CCF-华为胡杨林基金 系统软件专项2023年指南发布
编译器与编程语言课题指南

2023年4月7日

HUAWEI

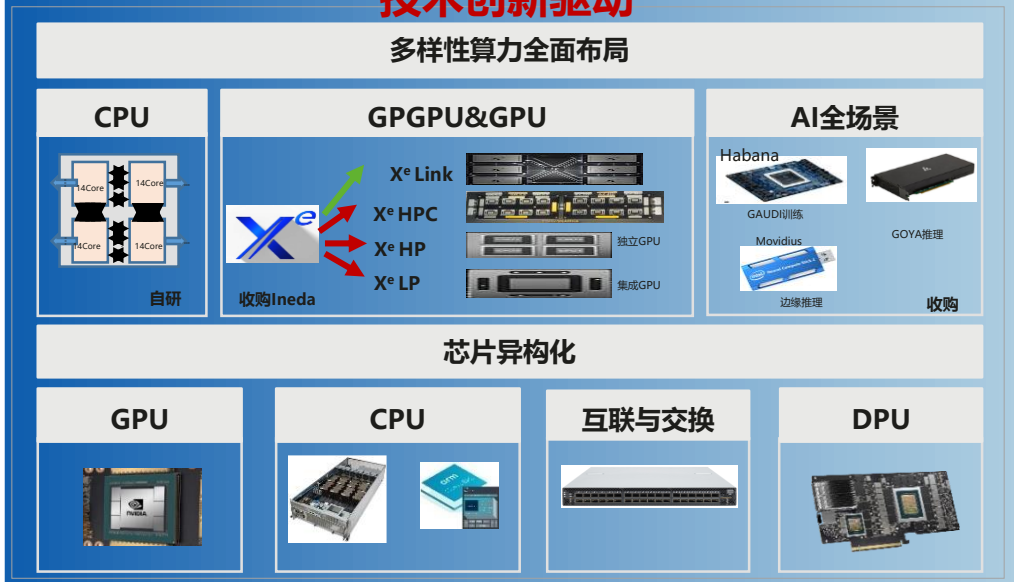
编译器与编程语言领域面临的挑战与创新机会

后摩尔时代创新趋势

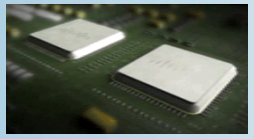
应用场景驱动

- 图形渲染，深度学习, HPC等计算密集数据密集应用，驱动异构算力创新，培育新生态
- HPC-AI等混合精度应用，AI训练和推理，视频分析，难以由单一算力承担，应用发展趋势在各行各业逐渐明显
- HPC、AI集群计算对算力利用率仍比较低，算力发展与应用性能和资源利用率不均衡

技术创新驱动



多样性算力面临的挑战



AKKD
昇腾
鲲鹏
麒麟
……

编译编程核心领域创新

编译与编程领域创新方向

编程语言方向创新： 新编程语言特性，用以解决既有编程语言所不适应的新场景中的编程问题。如，跨语言数据结构转换、不同内存管理机制的统一与交互，以及AI、HPC等场景的语言设计等

编译器方向创新： 面向新计算机体系结构、异构并行、新介质的编程模型，编译器，软件并行化工具创新。如，针对通用计算的细颗粒度异构/并行编程模型，感知存储介质与时延的编译优化技术等。AI技术在编译关键特性、编译优化、及自动编程的探索等

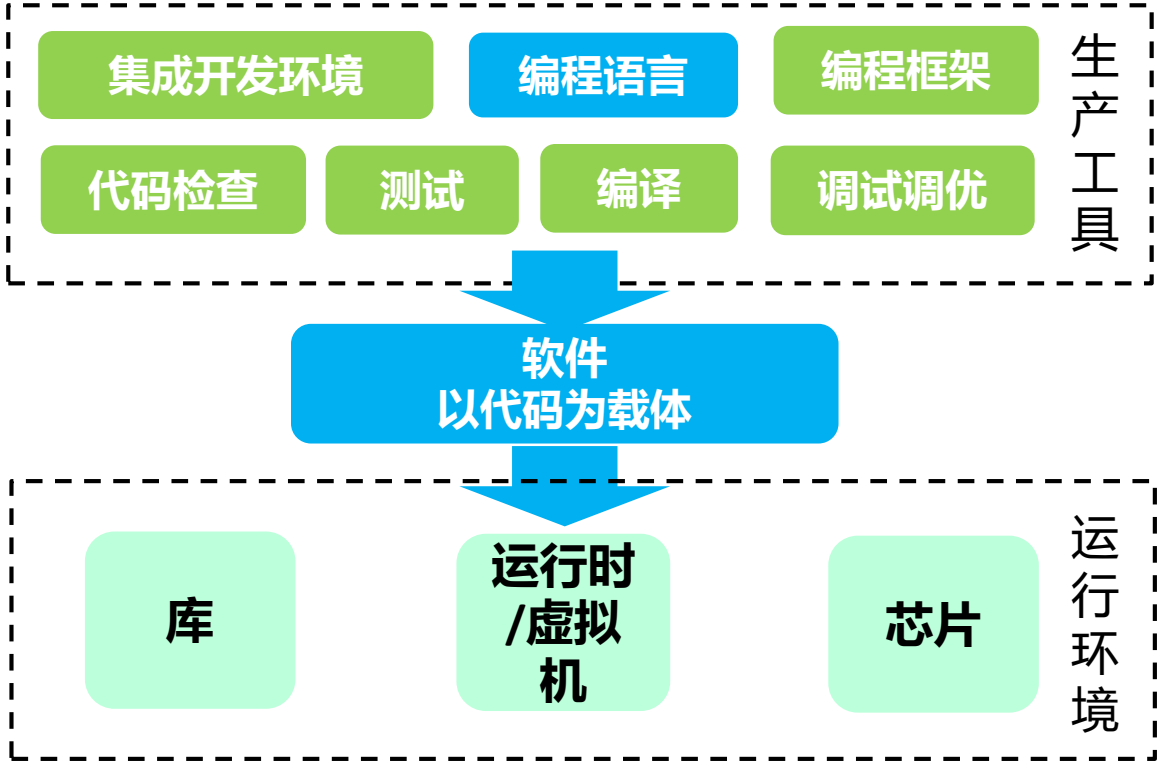
软硬件协同方向创新： 探索细颗粒度并行的新架构方案与设计，面向领域的定制化可编程架构设计，或者面向通用开放场景的软硬件垂直架构优化

语言虚拟机方向创新： 面向云原生和AI等新兴应用场景，探索下一代跨语言运行时，在“高性能、高安全、轻量化、高可移植”等方向进行创新

IDE方向创新： 围绕开发者应用全流程与AI相关技术结合，探索下一代智慧化IDE平台，在AI代码生成、高精度程序分析以及智慧化底座等方向进行创新

面向以低工艺芯片达到业界领先的算力，AI自动化编程，新应用场景、新软件形态、新体系结构、新存储介质、新语言/编译技术等方向驱动编译器及编程语言技术创新

编程语言：直面开发者，承上启下，既要软件开发效率，又要硬件极致性能



- 编程语言既是软件的生产要素和生产力工具，又是承载软件发展演进的智力资产。
- 在整个软件生态中，编程语言直面开发者，作为生产工具构建的基础，贯穿于软件生产的整个流程（开发、构建、测试、发布、维护），**助力降低开发门槛，提升软件生产力；同时培养开发者习惯，增强生态粘性**；
- 编程语言作为软件发布的载体，充分与软件运行环境（运行时、虚拟机、库、芯片）协同，**构建极致的软件体验（性能高，可移植）**；

自研编程语言：生态构成与关键特征

自研编程语言生态构成

关键特征

强安全

静态类型系统、垃圾回收、Null Safety等设计保证类型安全和内存安全，消除未定义行为，编译通过的程序运行安全、可靠

高性能

值类型、high level IR编译优化、高性能内存管理和轻量化运行时等设计确保华为自研编程语言在各种平台都能高效运行，且内存消耗低

高效率

现代化简洁语法、多范式编程、类型推断降低学习门槛，IDE提供良好的开发调试体验

全场景

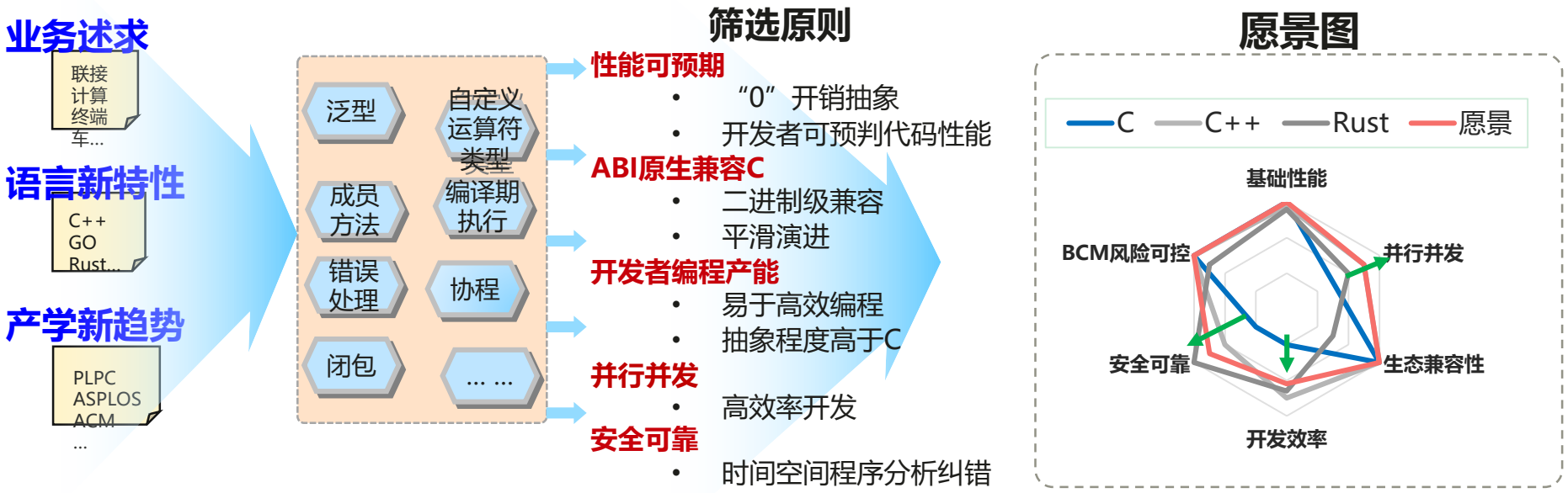
支持静态编译和虚拟机两种编译运行方式，高效跨语言互操作机制，提供元编程能力便于构建领域专用语言（DSL）

下一代系统编程语言：探索“最适合”华为的系统编程语言形态

华为CT领域系统编程，性能敏感、资源敏感，开发主要以C为主

- **大量CT嵌入式场景性能敏感，C语言仍然是主力的开发语言，但是随着业务量的庞大， 在安全、并行以及开发效率等方面遇到了挑战；**
- C++多样化特性符合计算产业未来演进趋势判断，计算领域语言技术正在积极布局；

愿景：探索“最适合”华为的系统编程语言形态，打造下一代系统编程语言



- **并行并发：**
  C, C++ < 毕昇C == Rust
- **安全可靠：**
  C, C++ < 毕昇C < Rust
- **开发效率：**
  C < 毕昇C < Rust, C++
- **基础性能/生态兼容：**
  毕昇C == C

兼容C语言，保持高性能、轻量化的前提下，扩展高并发、高安全、高开发效率竞争力

编程语言领域技术创新挑战

问题和挑战

新型应用编程语言的问题

新型语言基础竞争力挑战：

- 高级语言都有配套的程序分析工具，对于**新型语言分析规则**如何**制定**，是否继承现有语言，需要方法论支撑
- 新型语言漏洞扫描工具降低漏报率和误报率，提升用户体验
- 新型语言与现有语言性能对比需要理论支撑，和新的工具支持，并且降低性能分析对程序本身性能和底噪的影响

新型语言生态构建挑战：

- 语言生态基础设施是否完善是开发者选择一门语言的关键因素之一，是语言保持活力和持续发展的动力，语言生态基础设施包括系统服务、框架、中间件和三方库，涉及众多领域，专业性极高，包括分布式，网络，安全，程序分析，科学计算，图形图像，地理信息等；如何快速高质量的构建有竞争力的生态基础设施充满挑战。

新型语言质量挑战：

- 语言库的质量保障工具可以为仓颉语言的开发者提供更高效、便捷的自测能力
- 语言特性的交互场景较多，测试覆盖的难度较大，基于覆盖率引导的Fuzz等能解决一部分，但无法作为基线用例使用，也缺乏黑盒层面的覆盖评估

计算场景算力需求挑战：

- 计算场景算力需求通常来源于高维数据、循环和复杂运算，如何挖掘程序并行度来充分利用硬件平台的算力能力

系统编程语言的能力提升

- **安全：**海量存量C代码，存在大量内存安全问题；形式化验证证明“绝对安全”
- **效率：**C语言开发效率低，冗余代码多，异步并发表示能力弱，并发改造难度大
- **并行并发：**并发接口抽象层次过低，改造困难；并发编程与业务耦合高

关键技术创新方向

★ 高优先

★1、程序分析：

- 高级语言配套的**静态漏洞**扫描工具，通过**方法论**支撑扫描规则指定
- 新型语言**漏洞扫描**工具**漏报率和误报率降低**，并进行**自动修复**
- 新型语言**性能分析工具**，语言性能对比分析**方法论**，性能分析工具如何**降低**对程序本身**性能影响**
- 通过语言设计和编译技术针对高维数据、循环和数学运算（微分、FFT等）提升程序在硬件上的**并行度**从而**优化性能**

★2、形式化验证：

- 通过**形式化验证**手段**证明系统安全编程语言**在安全区中“绝对安全”

★3、系统语言增强：

- C语言安全增强：在兼容C语言的前提下，扩展**内存安全、时间安全、空间安全**的能力，探索不依赖生命周期的时间安全可能
- **新型编程范式**：在并行原语的基础上提供表达力，进一步降低开发门槛

★4、软件测试与验证：

- **语言库用例自动生成**：通过语言库各接口签名的分析，自动生成单接口的功能、性能、压力及Fuzz测试用例，后续可对接符号执行或基于覆盖率引导的Fuzz等方式，自动生成相关测试输入，完成UT用例的覆盖。
- **语言基础用例自动生成**：依据 BNF 规则，及各条语义规则描述，基于初始的用例，在一定组合覆盖程度下，进行枚举式的等效变异。自动生成对应规则的正、负向用例。

★5、竞争力语言基础设施

- **新型编程语言**和领域知识**融合创新**，构建高可靠语言基础设施，包括系统服务、框架、中间件等
- 业界主流语言**高频特性调研，**指导新型编程语言特性选型
- 面对新型语言替换存量代码，**支持兼容已有生态**，代码转换工具成功率增强

HUAWEI

语言虚拟机技术创新挑战

★ 高优先

问题和挑战

自动内存管理：

- 现有的用于评估GC性能的测试套涉及大量且长期的人力投入，通常需要人为抽取业务特征，较难大规模展开；且形成的测试用例无法复制到新的编程语言上，制约了新语言上的GC性能评估

Java虚拟机编译优化：

- 大数据等场景中，对性能敏感的部分工业界目前的最佳实践都是改用C/C++的Native实现或者提供新的Java API，但混合语言的实现，在开发成本、维护、部署、扩展、移植方面都不够友好
- 云原生等场景中，启动速度和性能达峰速度影响了服务的冷启动性能，进一步影响云服务的伸缩性能

并行机制（编译与运行时支持与优化）：

- 语言引入并行编程表达，如何在IR上表征数据并行信息，任务并行信息，任务依赖信息等，赋能与归一化并行模型表达能力，探索并行优化，提高代码并行性能。
- 系统编程语言选型stackless方案实现语言级别的标准协程，其代码编程存在约束，且深层嵌套场景栈跳转相对低效（逐层栈返回）

关键技术创新方向

★ 1.自动内存管理 – 华为自研编程语言：

- 构建基于华为自研编程语言的GC性能测试套

★ 2. Java虚拟机编译优化 - 毕昇JDK：

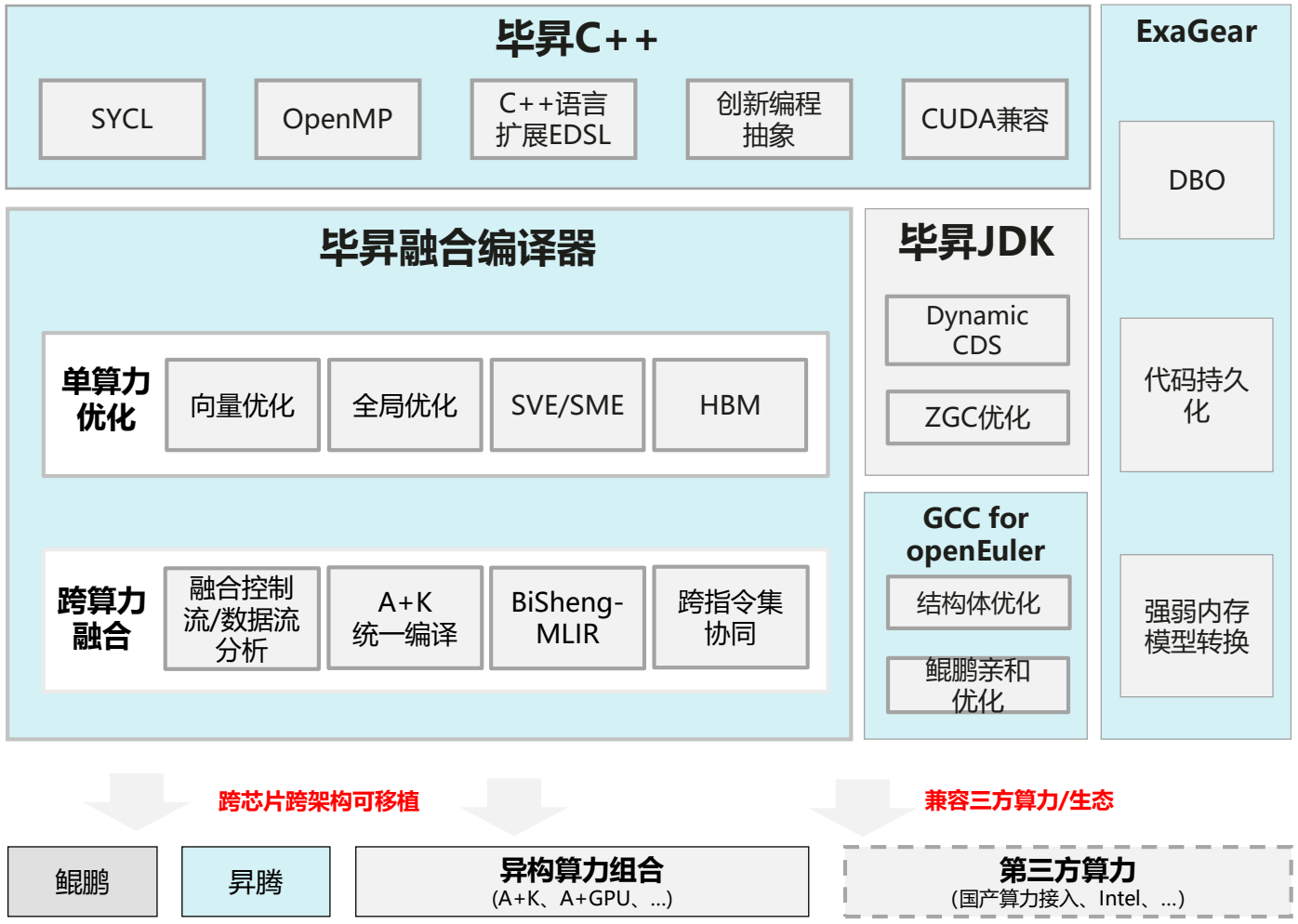
- JIT编译优化：通过在C2编译器上实现更优更激进的JIT优化，提升Java峰值性能。包括但不限于inline、逃逸分析、自动向量化、代码重排等技术
- ★ 混合编译技术：探索一种在不损害兼容性的前提下，通过结合AOT/JIT代码及相关辅助数据的缓存、共享、分发的方式，大幅提升Java服务冷启动性能

3.并行机制（编译与运行时支持与优化）

- 并行语义IR设计与优化：探索IR层级对目前关注的DLP和TLP的表达机制，以及可能的优化
- 面向系统编程语言低开销，细粒度并行机制：探索与目前主要技术路径（stackless）不同的技术路径（比如cactus stack，stacklet等stackful协程的实现机制），理解细粒度、轻量级有栈协程的实现开销（内存管理开销和空间开销）

毕昇编译器：高性能、高可信及易扩展的编译器工具链

业务全景图



关键商业竞争力挑战

毕昇编译器（for 鲲鹏）

- **基础性能**：突破基础性能天花板，在SPEC/CPUBench上体现性能优势
- **主力场景**： HPC瞄准国超、区超Top应用持续优化，竞争力领先
- **下一代芯片**：关键技术逐步落地，基于HPC/音视频等关键场景验证
- **行业推广**：HPC技术打穿，规模上量；突破金融、政府等行业头部客户，树立标杆
- **openEuler**： openEuler原生编译器，打造垂直整合竞争力

毕昇编译器（for 昇腾）

- **当代芯片**：构建动态shape及二进制发布竞争力，贡献关键场景性能提升
- **下一代芯片**：Hybrid-ISA混合编译、FixPipe多通路融合等，助力昇腾算力较上代提升x倍

毕昇C++融合编译器

- **标准语言扩展**：完善标准支持，突破语言抽象和Plugin EDSL技术
- **创新编程抽象**：SIMT on SVE、UMMI on SME，简化多算力编程难度
- **昇腾算子开发**：内存/计算/同步抽象，提升算子开发效率和性能

指令翻译（ExaGear）

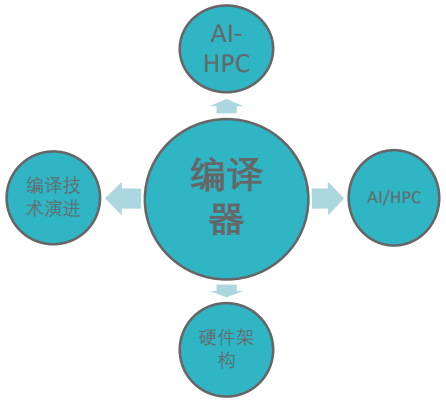
- **启动加速**：构建代码持久化技术，加速启动时间，节约能耗
- **内存序问题**：探索Binary lifting技术，改善强弱内存模型引起的性能损耗
- **下一代芯片**：支持AVX512指令，支持x86强内存序二进制翻译

编译器领域技术创新挑战

★ 高优先

关键技术问题和挑战

在后Moore时代下，当前业界共识的编译器下一跳技术方向是服务化和场景化。结合华为计算产业特点，编译器在AI、HPC、数据库等场景驱动下，除了进一步利用垂直整合设计思想提升性能外，也将引入横向融合的交叉范式，突破编译器本身和场景在跨算力编程、调优、性能、移植性等方面的关键技术瓶颈和挑战。



【技术问题和挑战】

（1）编译器/场景：**AI大模型** - 高性能dynamic shape算子生成；**HPC场景** - 架构迁移的性能提升和泛化；**AI4S等未来开放场景** - 多层次编译基础设施；**嵌入式场景** - CodeSize限制；**数据库场景** - 持续性能提升的联合编译优化挑战；

（2）编译器/硬件架构：异构编程抽象以及指令流水相关的同步、并行；DSA架构性能建模和自动调优工程；间接和离散等非规则访存的指令集；DSA的SIMT抽象；

（3）编译器/AI：AI for Compiler；AI for auto-tuning; Automatic Compiler Generation；HPC for AI

（4）编译器/并行计算：多Kernel差异分析和高并发融合编译问题；大规模编译并行寻优问题；云化并行编译难题；超节点异构编程编译问题

关键技术创新方向

1、异构编程编译：

- 昇腾芯片编译优化：SIMT over SIMD、多流水线并行与自动同步、DSA性能模型-编译联合优化 …
- ★ 昇腾高性能AI算子： 编译参数化建模、AI for auto-tuning、基于代价模型的dynamic shape算子生成
- 鲲鹏片内异构编译：稀疏矩阵的模式识别及SVE自动向量化、基于MLIR的算子融合优化
- ★ 鲲鹏+昇腾面向领域定制化：混合精度中间基础设施、非结构网格SVE自动向量化及eDSL-MLIR构建

2、编译基础创新：

- 基于MLIR的HPC/AI共性编译器内核底座
- ★ 编译器服务化(CaaS)
- 编译器自动生成技术
- ★ 超节点（异构节点）编程编译技术

3、场景化编译创新：

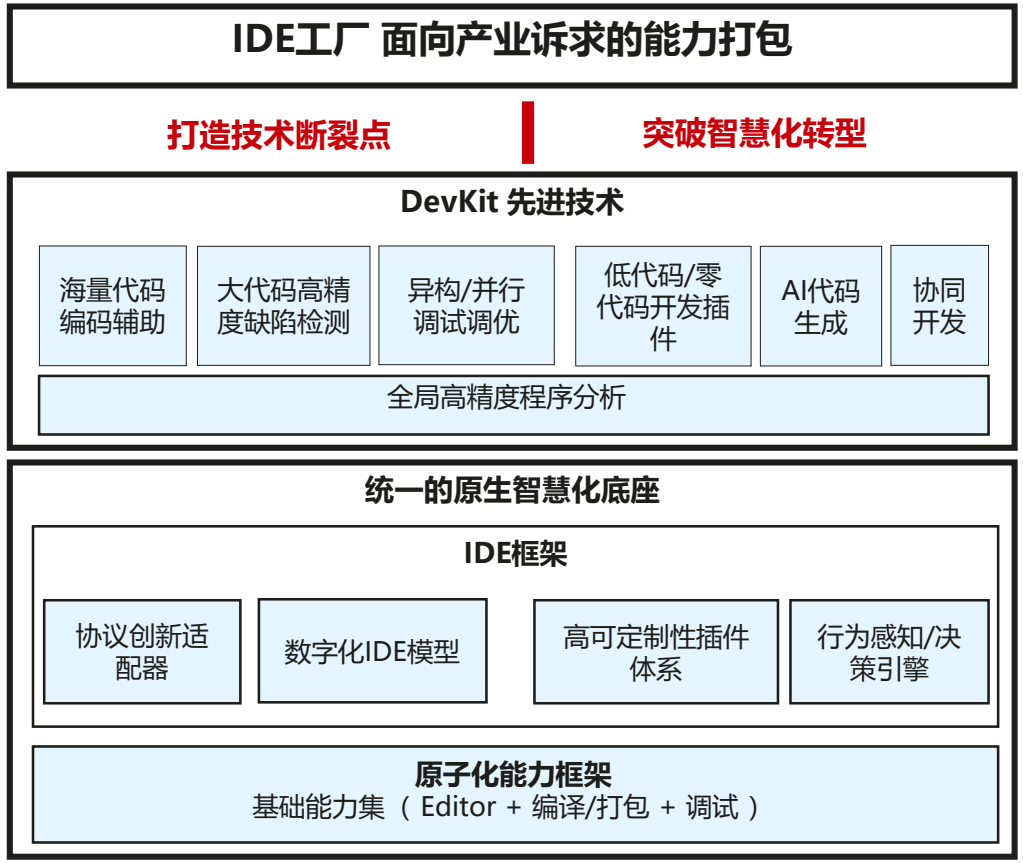
- ★ 针对MySQL等数据库场景，利用PGO+LTO+BOLT的联合优化；
- 利用硬件相关的Roofline Model和编译器的联合分析优化编译过程；
- X86-ARM跨架构二进制翻译翻译
- ARM嵌入式场景下降低CodeSize的编译技术

★ 4、软硬件协同：

- 面向GNN、HPC非结构网格等领域应用中的不规则稀疏访存软硬协同设计；
- 低功耗、高性能的高并发多核/众核架构

HUAWEI

软件IDE：构建程序分析、底座两大根技术，突破IDE智慧化转型，助力产业商业成功



核心创新：先进技术+智慧化底座，构建有竞争力的IDE平台

创新技术：

- **AI代码生成：**通过构建领域AI模型+程序理解增强，代码推荐精准度超过业界竞品
- **海量代码编码辅助工具：**通过高精度大规模程序分析能力，实现海量代码高效开发。

根技术：

- **全局高精度程序分析：****突破百万行级高精度程序分析技术**，为缺陷检查、自动修复、AI辅助编程、调优、协议创新等先进性技术提供基础
- **原生智慧化底座：原生智慧化+原子化能力框架，****打造全方位行为感知、高可定制性化IDE**，成为公司IDE智慧化转型的首选底座

全流程智慧化开发技术创新挑战

★ 高优先

问题和挑战

IDE先进技术插件：

- 探索IDE先进性技术（比如：AI代码生成、智能调试调优等），构建尖端竞争力。

智慧化引擎技术：

- 包含知识智能推荐引擎、混合语言全程序分析，打造全方位感知、高效反馈的IDE智慧化引擎。

IDE基座及通用化服务：

- 构建IDE基座能力（比如：协同开发）；针对IDE通用化服务进行探索（比如：LSP、DAP、PAP等相关技术）。

关键技术创新方向

1、IDE先进技术插件：

- 智能调试调优插件：调试过程中，自动断点推荐、自动修复推荐能力
- 智能调试调优插件：静态分析结合动态profiling自动识别程序性能瓶颈

★2、智慧化引擎技术：

- AI代码生成：全面深入理解程序，支撑AI代码推荐准确率突破60%
- AI代码生成：代码大模型实时反馈学习探索
- 全流程智慧化软件开发体验：不同语言的代码相似性检测
- 全流程智慧化软件开发体验：编程时代码知识自动推荐
- 混合语言全程序分析： C/C++语言多组件快速构建

3、IDE基座及通用化服务：

- IDE 底座框架： 协同和分布式开发（算法）

Thank you.

把数字世界带入每个人、每个家庭、
每个组织，构建万物互联的智能世界。

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI