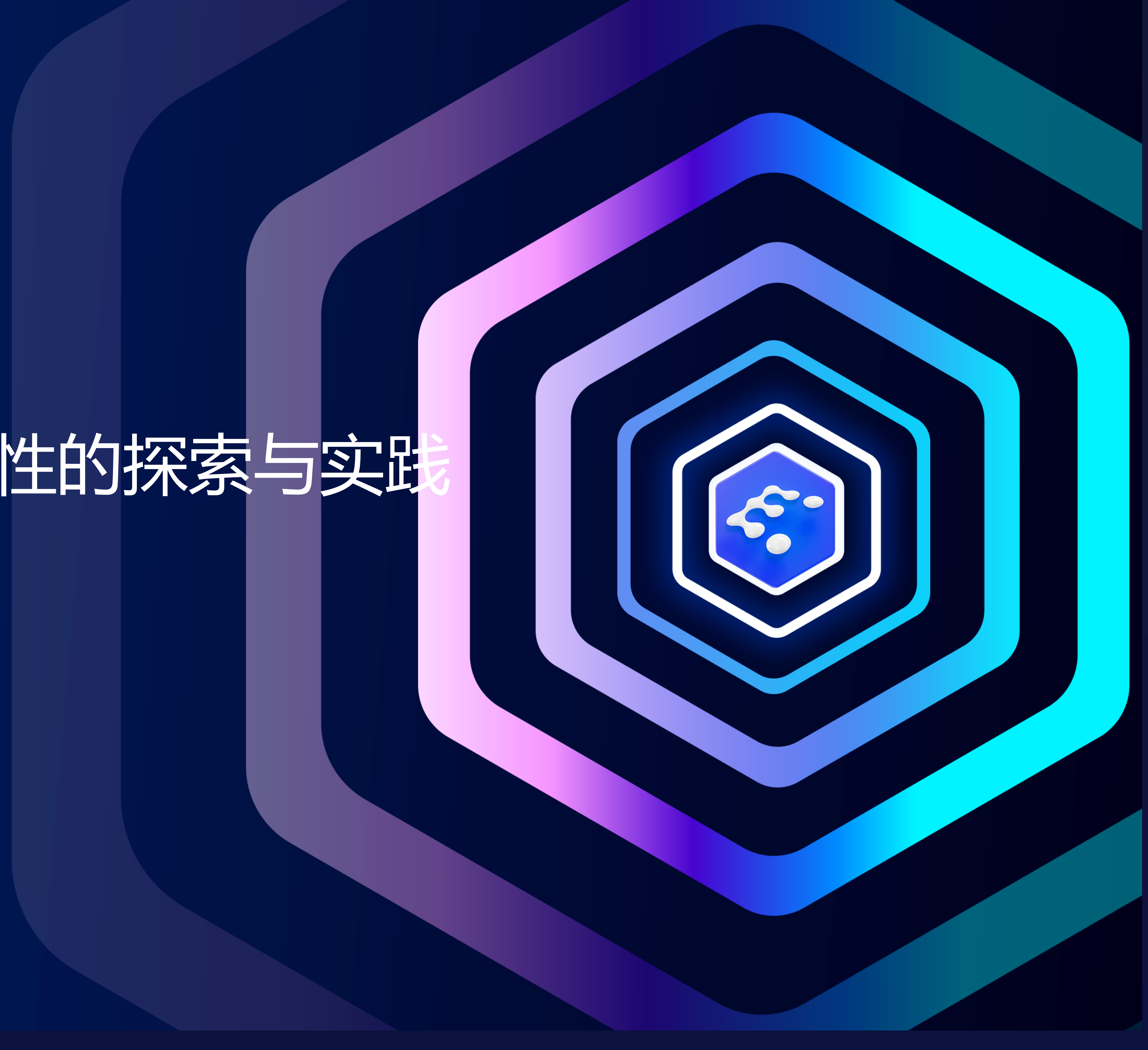


编译器反馈优化泛化性的探索与实践

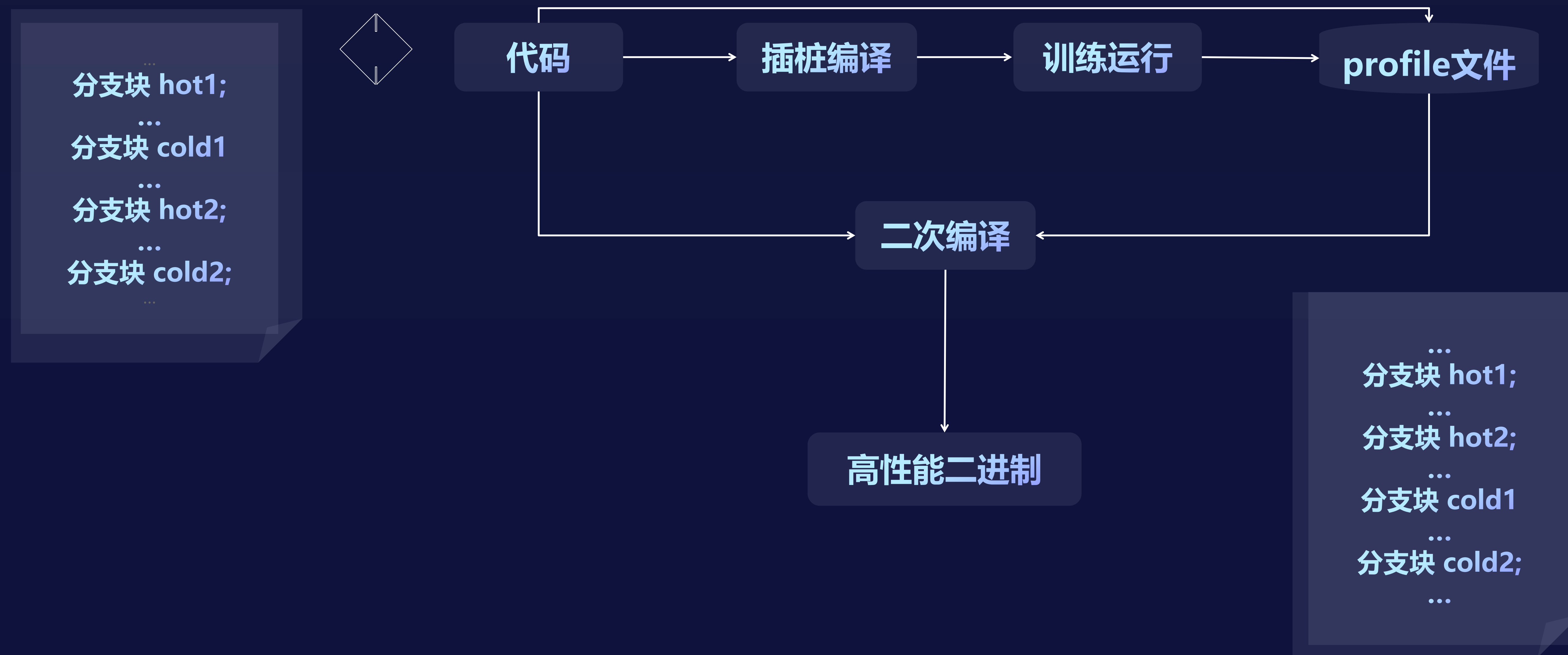
魏伟

华为毕昇编译器架构师



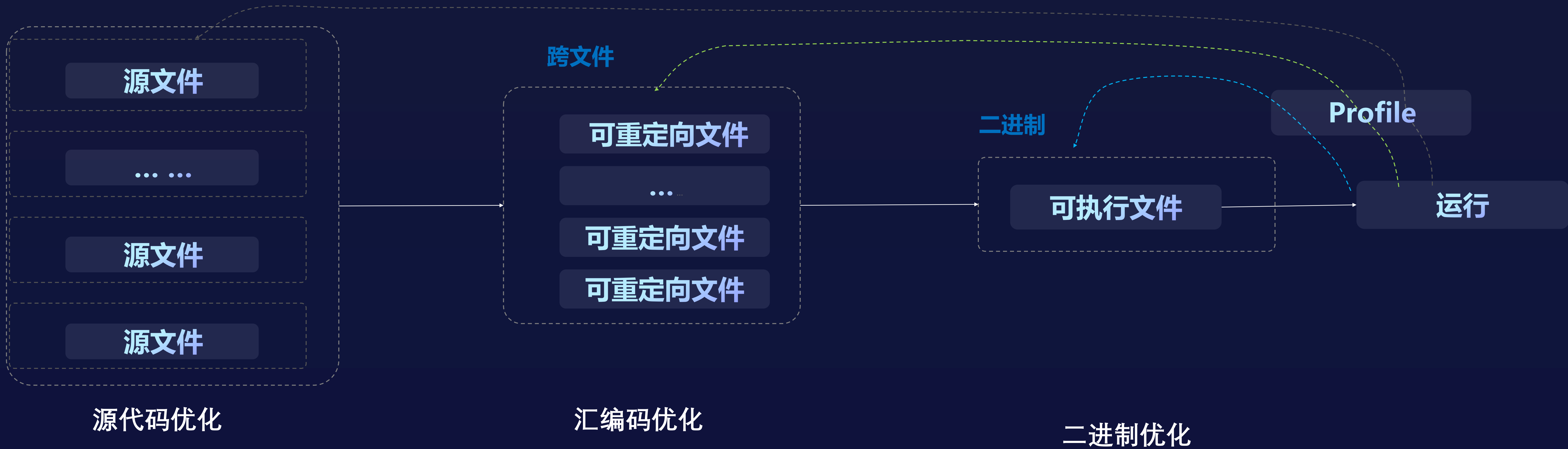
FGO (PGO) 特征指导优化

FGO (Feature Guided Optimization)：根据程序的运行行为进行性能优化，和其他优化相比，反馈优化的输入不仅仅是源文件，还包括程序的运行时数据，根据运行数据来指导编译器优化。



毕昇编译器CFGO (Continuous Feature Guided Optimization)

FGO (单文件) ➡ CFGO (多模态(源代码、汇编码、二进制)、全生命周期(编译、链接、后链接) 的持续优化)



两大关键创新:

- 1、优化对象多模态: 从源代码、扩展到可重定向汇编文件、二进制等
- 2、优化周期全流程: 从静态编译阶段, 扩展到链接时、后链接时等

优化举例1：剥离冷分支，使热点代码聚合，提升cache命中率

业务代码特征：热点函数存在冷分支，冷分支内代码用宏封装，宏展开后代码量大。



优化举例2：全局分析，优化调用栈，内存排布更优

业务代码特征：功能模块化，热点函数跨文件调用频繁。



优化举例3：超大switch结构分支调整，降低branch miss率

业务代码特征：多switch结构，case分支数较多。

Switch
Case 1: // 2041
Case 2: // 1
Case 3: // 140
Case 4: // 7819
Case 5: // 0

CFG0

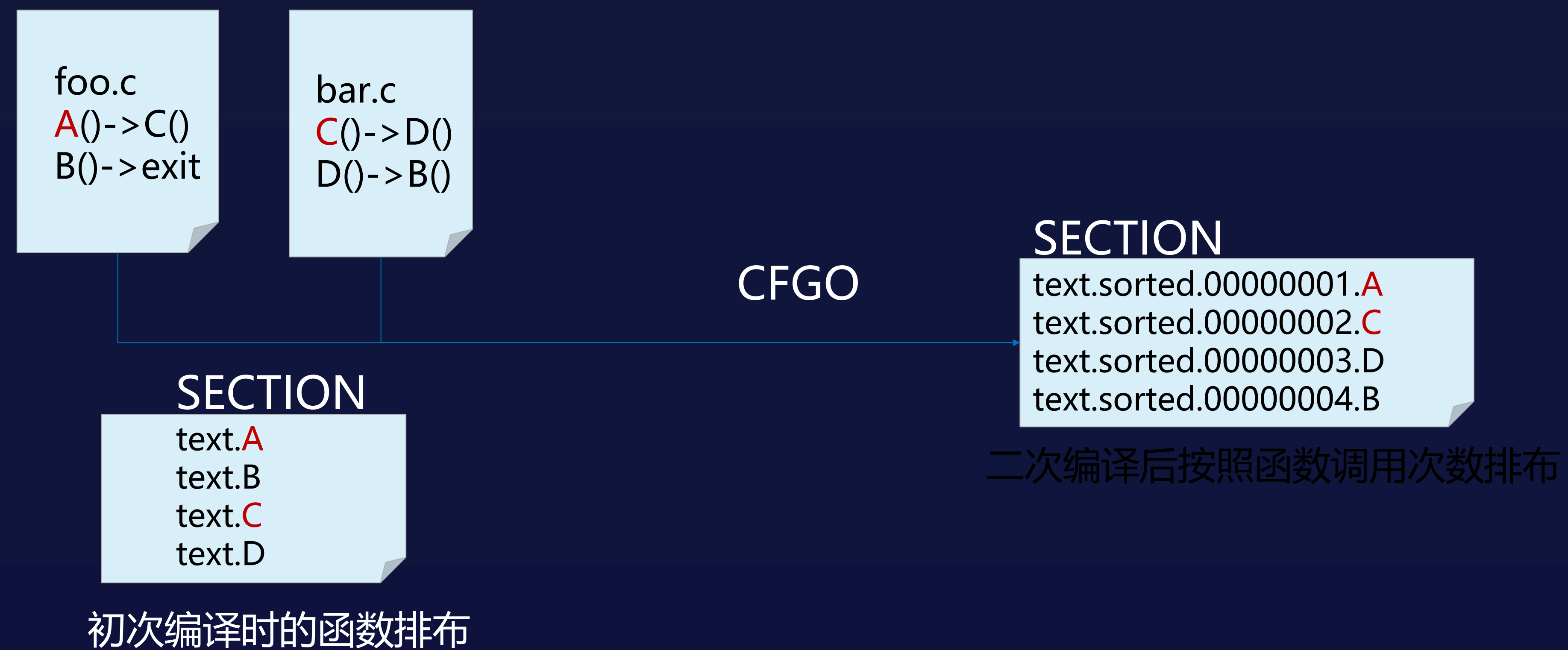


采集各case执
行次数

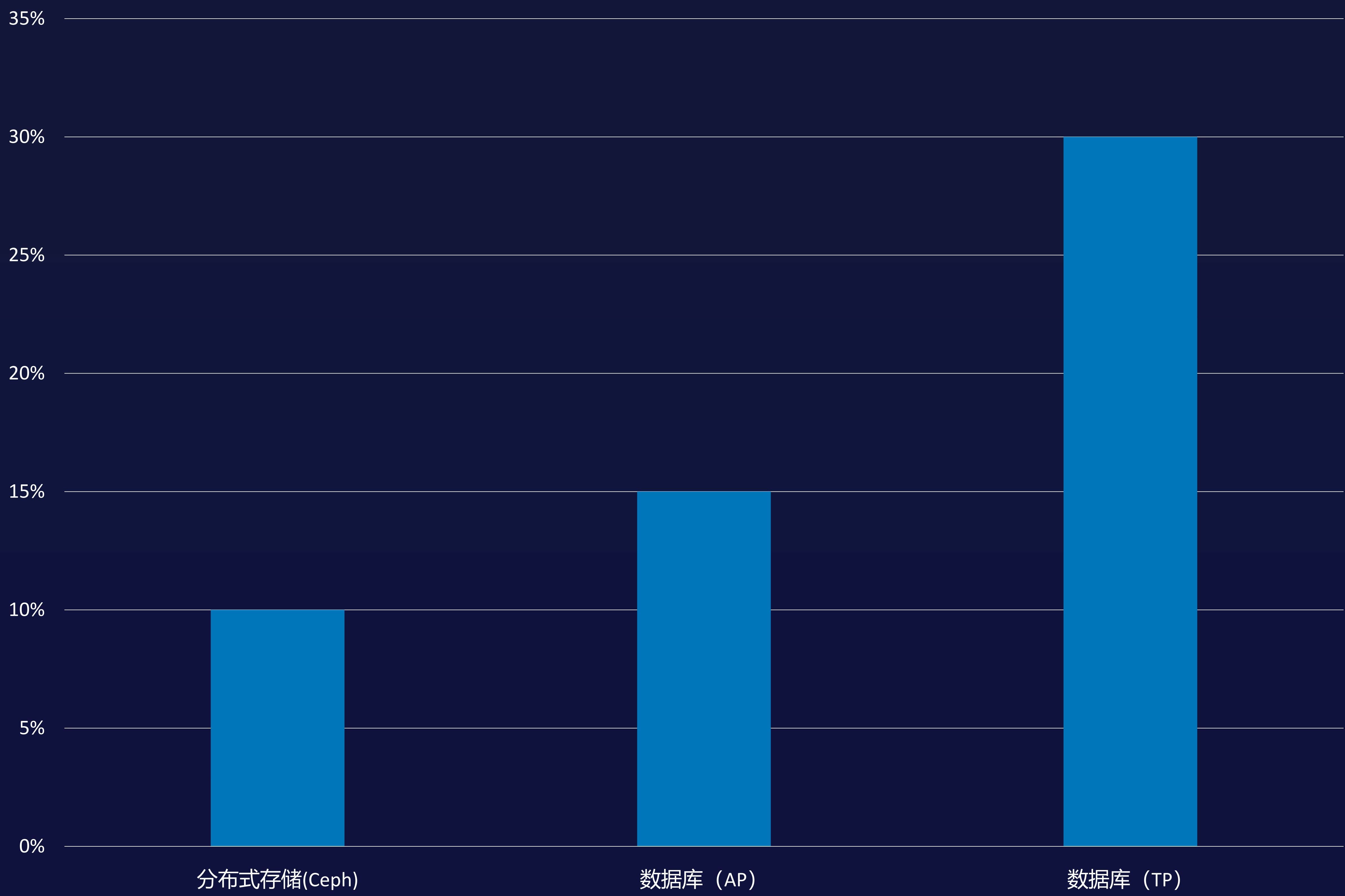
Switch
Case 4: // 7819
Case 1: // 2041
Case 3: // 140
Case 2: // 1
Case 5: // 0

优化举例4：代码段函数重排，使热点函数聚合，降低iTLB miss

业务代码特征：代码段热点函数排布散乱，iTLB miss高。



CFGGO优化技术在通用计算场景性能提升10%~30%

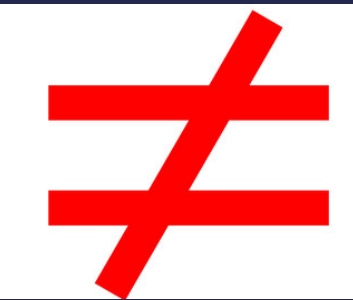
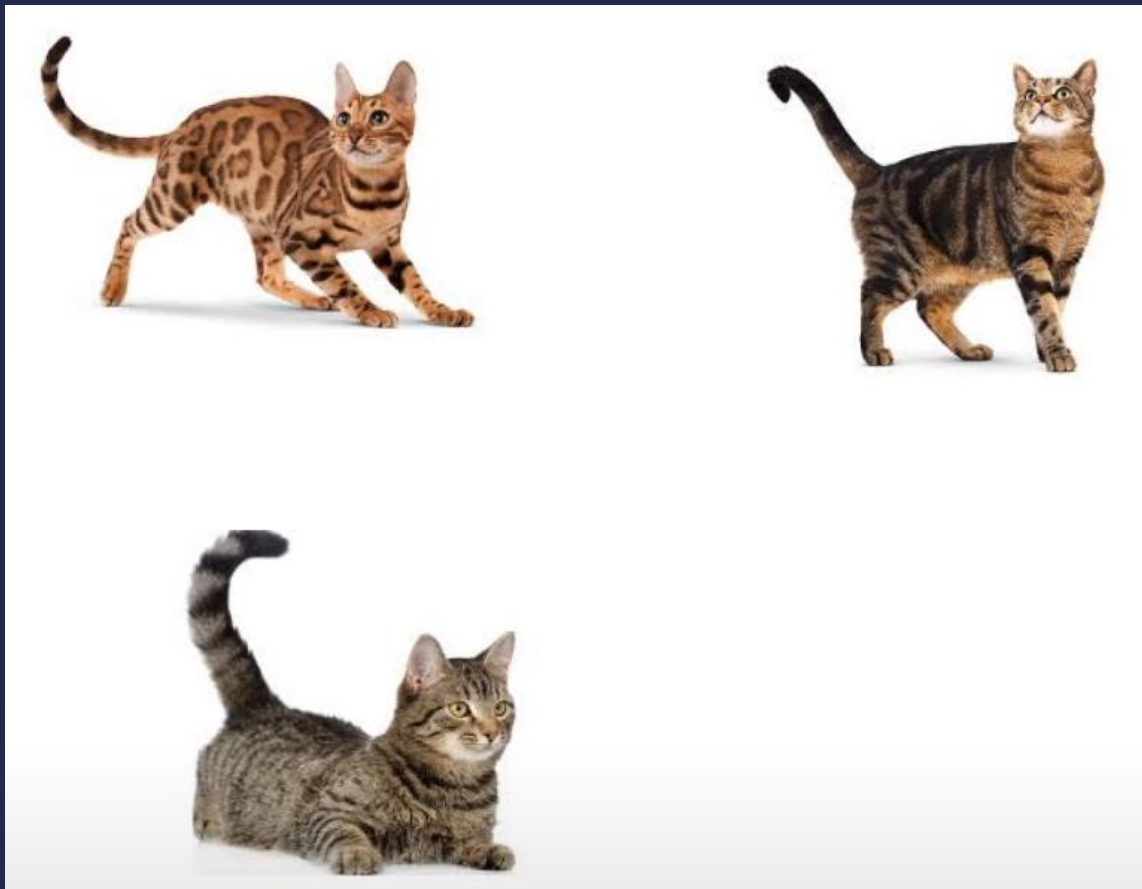


CFGGO商用落地存在问题

性能泛化性不能保障

训练集与使用场景差别大时，导致负优化

训练集



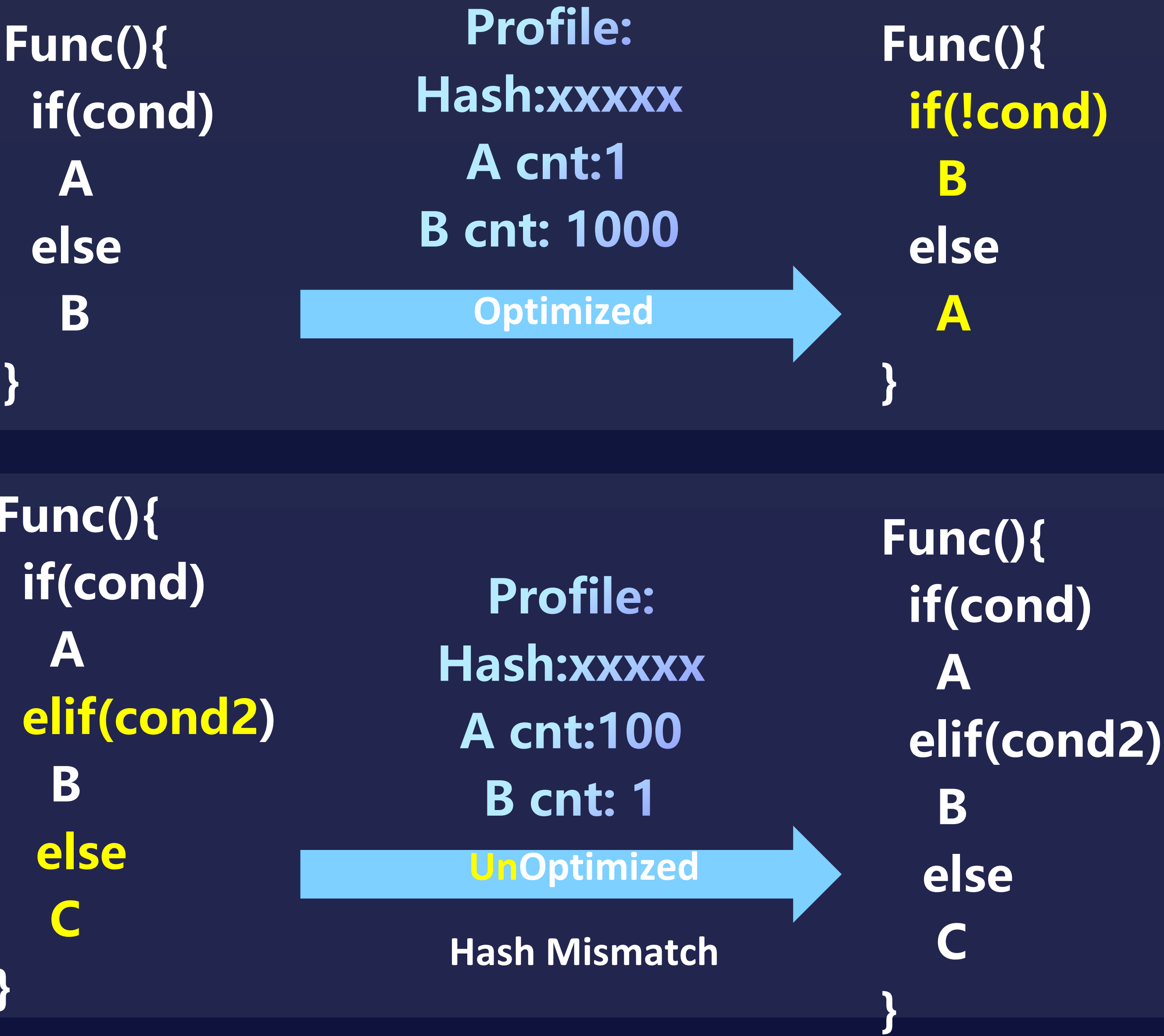
特征差异大,可能导致负优化

测试集

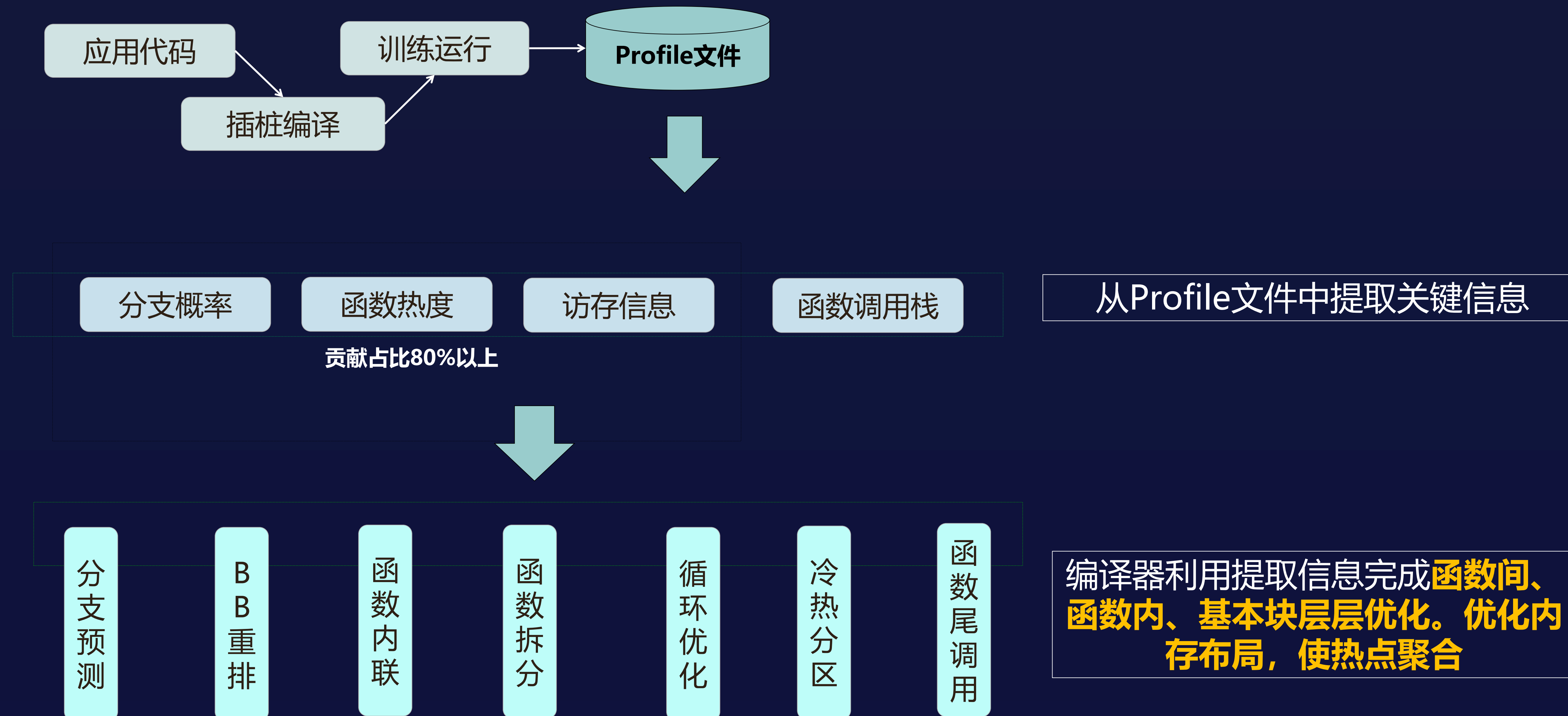


升级维护困难

代码升级后，导致优化失效，需要重新训练

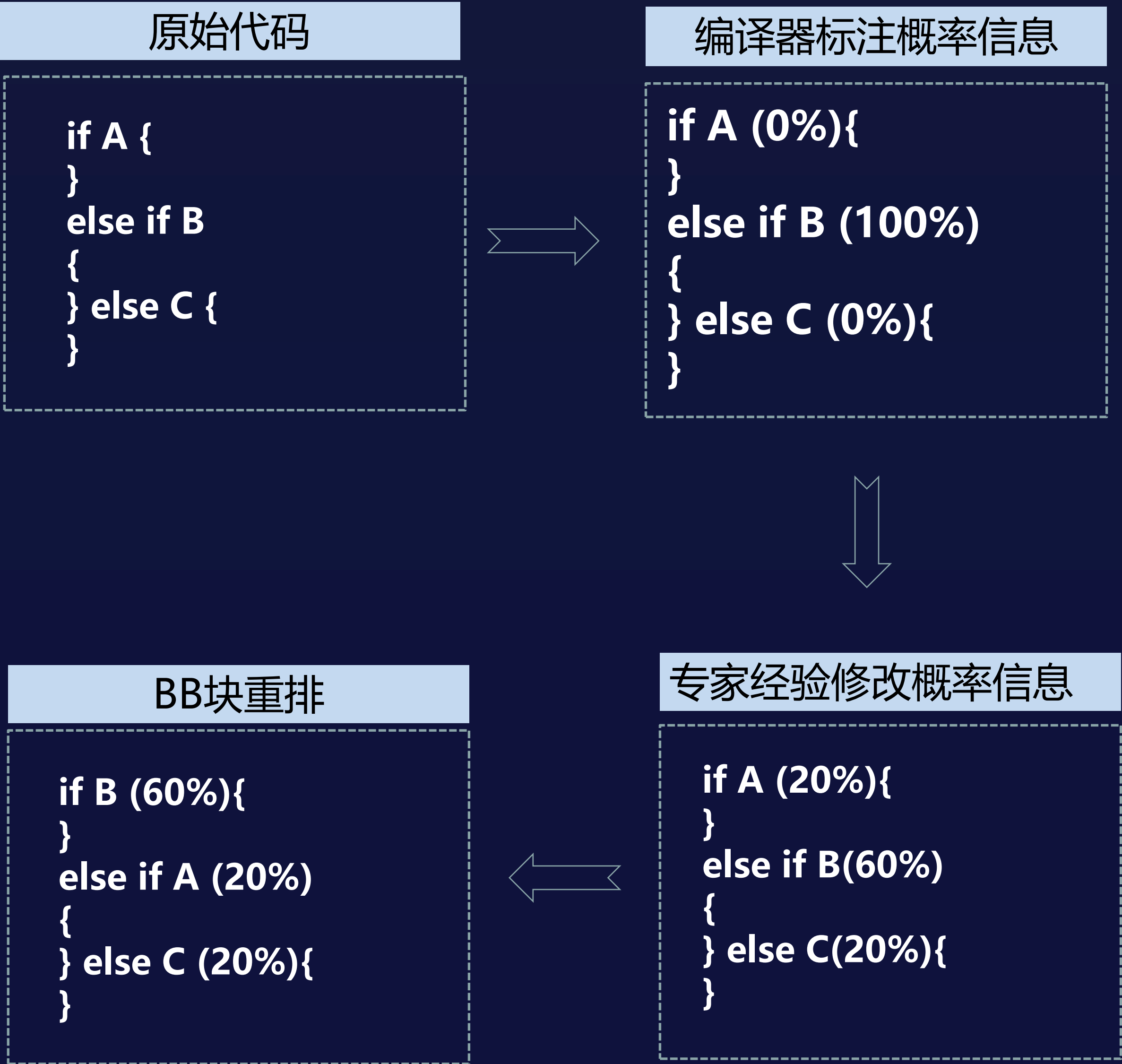


CFGO优化需要的信息：分支概率、函数热度、访存信息

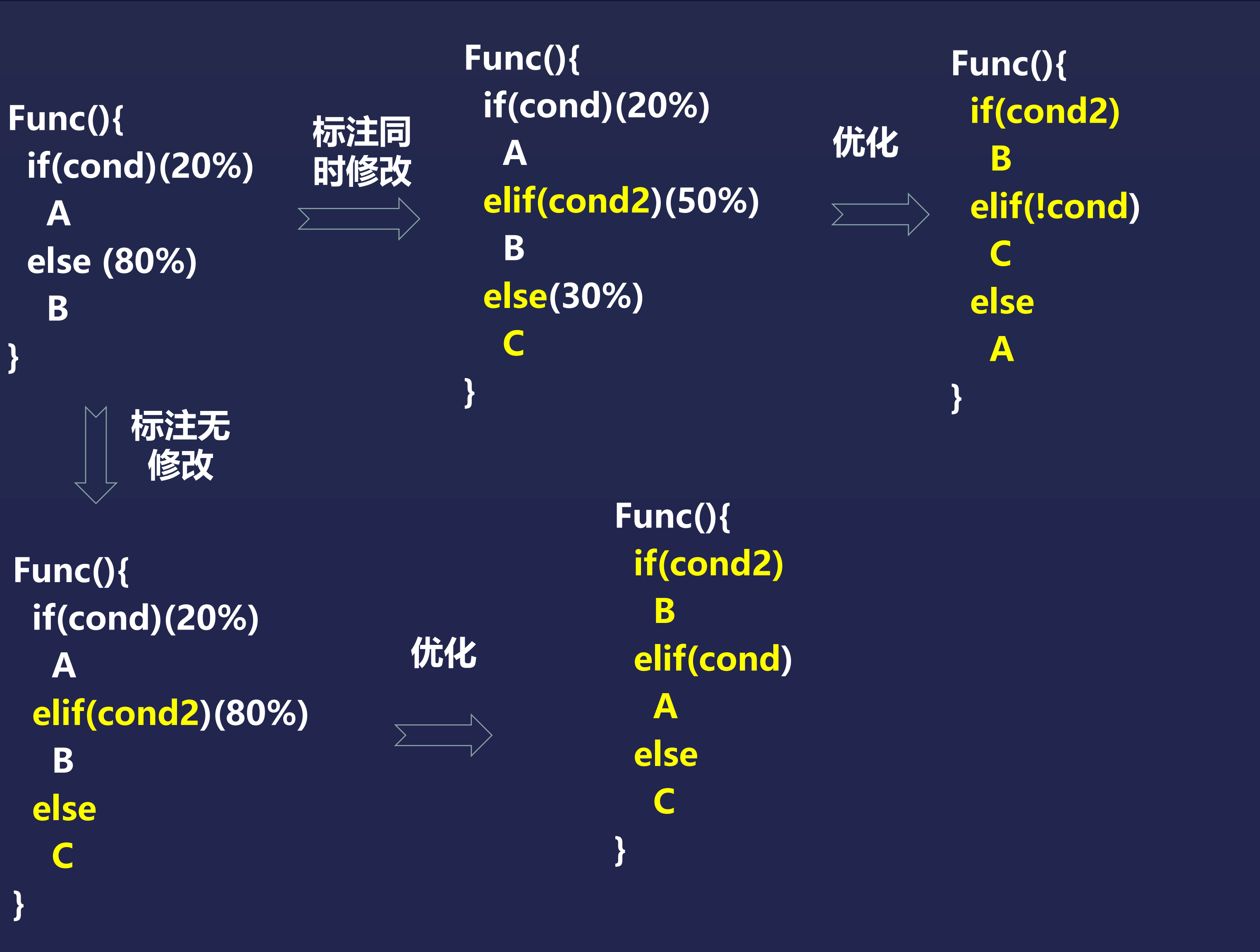


CFGGO泛化性解决方案：反馈编译优化白盒化+专家经验泛化性修改

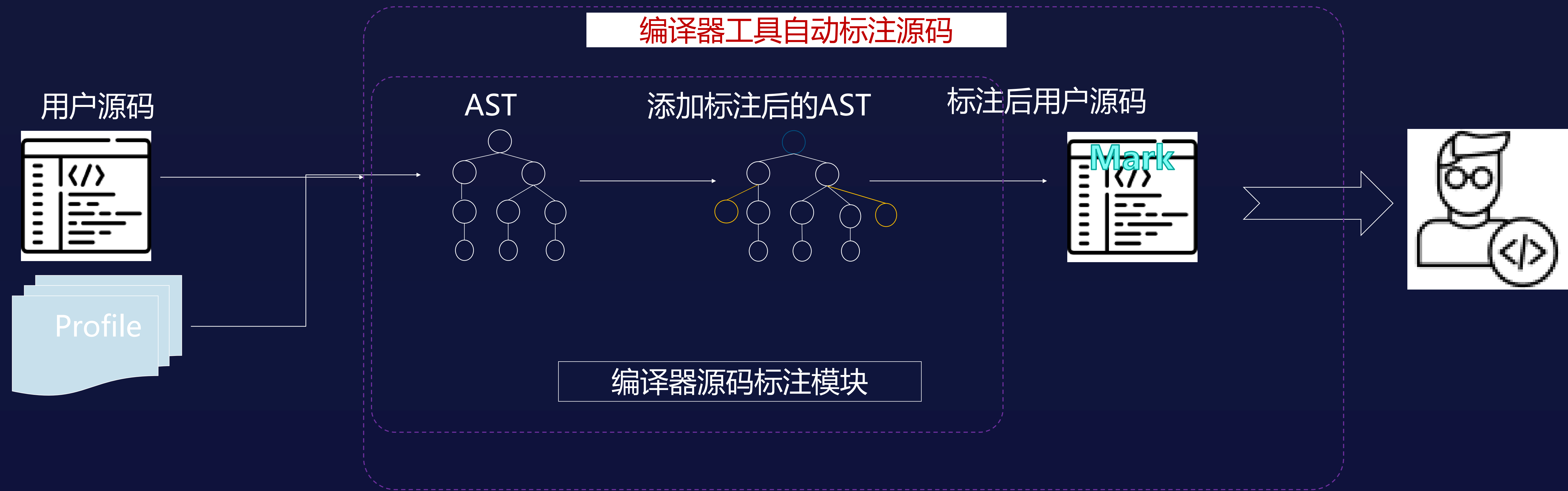
白盒化+专家经验泛化性修改



代码部分修改，不影响整体优化效果

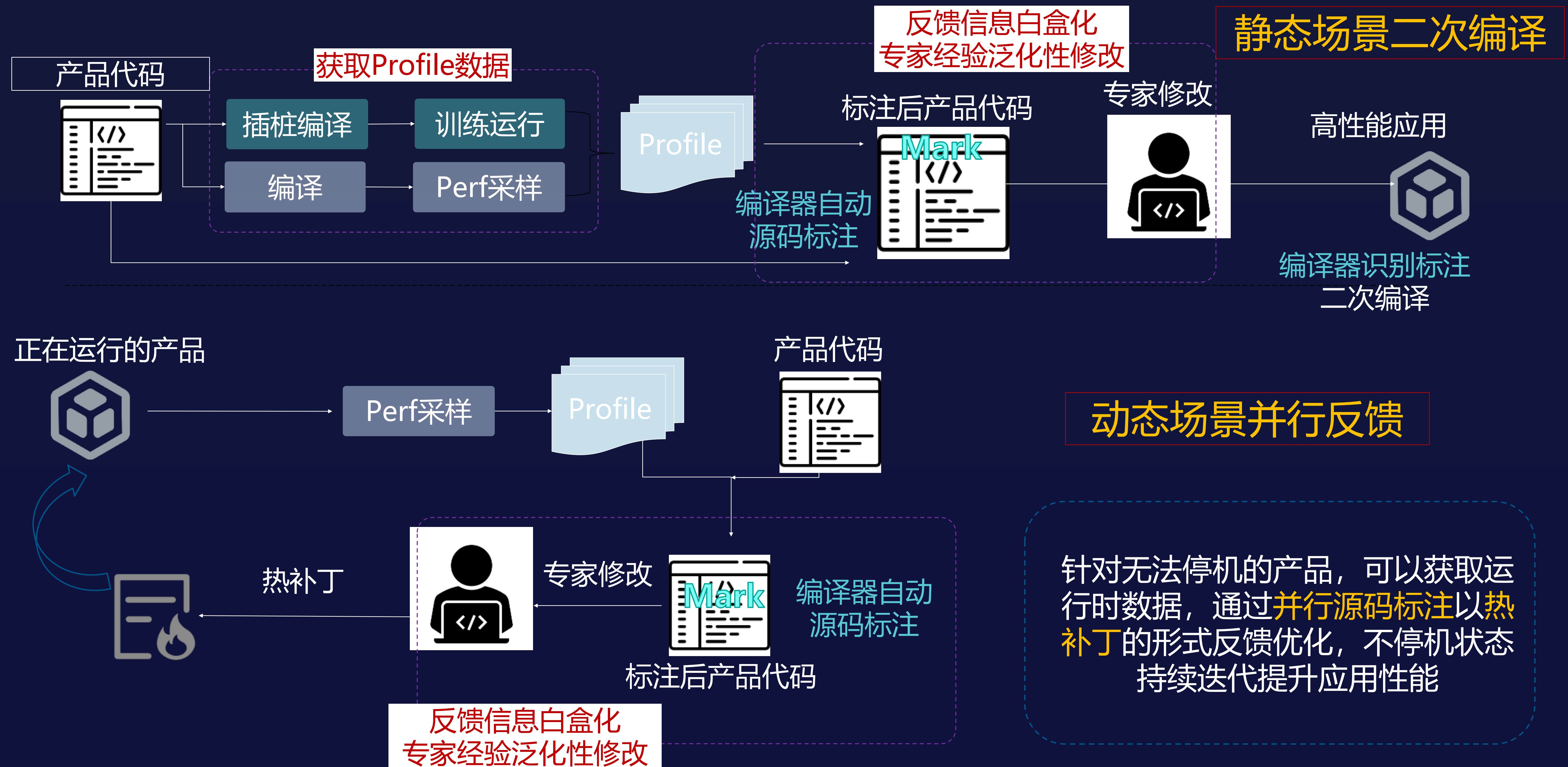


CFGO泛化性方案实现方法



典型数据库TP场景下，性能可达到PGO性能的90%+

泛化性方案可静态、动态部署



谢谢！