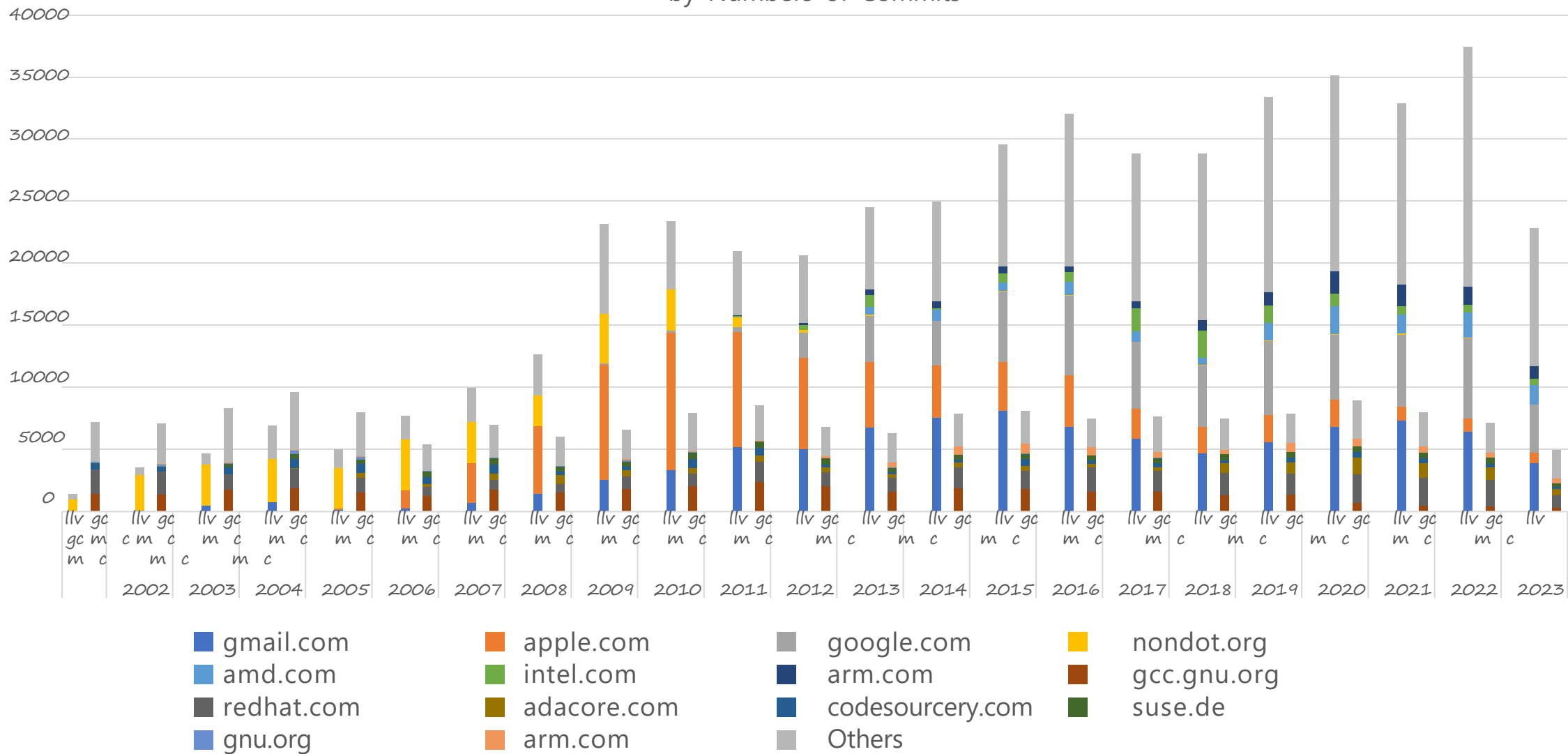
为什么换用 LLVM : LLVM 的特点

- **架构良好**：模块化解耦，统一的 IR 表示，强大的 Pass 系统
- **协议友好**：LLVM 9.0 后以 Apache 2.0 协议分发，许可协议较为宽松，对商业用户和贡献者更友好
- **繁荣的生态活跃度**：截至目前，LLVM 社区贡献者已经达到两千余人，2022 年增加三百余人，涉及公司 150+，周 Commit 数量超 500+ 以上



为什么换用 LLVM：社区活跃度及贡献者对比

LLVM vs GCC
by Numbers of Commits



为什么换用 LLVM ：当前生态适配较充分，可行性高

- 在 GCC 之外 Linux Kernel 支持使用 LLVM 编译器套件构建多个架构的产物（见右图）
- Android 与 ChromeOS 已使用 Clang 构建的 Kernel
- Google 和 Meta 已在 DC 运行 Clang 构建的 Kernel
- 其他 Linux 发行版也有评估验证

Architecture	Level of support	make command
arm	Supported	LLVM=1
arm64	Supported	LLVM=1
hexagon	Maintained	LLVM=1
loongarch	Maintained	LLVM=1
mips	Maintained	LLVM=1
powerpc	Maintained	LLVM=1
riscv	Supported	LLVM=1
s390	Maintained	CC=clang
um (User Mode)	Maintained	LLVM=1
x86	Supported	LLVM=1

为什么换用 LLVM : 潜在性能竞争力

LLVM Clang 16 vs. GCC 13 Compiler Performance On AMD 4th Gen EPYC "Genoa" (2023-6)

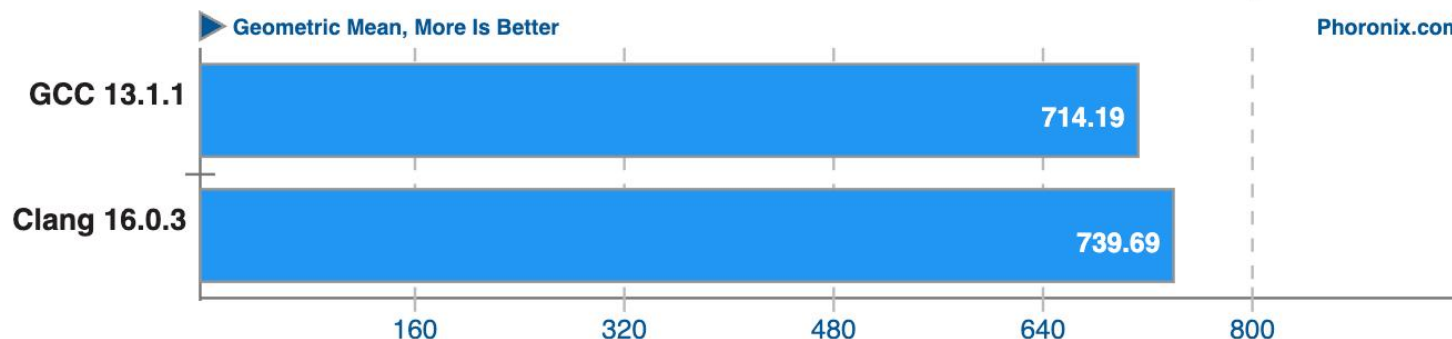
<https://www.phoronix.com/review/amd-znver4-gcc13-clang16/6>

测试的 131 个用例中

- Clang 16 在其中 75 个用例的性能领先 GCC 13.1
- 原始测试数据的几何平均值领先 4%

Geometric Mean Of All Test Results

Result Composite - AMD EPYC Genoa Compilers



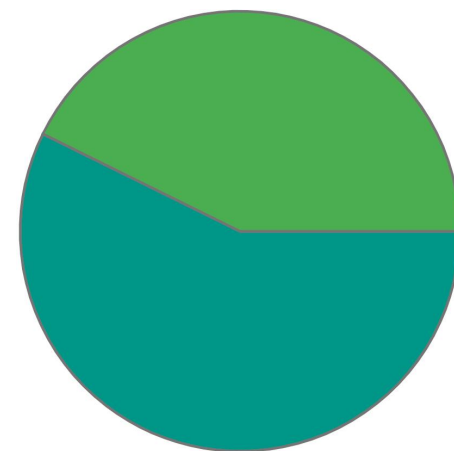
Number Of First Place Finishes

Wins - 131 Tests

Clang 16.0.3 75 [57.3%] GCC 13.1.1 56 [42.7%]

pts

Phoronix.com



为什么换用 LLVM : 潜在性能竞争力

LLVM Clang 12 vs GCC 11 Squeezing More Performance Out Of The Linux Kernel With Clang + LTO (2021-7)

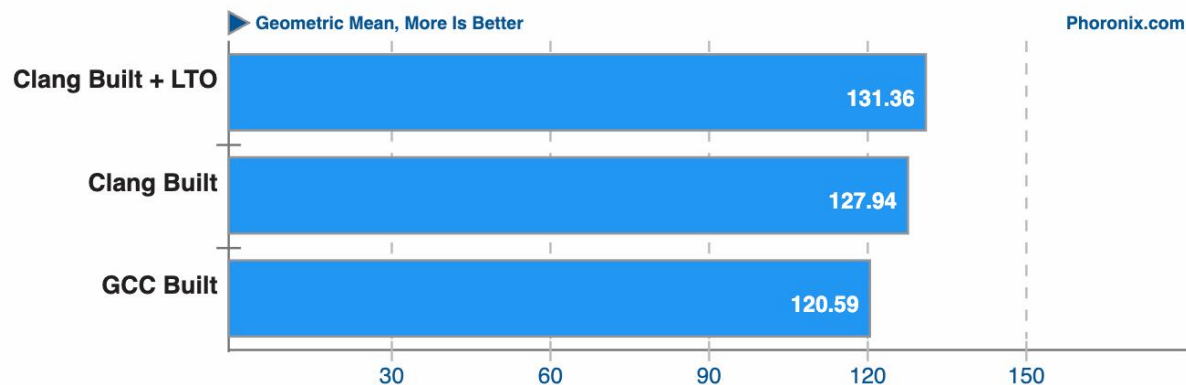
<https://www.phoronix.com/review/clang-lto-kernel/4>

测试的 23 个用例中

- Clang 12 开 LTO 构建的 kernel 相比 GCC11 未开 LTO, 其中 20 个用例性能领先 (上图/饼图)
- 原始测试数据的几何平均值领先 9% (左图)

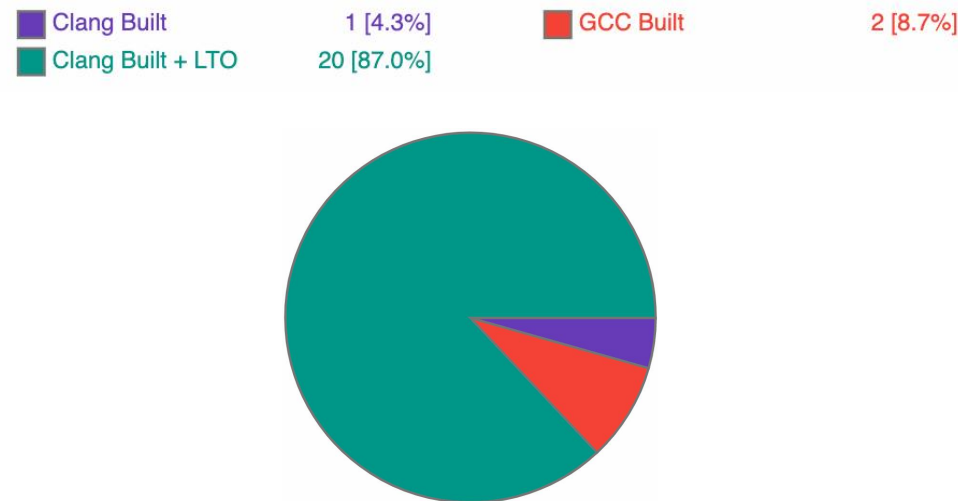
Geometric Mean Of All Test Results

Result Composite - Clang LTO Linux Kernel Build Intel Benchmarks



Number Of First Place Finishes

Wins - 23 Tests



为什么换用 LLVM ：其他收益

- 更好的代码质量检测
 - 静态检查 clang-tidy
 - 原生的运行时检查 sanitizer
 - 地址消毒、未定义行为检查、线程检查、控制流检查、堆栈溢出保护
- 原生支持 rust 内核驱动开发

为什么换用 LLVM ：其他收益

- 模块化架构较为友好，针对性修改和优化更加容易实现
- 工业界和学界均较为熟悉
 - 学生群体在 LLVM 社区的参与度普遍较高，且相关论文/顶尖学术分享数量大幅领先
 - MLIR、CIRCT、BOLT 等新编译能力/特性不断涌现
 - 头部企业自研编译器转型 LLVM，专业人才可获得性高

「平行宇宙计划」各方

- 知名&初创企业、科研院所、在校学生、社区爱好者共建
- 双周例会讨论、IM 群组沟通、同步文档跟踪
- 欢迎所有有志者参与一同！

「平行宇宙计划」运作方式

- 代码同源
 - 软件包源码来自 src-openEuler 社区源码仓
 - 修改向 src-openEuler 社区源码仓/上游社区提交
 - 极少部分软件包异步提交，暂存于 openeuler-risc-v 中间仓（加速构建）

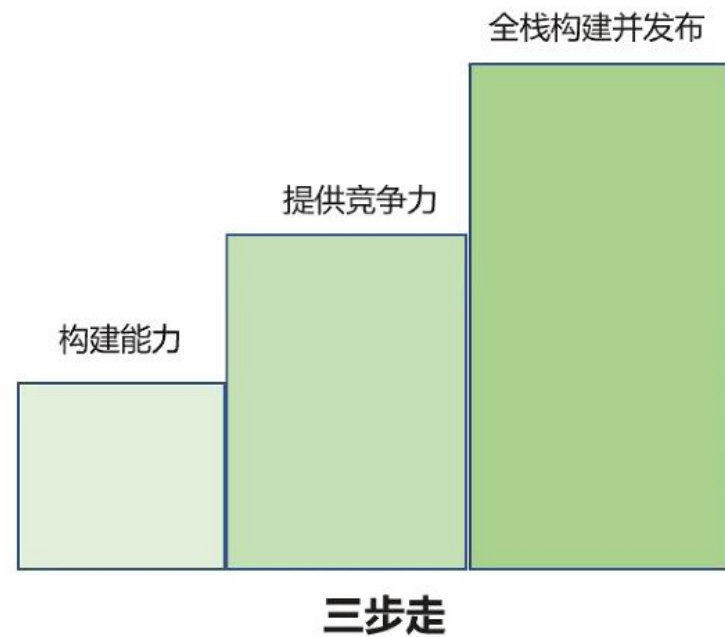
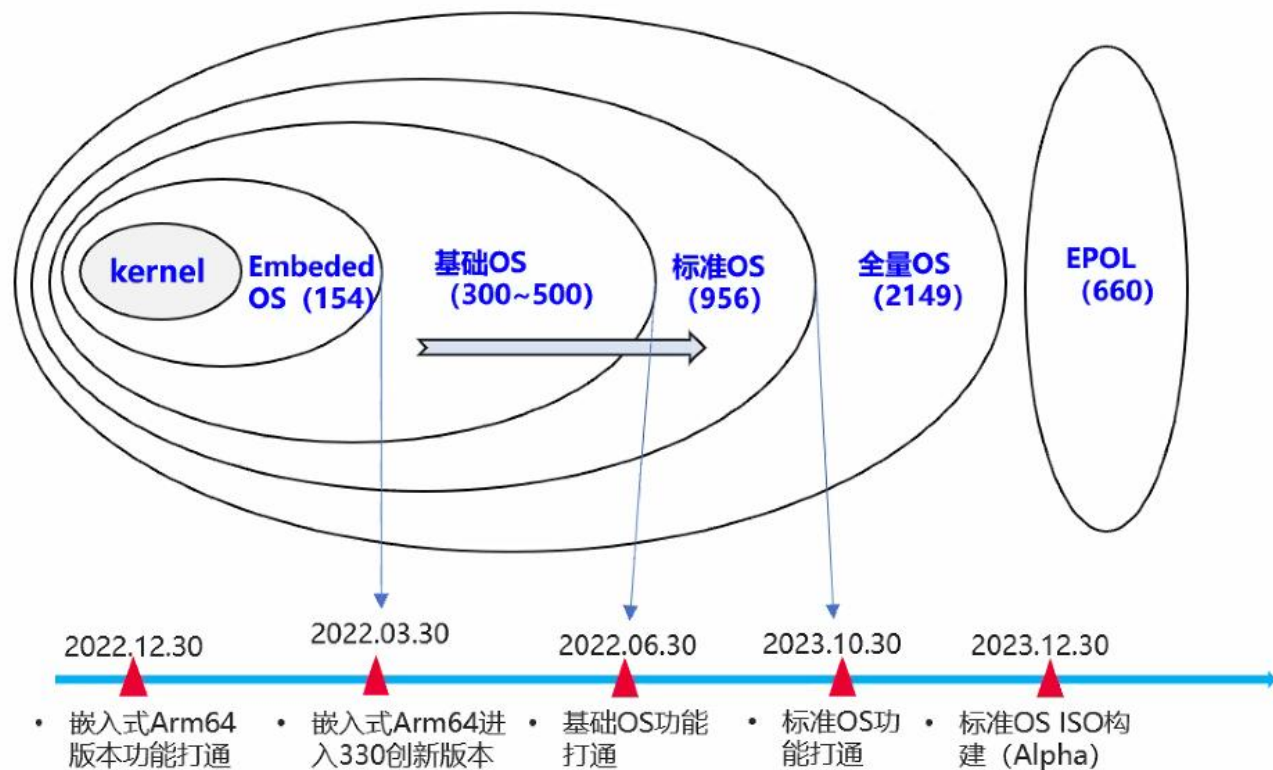
「平行宇宙计划」运作方式

- 切换 LLVM/Clang 编译器方式
 - 涉及软件包范围预先大致确定，**逐步演进扩大**
 - 构建工程仅对涉及软件包统一配置注入变量，**变更少、侵入程度较小**
 - 回合的修改均仅在 LLVM 构建时生效，**不影响社区 GCC 版本构建**

「平行宇宙计划」运作方式

- 切换涉及软件包范围
 - 当前：社区 **BaseOS** 范围 1400+ 软件包
 - 数个与 gcc 生态耦合过深的核心软件包暂未列入 (<20)
 - 持续：社区 Mainline (4800+) 及 Epol (1300+) 范围全量软件包

「平行宇宙计划」运作方式

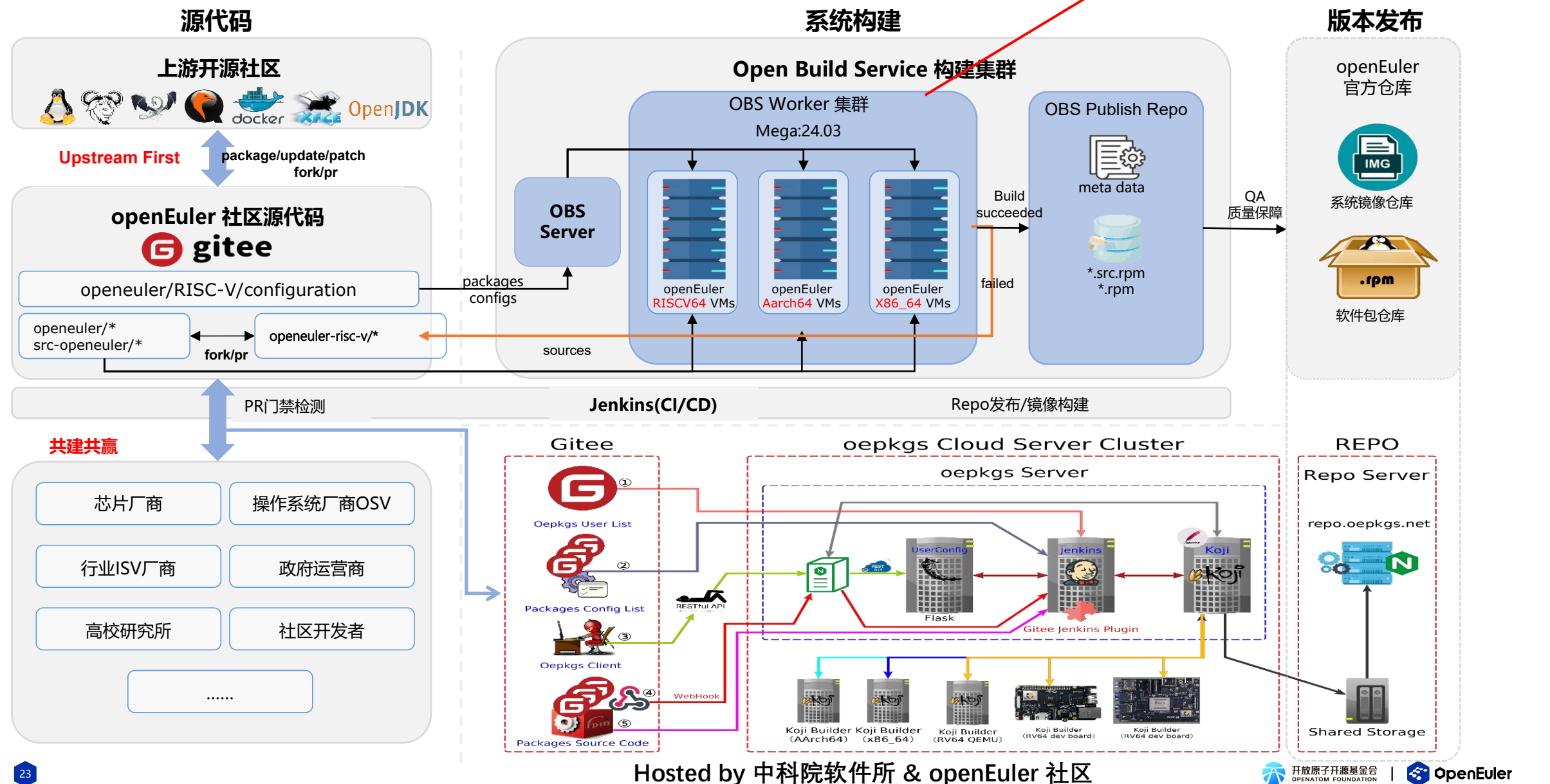


「平行宇宙计划」版本规划

- 社区 Preview 版本，后续逐渐演进为正式版本
- 当前对标社区 24.03 版本，面向 riscv64 架构
- 预期发布范围：**社区 BaseOS 范围 – 部分困难软件包 + 部分测试无误的 Mainline 软件包 (Best Effort)**
- **预期发布时间：2024-05-30**

「平行宇宙计划」当前基础设施

x86_64/aarch64/riscv64 三架构支持



「平行宇宙计划」当前推进情况

- 讨论确定发版计划和时间表：完成 ✓
- 发版计划报送 QA、Release 以及 TC 会议审批：完成 ✓
- 确定 BaseOS 软件包范围以及 LLVM/GCC 各自编译部分范围：完成 ✓
- BaseOS 软件包构建问题分工摸排、问题记录归类：完成 ✓

「平行宇宙计划」当前推进情况

- 编译器侧统一处理、忽略高频出现的报错 flag：大致完成 ✓
- 软件包构建问题修复：进行中➡
- 构建工程配置调整、软件包源码更新及 rebuild：逐步开展 ➡
- EulerMaker 平台及镜像生成工具适配：进行中➡
- 测试工作准备：进行中➡

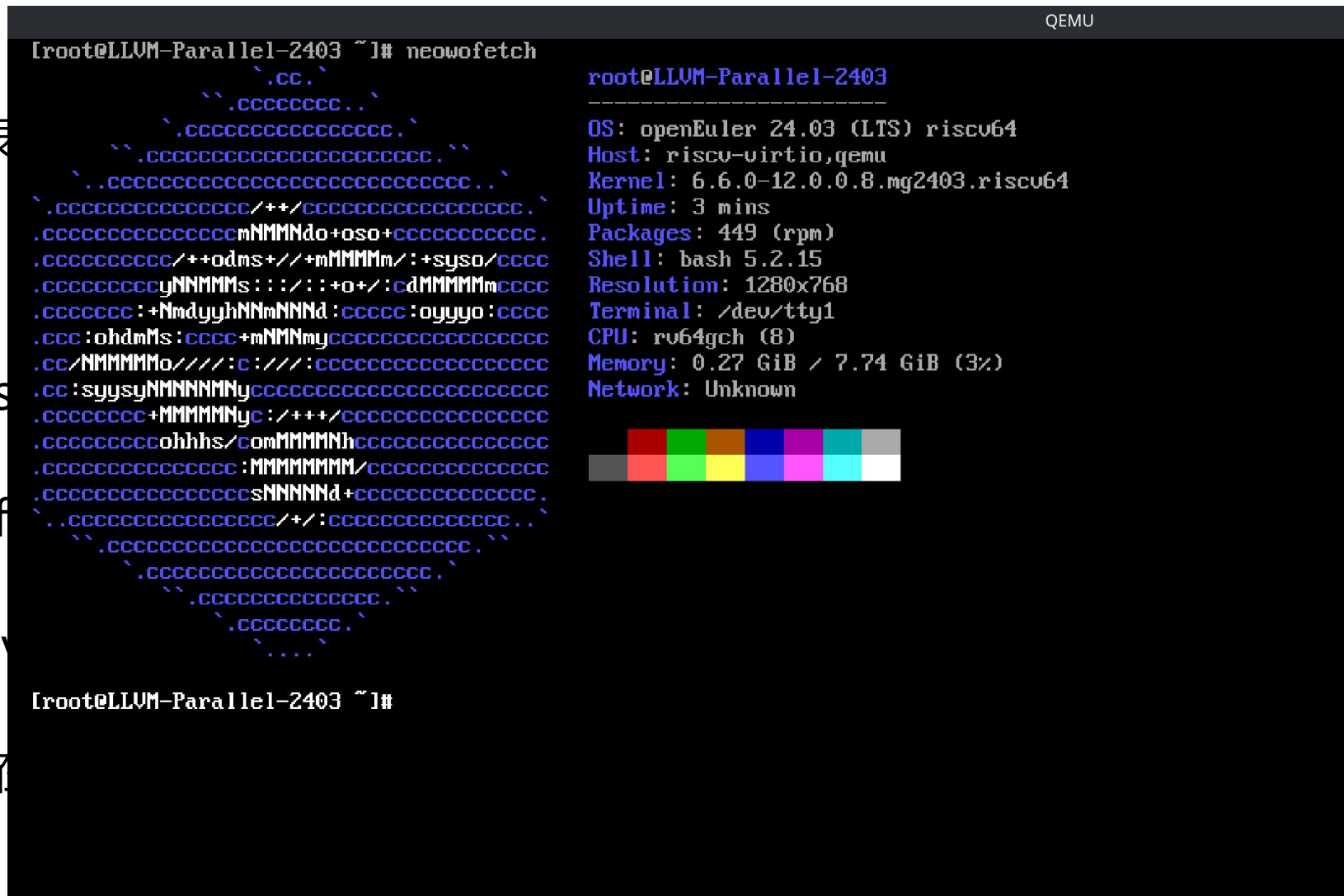
「平行宇宙计划」构建问题修复情况

- BaseOS 状态跟踪表 1411 个软件包中
- 上一构建轮次
 - riscv64 标记为 succeeded 的有 1368 个 (96.95%)
 - riscv64 标记为 failed 的有 27 个 (1.91%)
 - 余下大多为 riscv64 未适配、环境问题、新的告警错误等
- 当前构建轮次：正在 rebuild 中，目前大部分软件包已经构建通过

standard_aarch64 ▼	
 aarch64	 succeeded: 4767  failed: 108  unresolvable: 22  excluded: 27
standard_riscv64 ▼	
 riscv64	 succeeded: 4742  failed: 118  unresolvable: 12  building: 3  excluded: 49
standard_x86_64 ▼	
 x86_64	 succeeded: 4754  failed: 128  unresolvable: 27  excluded: 15

「平行宇宙计划」构建问题修复情况

- BaseOS 状态跟踪表
- 上一构建轮次
 - riscv64 标记为 s
 - riscv64 标记为 f
 - 余下大多为 riscv
- 当前构建轮次：正在



「平行宇宙计划」当期后续计划

- 四月底：完成 riscv64 外围包修复、合入
- 五月中旬：完成测试报告编纂，提交 QA 例会审阅
- 五月下旬：向 TC 例会汇报
- 五月底：提交发布评审
- **预计发版时间：2024-05-30**

「新增实验性 C 标准库 LLVM Libc 至社区」#[246970226](#)

- LLVM Libc：来自 LLVM 社区的新生 Libc，逐步完善中
 - 几乎纯 C/C++ 实现，上游 riscv64 支持较好
 - Sanitizer 友好、CI 测试覆盖充分、模块化&配置性高
 - 具体实现情况可查阅项目主页 -> libc.llvm.org/

「新增实验性 C 标准库 LLVM Libc 至社区」#[246970226](#)

- 任务内容：
 - 研究 LLVM Libc 的构建、测试和使用方式（Overlay vs Full Build ?）
 - 设法将 LLVM Libc 的构建和测试过程封装为 rpm 软件包
 - 推动在 openEuler 创建 llvm-libc 源码仓，并提交更改到 master 分支

感谢聆听

感谢 openEuler LLVM 平行宇宙计划的推动者们
对本次演示内容和数据上的大力支持