

当 ABCoder 遇上 Deep Research

演讲人：尹旭然

2025.9.20



目录 | Contents

Part 01
背景

Part 02
解法

Part 03
Demo 展示

Part 04
落地与展望

01

背景

传统 Research 方式面临的问题

LLM 在复杂场景挑战

找出 xx 项目所有对外暴露的接口及其详情，包括包括
请求参数、响应参数、功能描述、核心逻辑实现



文件目录

代码规模20w行, 944 files

biz/app/application.go:118

```
func (a Application) RegisterHandlers() {  
    value := reflect.ValueOf(a)  
    for i := 0; i < value.NumField(); i++ {  
        field := value.Field(i)  
        if field.Type().Implements(reflect.TypeOf((*RouteRegister)(nil)).Elem()) {  
            httpHandler := field.Interface().(RouteRegister)  
            httpHandler.Register(a.ApiRouteGroup)  
        }  
    }  
}
```

biz/handler/acl.go:60

```
func (h *ACLHandler) Register(r *route.RouterGroup) {  
    groupHandler := r.Group("acl")  
  
    groupHandler.GET("strong_auth/can_use", h.CanUseStrongAuth)  
    groupHandler.GET("strong_auth/status", h.GetStrongAuthStatus)  
    ...  
}
```

Deep Research 如何做 --- 原料

关联性不强、搜索内容多

语义化搜索



文字匹配



Deep Research 问题 --- 流程

没办法精确找到需要的代码片段范围

上下文过多



无用信息过多



Deep Research --- 示例

User: 找到项目中所有的 API

LLM: 调用工具 search

Tool: `grep "register" |" gin" |" http" |" router" |" path" |" endpoint"`

LLM: 读取文件 application.go

Tool: Read application.go 1-200 lines (total 2000 lines)

LLM: 该项目中所有对外暴露的 API 为 ...



项目**过大**? 注册路由的地方**过多**? 同名的干扰**太多**?

02

解法

Deep Code Research 如何解决

原料解法 --- ABCoder



nodes id: {repo_name} ? {package_name} # {func / variable / struct.method }

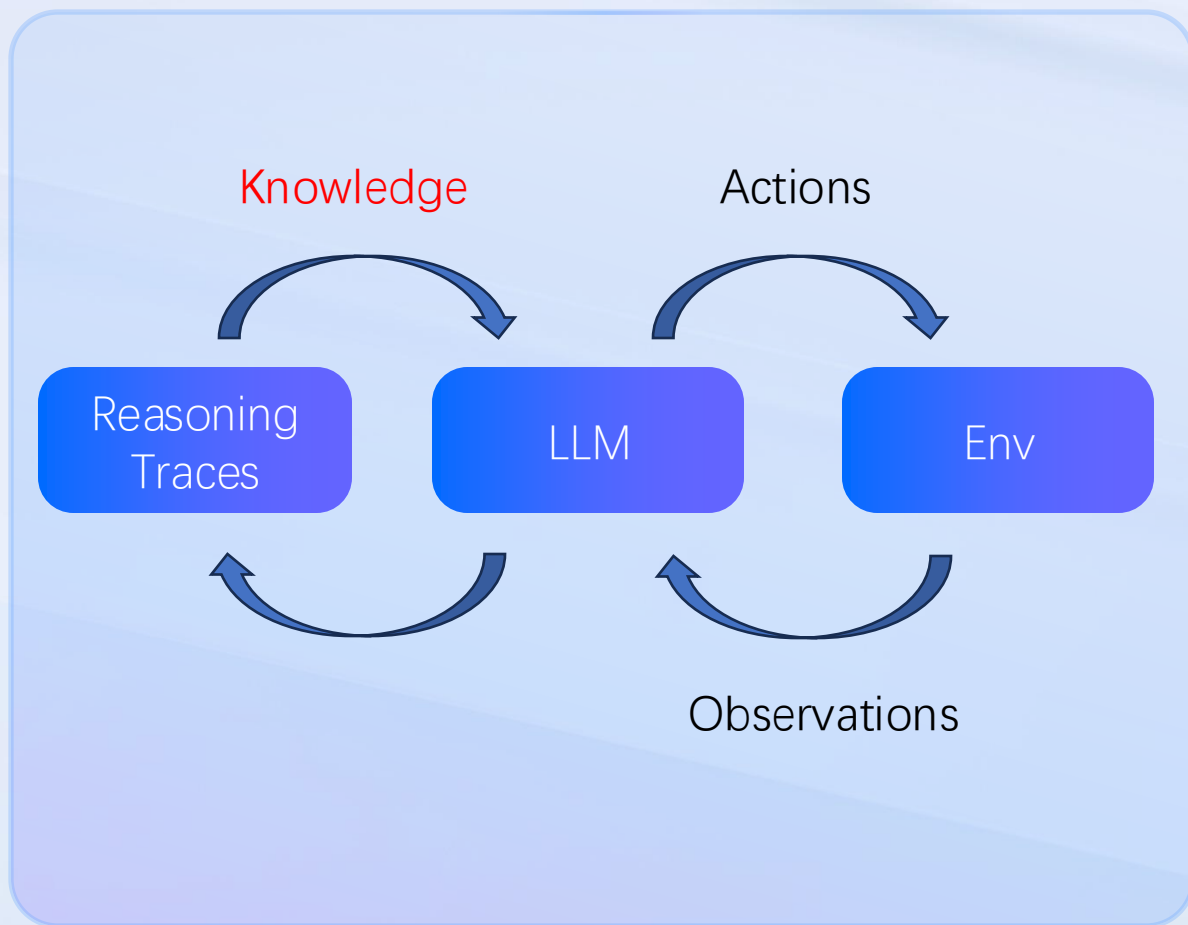
原料解法 --- ABCoder



nodes id: {repo_name} ? {package_name} # {func / variable / struct.method }

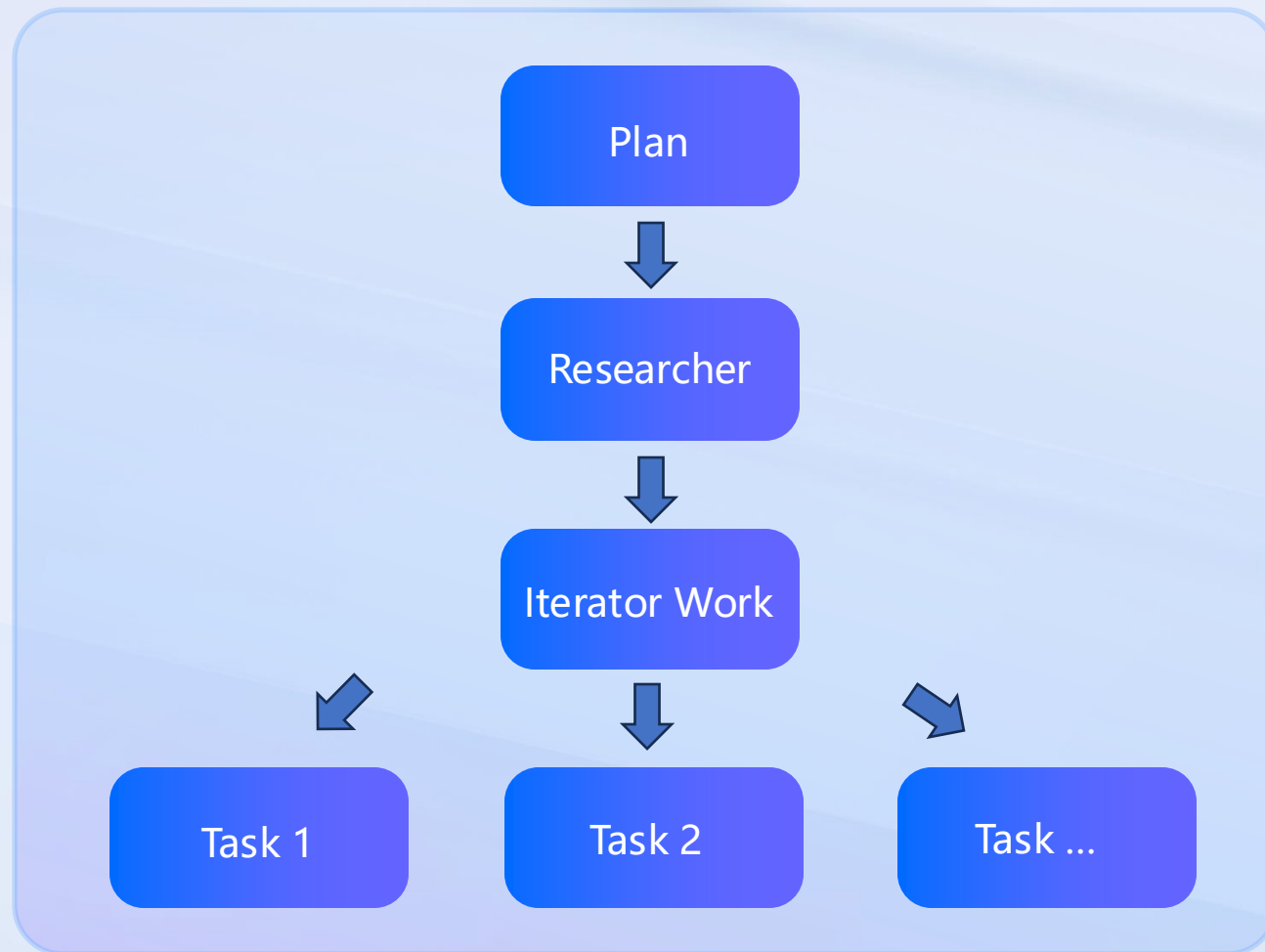
流程解法 --- Deep Code Research

只保留有用的内容



流程解法 --- Deep Code Research

避免“所有”，使用“每一个”



Deep Code Research --- 示例

Deep Research

User: 找到项目中所有的 API

LLM: 调用工具 search

Tool: **grep** "router" |" http "|" router" |" path"

LLM: 读取文件 router.go

Tool: Read router.go **1-200** (total 2000 lines)

LLM: 该项目中所有对外暴露的 API 为 ...

ABCoder + Deep Code Research

User: 找到项目中所有的 API

LLM: 调用工具 get_nodes_detail 获取 main 函数信息

Tool: **get_nodes_detail**, `reponame?packagename#main`

Tool Res: {nodes:xxx, codes: xxx, reference: xxx, dependencies:xxx,}

LLM: 我已经获取了 main 函数的信息, 下一步需要检查 Register 函数

Tool: get_nodes_detail, `reponame?packagename#Register`

LLM: 该项目中所有对外暴露的 API 为 ...

03

Demo 演示

查找工厂类实现

```
rr, err := registry.Execute(ctx, "rpc", "app", req)

func Execute(ctx context.Context, callType, key string, req interface{}) (interface{}, error) {
    if callType == "http" {
        return httpRegistry.Execute(ctx, key, req)
    } else {
        return rpcRegistry.Execute(ctx, key, req)
    }
}

func (r *Registry) Execute(ctx context.Context, key string, req interface{}) (interface{}, error) {
    handler, err := r.Get(key)
    if err != nil {
        return nil, err
    }

    return handler.Handle(ctx, req)
}
```

查找工厂类实现

同样使用 doubao-seed-1.6

```
claude
/help for help, /status for your current setup
cwd: /Users/zekin/code/poc/kitex_b

Overrides (via env):

• API timeout: 600000ms
• API Base URL: http://127.0.0.1:3456

What's new:
• Improve thinking mode display with enhanced visual effects
• Type /t to temporarily disable thinking mode in your prompt
• Improve path validation for glob and grep tools
• Show condensed output for post-tool hooks to reduce visual clutter
• Fix visual feedback when loading state completes

> 我查找 handler.go 中 ItemServiceImpl.GetItem 这个函数通过 Execute 调用了哪个 Handler, 你需要找到这个 Handler 具体的注册代码
```

```
* Handler 查找

Warn 2025-09-16 16:27:14,754 v1(7) util.go:108 10.3.191.55 - - default - 0 [KITE X] config file conf/kitex_remote_config.json is not writable, will try next candidate: err=<Error: open conf/kitex_remote_config.json: no such file or directory >
Info 2025-09-16 16:27:14,764 v1(7) manager_wrapper.go:224 10.3.191.55 - - default - 0 [KITEX] load config and notify succeeded: service=-/default/0@BYTED-CP-1-caller
Info 2025-09-16 16:27:14,784 v1(7) manager_wrapper.go:224 10.3.191.55 - - default - 0 [KITEX] load config and notify succeeded: service=overpass.repo.info//1@BYTED-CP-1-callee
Warn 2025-09-16 16:27:14,787 v1(7) client.go:170 10.3.191.55 - - default - 0 _env=prod _env_type=- [TCC] ZTI Token Get Error, error = the token is empty
Warn 2025-09-16 16:27:14,813 v1(7) client.go:170 10.3.191.55 - - default - 0 _env=prod _env_type=- [TCC] ZTI Token Get Error, error = the token is empty
Debug 2025-09-16 16:27:14,844 v1(7) mysql.go:177 10.3.191.55 - - default - 0 connected database with primary dsn :@sd(toutiao.mysql.hkong_write)/hkong?charset=utf8mb4&interpolateParams=true&loc=Local&parseTime=True&readTimeout=5s&timeout=5s&use_gdpr_auth=true&writeTimeout=5s
Info 2025-09-16 16:27:14,852 v1(7) mysql.go:36 10.3.191.55 - - default - 0 _env=prod _env_type=- init mysql for toutiao.mysql.hkong success
帮我查找 code.byted.org/middleware/kitex_b@b4f8ffe?code.byted.org/middleware/kitex_b#ItemServiceImpl.GetItem 这个函数通过 Execute 调用了哪个 Handler, 你需要找到这个 Handler 具体的注册代码
<end>[Stats] 2025-09-16 16:27:34.765 failed to collect cgroup info: stats is nil
```

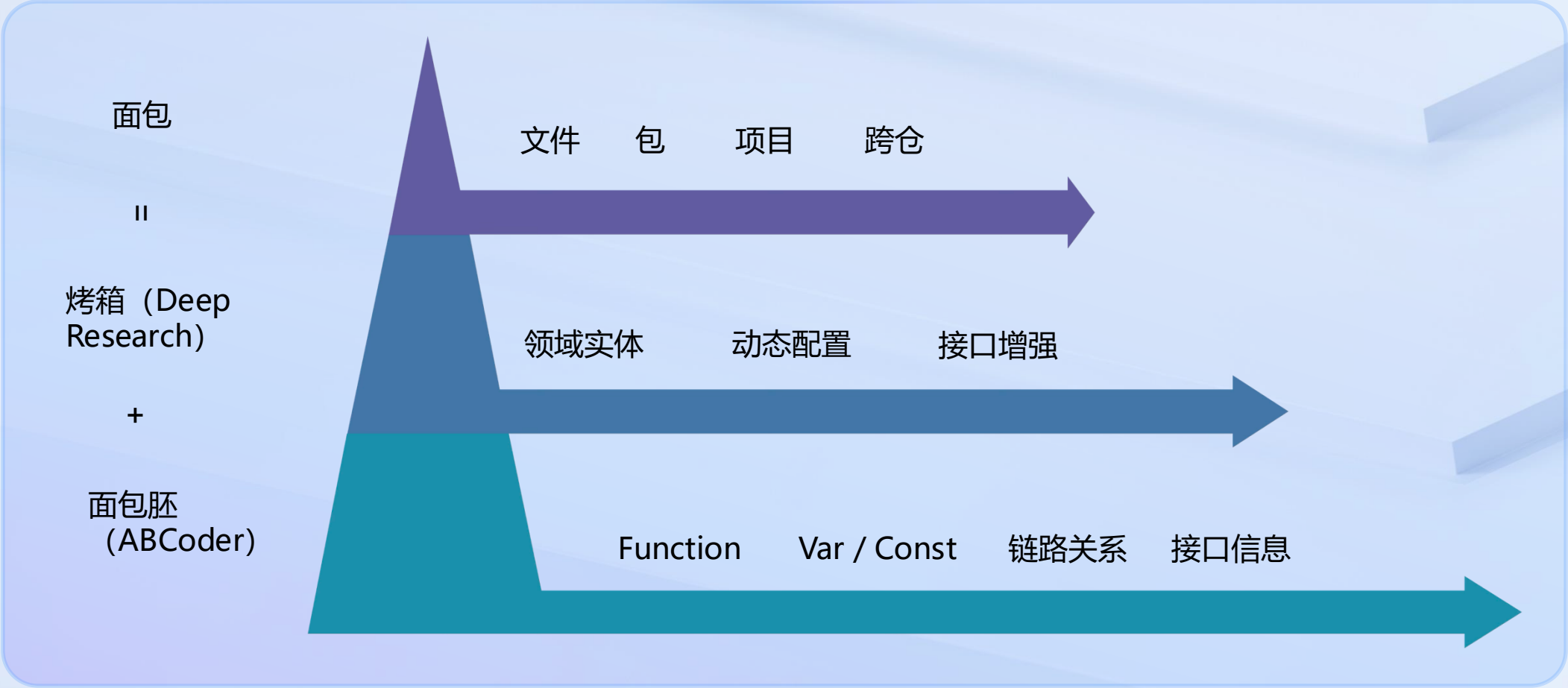
04

落地与展望

当 ABCoder 遇上 Deep Research

落地

知识库多维度增强



落地

01



火焰图分析

根据火焰图和源代码，找到可以优化的点

02



稳定性分析

找到源代码中关于稳定性相关的配置，并分析配置的是否合理

03



Code CR

分析代码变更行数前后依赖的函数，为 Code CR 提供更多的信息。

展望

从 AI Coding 到 AI Development 之路

01



技术价值

结构化表征 + 工具
链创新

02



业务价值

提效 + 降本 + 生态
共建

Q & A

项目地址: <https://github.com/cloudwego/abcoder>

THANKS

