

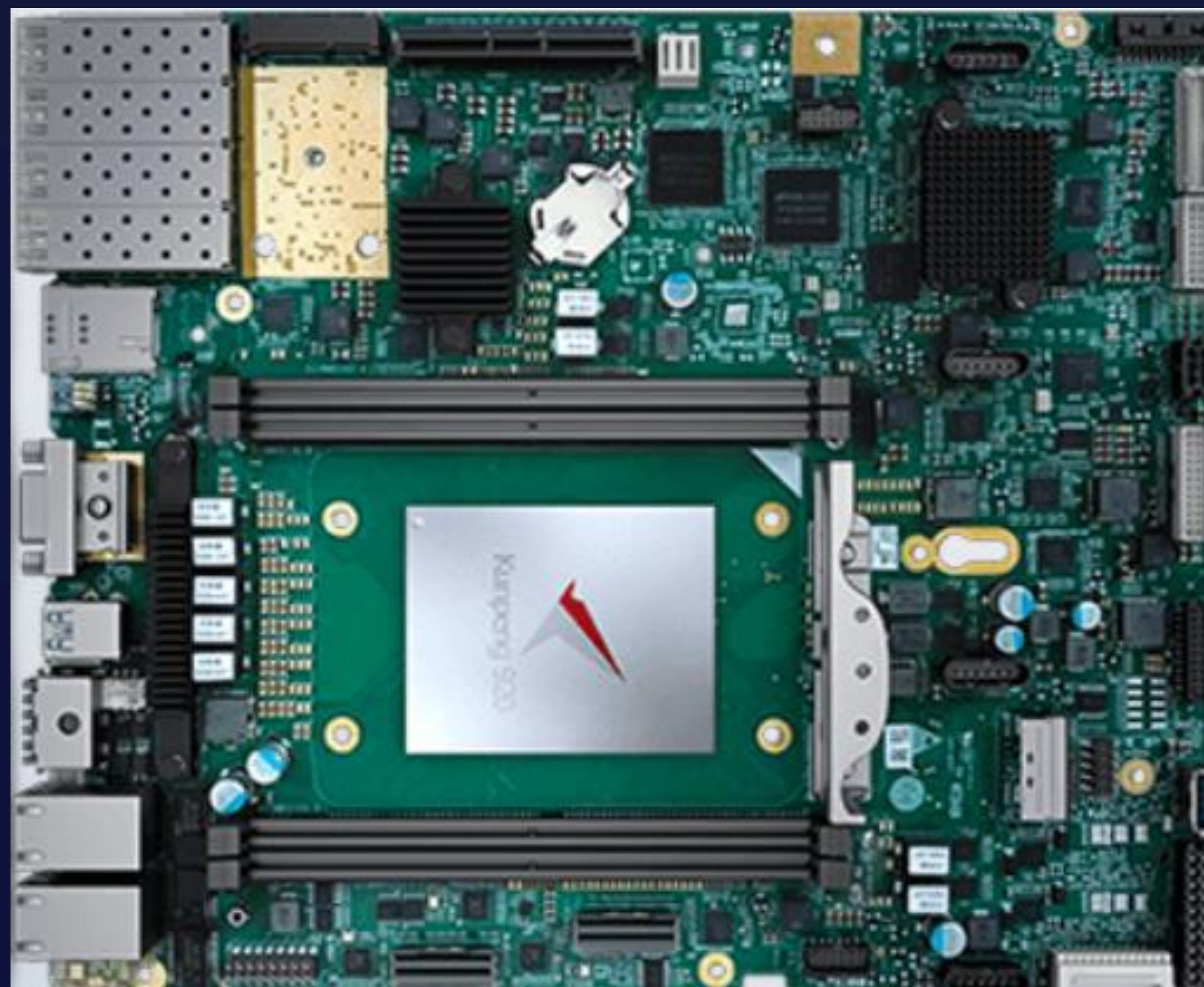
在线优化技术的实践与探讨

钟璇

华为编译器研发工程师



计算机系统与在线优化技术



鲲鹏服务器主板

提升计算
资源利用效率

提升通信
资源利用效率

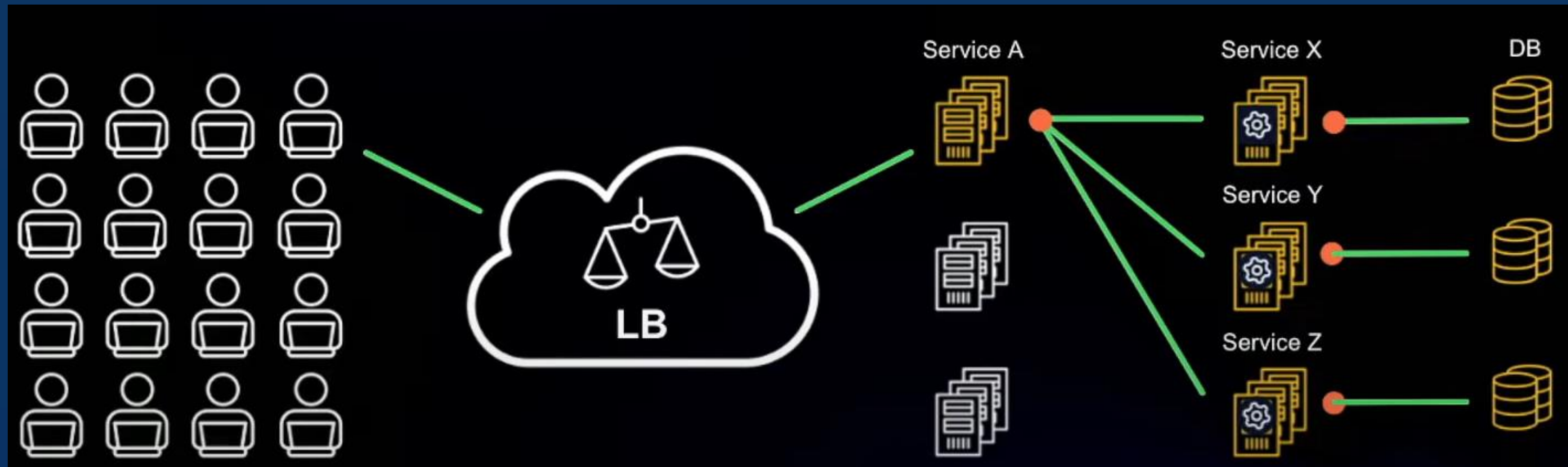
提升存储
资源利用效率

提升Power
资源利用效率

在线优化技术涉及不同维度

优化通信资源利用效率： 在线优化学习程序行为优化网络连接

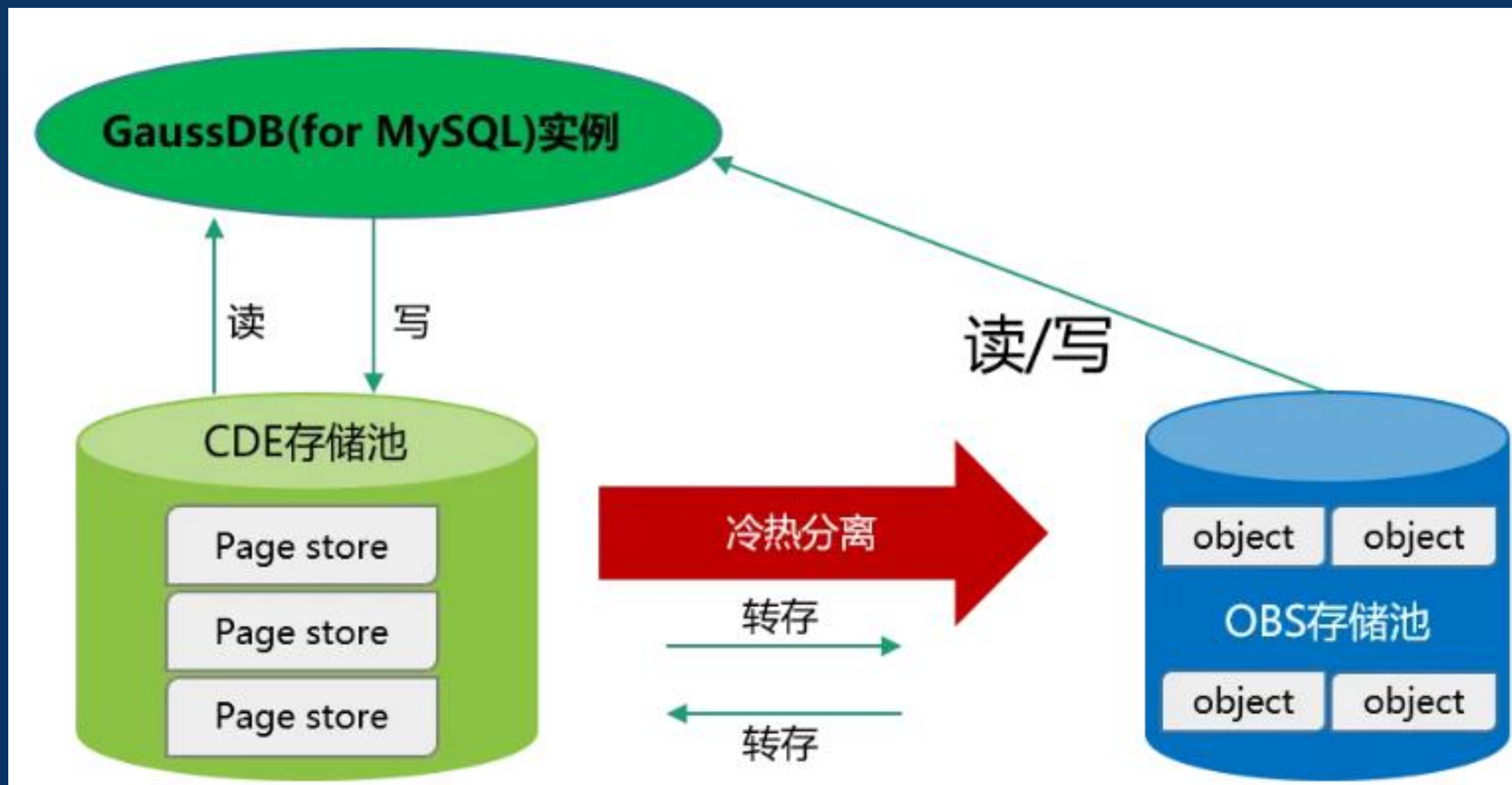
互联网服务处理流程



数据分析： 获取并分析服务请求在云计算环境中处理时的数据路径，以识别系统网络资源的使用情况。基于数据路径分析的结果，确定预期能够改善系统网络资源利用率的优化行动。

优化网络资源： 通过改变线程使用网络资源的优先级、改变连接点的类型以及预先打开连接点来减少建立连接所需的时间等不同手段优化网络资源利用效率。

优化存储资源利用效率： 热数据分层存储



GaussDB的冷热分
离

GaussDB(for MySQL)冷热存储分离新特性，支持冷表/分区可读和混合分区管理。

冷热存储分离特性能够通过一条简单SQL语句实现表/分区的灵活转储，并快速将物理页并行地从CDE转储到OBS中，降低单位存储成本。

优化Power资源利用效率： 根据App的访存特征进行DVFS

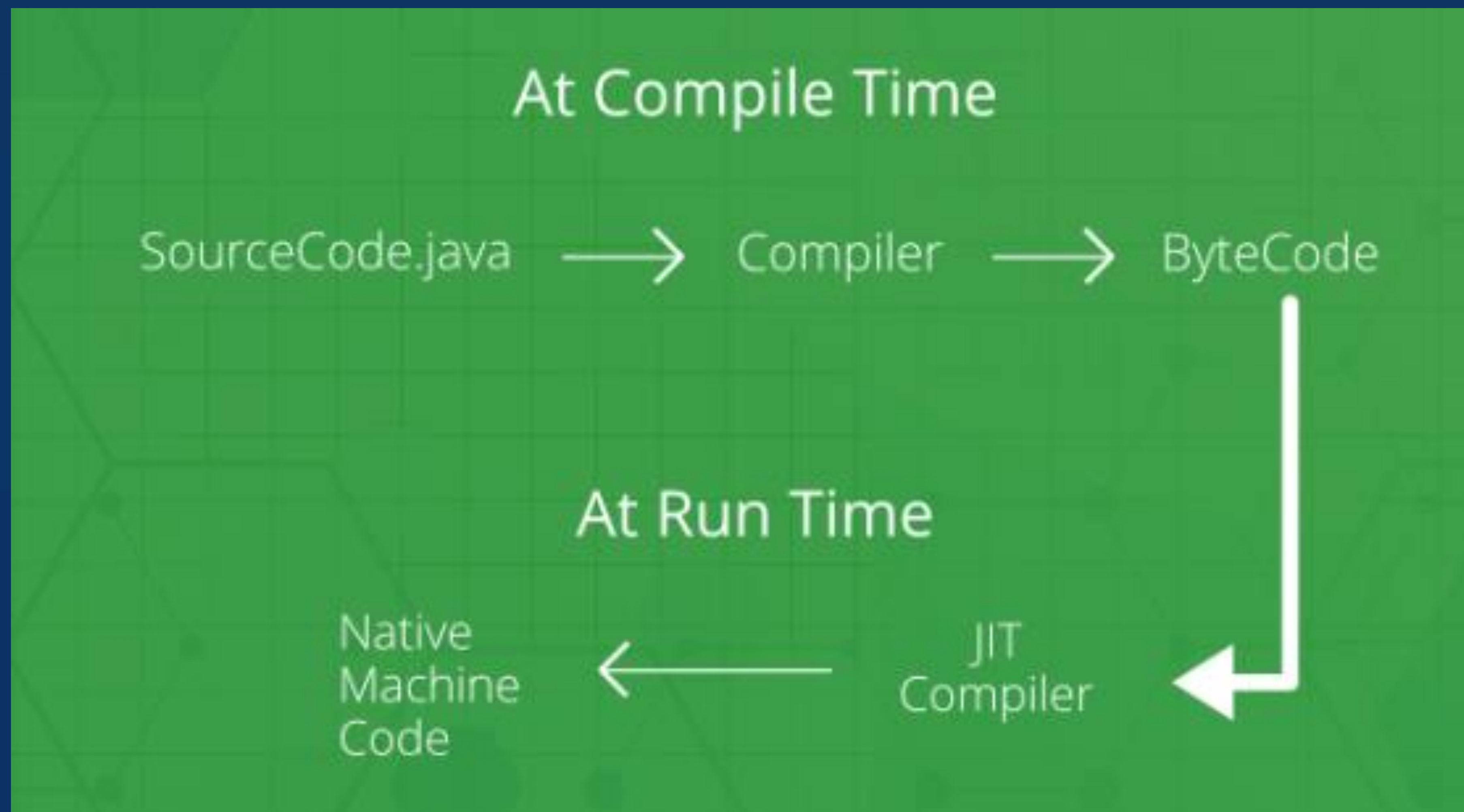


终端游戏原神

手机等终端设备对功耗敏感，经常使用DVFS技术进行能耗的优化。

游戏应用里某些复杂的渲染场景，存在大量读取存储设备上游戏资源的情况，此时后端DDR上请求多，但是CPU核的计算量较少，可以通过降低CPU核此时的频点进行节能。

优化计算资源利用效率： Java、JavaScript的JIT



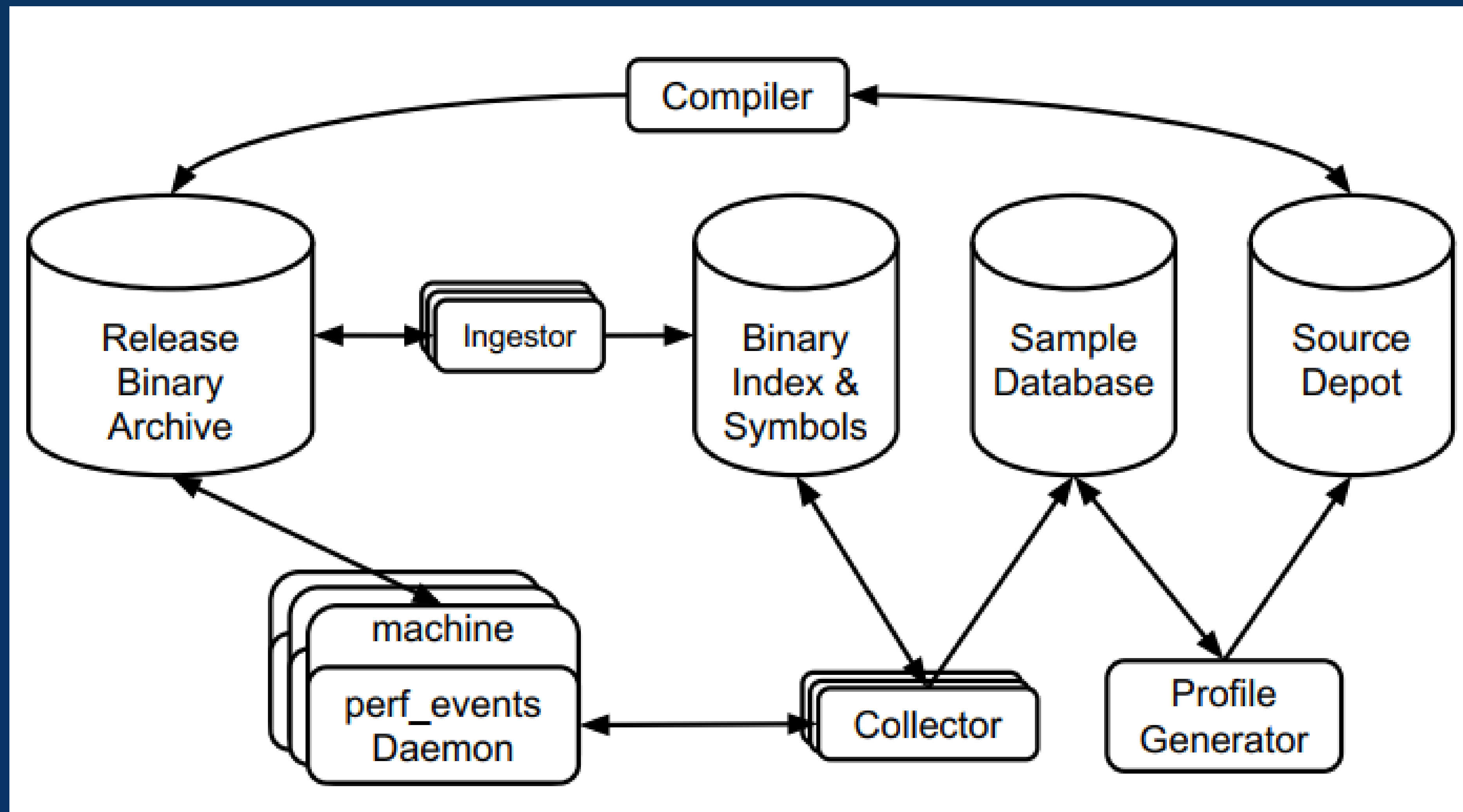
Java代码的编译流程

动态语言的程序在静态时编译成ByteCode，方便在不同的平台上运行。

运行时经常执行的ByteCode代码，会被JIT Compiler在线编译成Native的机器码。

JIT生成的代码执行效率相比解释执行效率可以有数量级的提升。

静态语言C, C++程序的反馈式优化



自动反馈式优化框架

C, C++等静态语言程序也存在根据实际的负载进行进一步优化的空间。

反馈式编译优化通过采集实际负载下的 profiling 数据, 进行下一次软件版本编译时的PGO优化

优化计算资源之毕昇BinXO在线二进制优化软件

毕昇BinXO- Binary X Optimizer

Dynamic（动态）：

解决反馈式优化需兼顾各场景的负载特征，无法针对性优化的优化的问题。直接提供无需开发者参与的在线优化，直接根据负载特征提升应用性能。

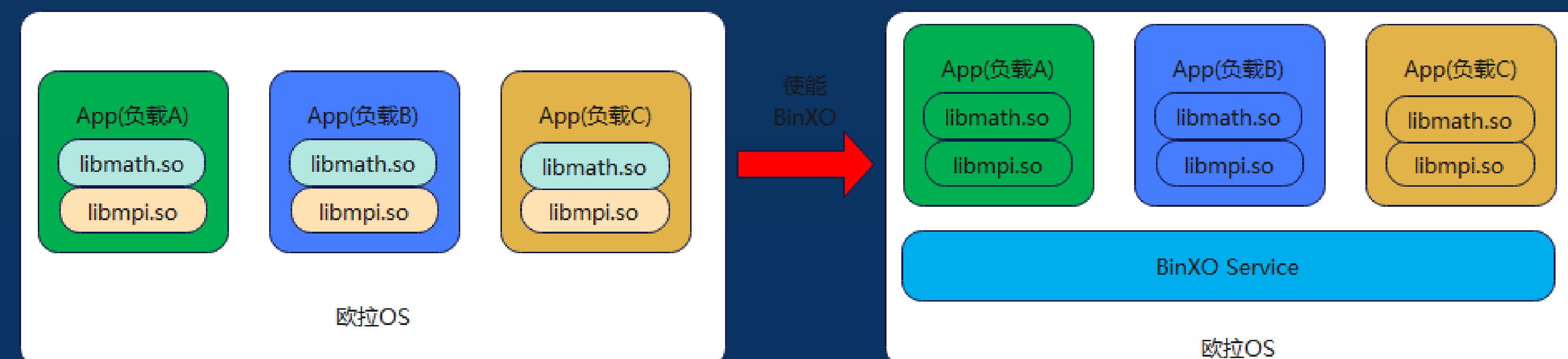
Sourceless（无源码）：

解决反馈式优化需要源码适用场景有限的问题，直接在二进制层面进行优化，不依赖源码，扩展优化的适用场景。

Boundless（跨界）：

可实现跨库的融合优化，将不同库的热代码融合在一起，提升代码空间的利用效率。

BinXO在线优化效果示例



解决的业务挑战

- 传统的反馈式优化需兼顾各场景的负载特征，无法做到全场景极致优化
- 传统编译技术无法优化无源码(三方)应用，存在潜在的优化空间未被挖掘

关键技术

- 优化自适应**：根据不同负载的运行特征进行不同的优化
- 跨库的融合优化**：不同库的热点代码融合在同一个Code Cache
- 无源码三方库的优化**：三方库的控制执行流CFG识别