



LLVM for openEuler的版本规划

Compiler SIG Maintainer

谢志恒

目录

1. 什么是LLVM for openEuler?
2. LLVM for openEuler版本发布节奏
3. LLVM for openEuler版本配套关系
4. LLVM for openEuler特性预览
5. LLVM for openEuler性能优化效果

什么是LLVM for openEuler?



LLVM for openEuler是基于上游LLVM社区，**源码开源在openEuler社区，匹配跟随openEuler发布维护节奏的LTS编译器工具链版本**，具有优体验、高性能、易创新的特点。



优体验

- 源码仓搭载Compiler CI工程，百万级用例日常监控版本质量
- 支持LLVM多版本部署，随心切换不同版本进行集成开发
- 搭配智能迁移助手，轻松在不同编译器之间进行应用迁移



高性能

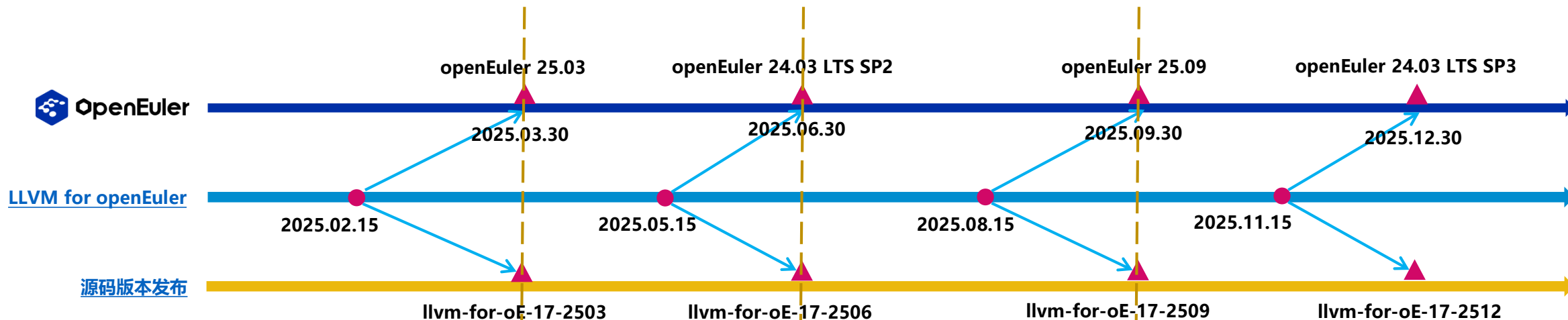
- 扩展支持鲲鹏后端硬件特性及流水线，释放极致性能
- 支持软件SoftBR采样功能，补充部分机型缺乏相应硬件能力
- 构建智能预取、BOLT内核优化、Propeller支持等能力，常用数据库应用性能大幅提升



易创新

- 架构解耦易扩展，持续接入RISC-V、龙芯、申威等后端处理器
- 引入AI for Compiler技术，支持AI模型一键式自动应用调优
- 探索LLM结合编译器优化，深度挖掘应用性能潜力

LLVM for openEuler版本发布节奏



优体验:

- Compiler CI工程支持
- LLVM主副版本支持
- LLVM副版本制品仓归一

高性能:

- 鲲鹏系列后端流水线支持
- 硬件扩展指令集支持
- LLVM内核PGO支持

易创新:

- AI4C-Autotuner的支持

[使用文档链接](#)

优体验:

- LLVM主版本制品仓归一
- 编译器切换兼容性问题处理

高性能:

- SoftBR能力支持
- 智能预取特性
- BOLT内核优化

易创新:

- 龙芯、申威后端持续接入
- AI编译-XLA图编译强化

优体验:

- LLVM副版本持续扩展

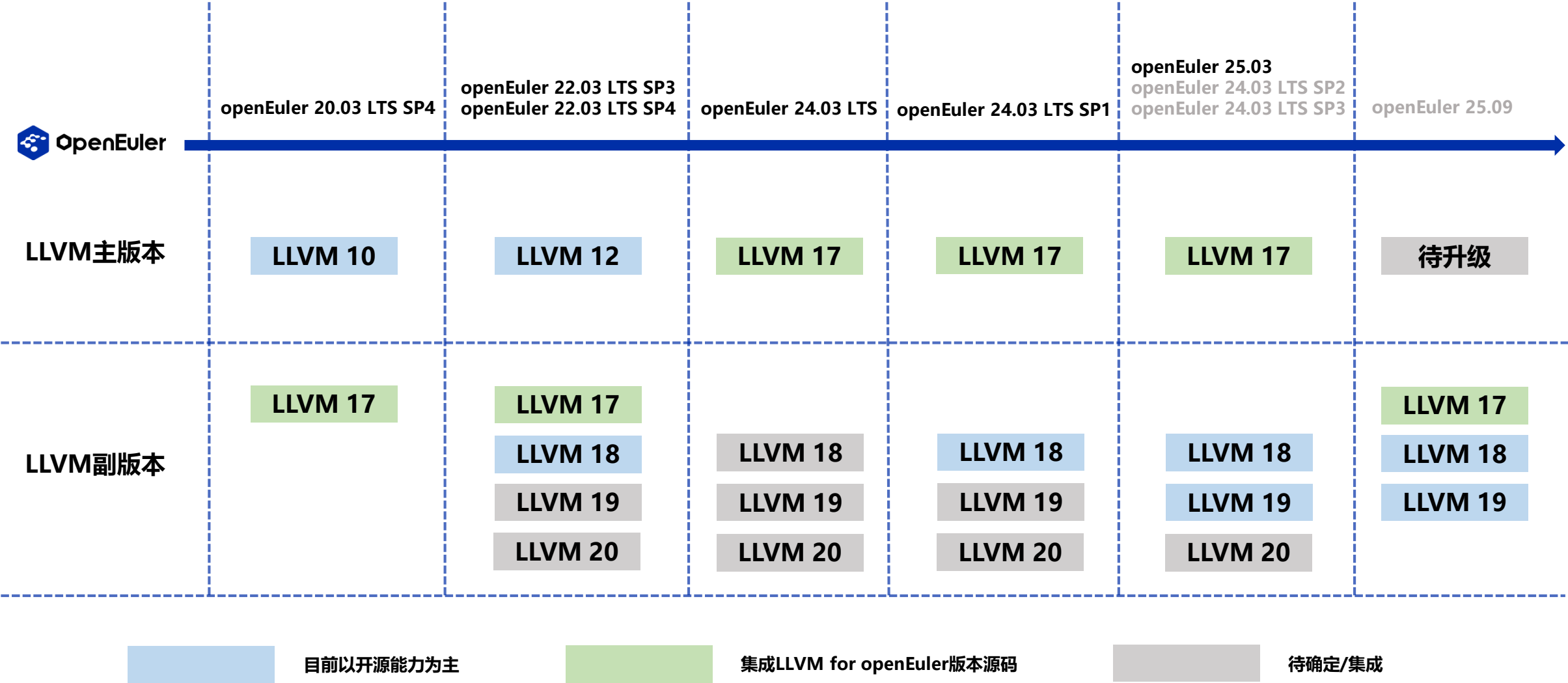
高性能:

- Propeller能力支持

易创新:

- AI4C模型推理-分支预测强化、优化选项推理等

LLVM for openEuler版本配套关系



LLVM for openEuler特性预览 — 智能预取



场景与诉求

- 一些应用（如数据库类）普遍存在由于dcache miss带来的CPU后端瓶颈问题，影响发挥CPU的极致性能
- 数据预取的有效性和收益大小受**预取时机**、**预取代价**等影响

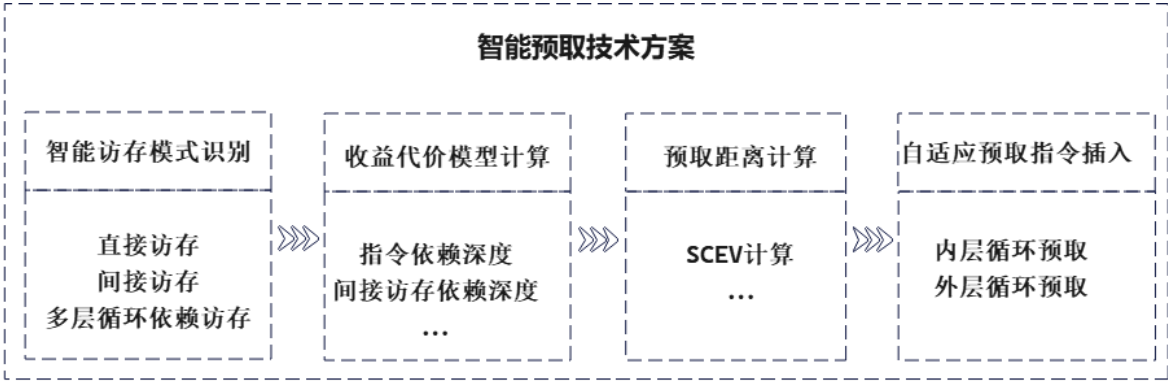
诉求

更广泛的有效预取场景识别和更精确的预取时机计算
自适应选择预取指令的cache level和循环层级

编译器预取能力

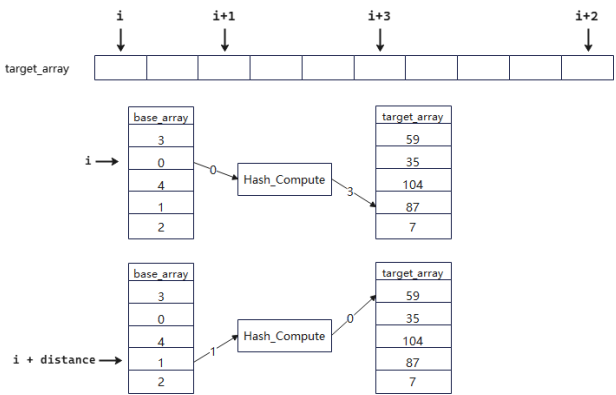
访存类型	描述	上游LLVM支持	LLVM for openEuler持续构筑
a[j]	内层循环下的直接访存	✓	✓
a[i+j]	多层循环下的直接访存	✗	✓
a[f(b[j])]	内层循环下的间接访存	✗	✓
a[f(b[i])+j]	多层循环下的间接访存	✗	✓

技术方案



方案效果

- Clickhouse：多层循环下间接访存

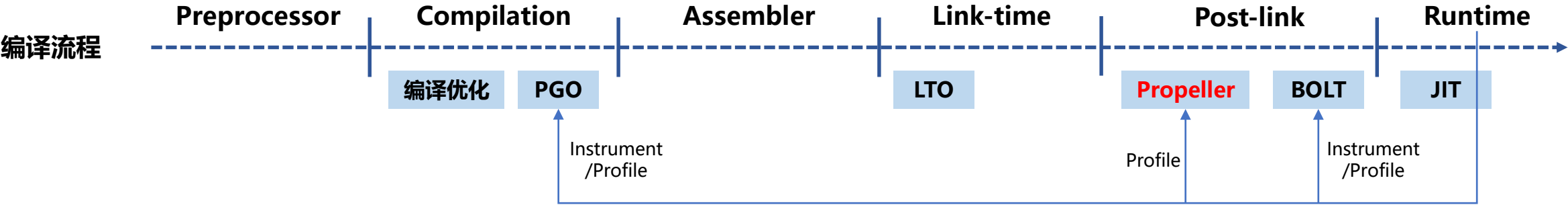


- 性能提升：10%+

LLVM for openEuler特性预览 — Propeller



参考文献



Propeller: A Profile Guided, Relinking Optimizer for Warehouse-Scale Applications

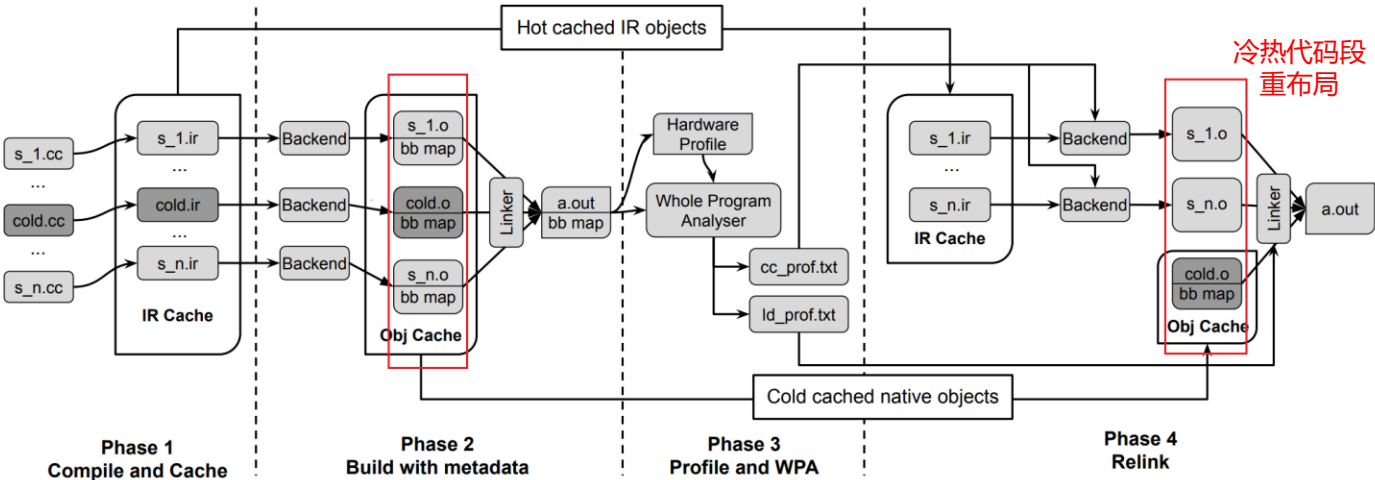


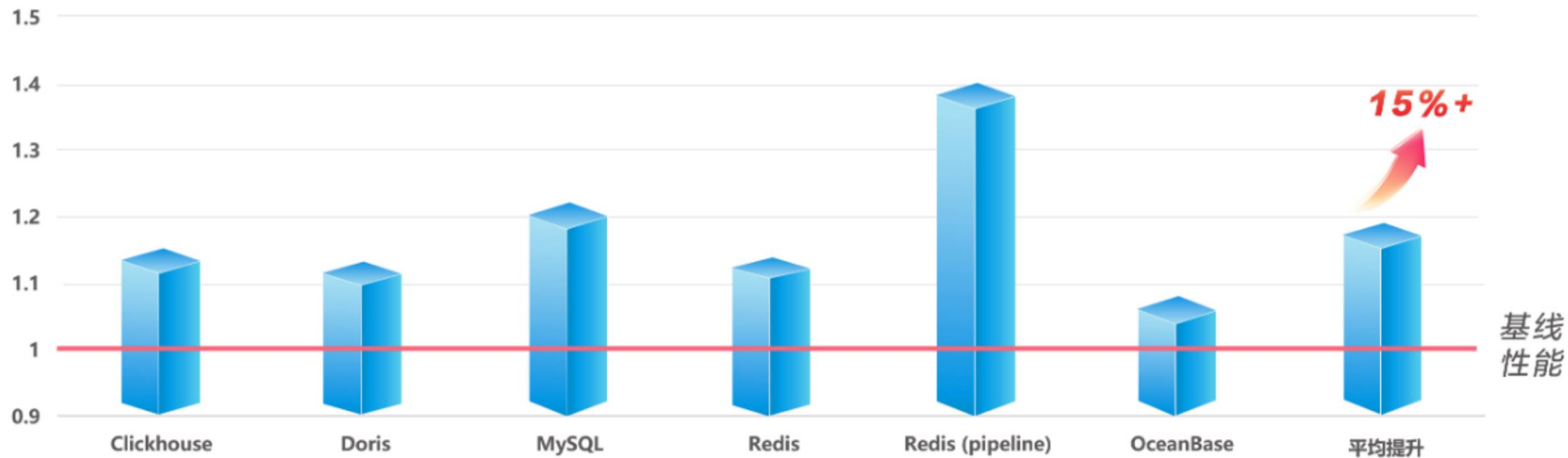
Figure 1: Design of a Profile Guided, Relinking Optimizer

Propeller VS. BOLT

	Propeller	BOLT
可调试性	较好	直接二进制修改, 相对较差
支持架构	理论上都支持	X86较成熟, AArch64可用
编译开销	减少30%以上开销 (内存消耗、编译时间)	—
性能收益	基本持平BOLT效果	—

LLVM for openEuler性能优化效果

数据库应用性能优化效果



互联网场景中常用数据库应用平均性能**提升15%以上**，其中涉及的优化能力会分阶段逐步合入LLVM for openEuler版本当中。



Thank You.

Compiler SIG 专注于编译器领域技术交流探讨和分享，包括 GCC/LLVM/OpenJDK 以及其他的程序优化技术，聚集编译技术领域的学者、专家、学术等同行，共同推进编译相关技术的发展。



毕昇编译公众号



Compiler 交流群小助手