

关键技术清单

版本：2012/12/3（修订 2）

技术：任务组件模型（TCM）

概要

任务组件模型是一种抽象架构理念。它描述了一种推动软件开发组件化的技术框架。

适用场合

任务组件模型适用于任何以丰富算法为核心的软件。

要求

应用任务组件模型的软件模块，需要描述“不需要注入人工操作”的自动化功能。

即，模块中没有直接进行用户交互的代码。

凡是符合此要求的代码都可以按照任务组件模型的组件开发规范改造封装成任务组件。

参见《TCM 组件开发规范》。

组成

用于 C++ 的 TCM 运行时开发库

用于 C# 的 TCM 运行时程序集

描述文档、技术文档

辅助开发工具

功能

将算法与用户界面解耦，有效简化自动化组件的开发，明确开发职责。

将平台开发与组件开发解耦，有效避免组件开发对平台开发产生影响。

将平台执行与组件执行解耦，有效避免组件的不稳定性的传播。

异步调用组件，使用多线程方式隔离用户界面与业务的执行。

支持远程执行与控制（参见 TCM RPC）。

优势

减少组件开发人员的工作量，开发人员不必再编写组件通信代码。

减少平台开发人员的工作量，开发人员不必再编写组件控制代码。

规范组件与平台开发，易于平台扩展和组件移植。

支持高性能、线程安全和丰富的组件间通信，支持状态、任意参数的传递。

支持多种开发语言，开放一套易用的 C 风格接口，可被许多语言调用。

支持多种操作系统（待研发）。

易于集成，已有的平台和组件不需要太大的改造即可应用任务组件模型。

国际化，支持任意语言与地区

不足

仅支持自动化过程的组件化

目前可用组件较少

技术：任务组件流程化（TCM Flow）

概要

任务组件流程化是流程控制与任务组件模型结合后产生的技术。

适用场合

应用了任务组件模型的项目。

要求

有一批可用的任务组件，这些组件应当具备一定的业务联系。

组成

- 用于 C# 的调度执行引擎
- 用于 C++ 的调度执行引擎
- 可视化流程（模型）集成开发环境

功能

- 设计开发、调试由任务组件构成的流程（模型）。
- 可以把流程发布成可执行文件。
- 可以导入导出多种脚本。
- 强大而智能的调度引擎，
- 支持远程调度与控制

优势

- 易用性。全可视化的流程建模体验，对用户没有技术基础要求。
- 可以进行低成本、高效的自动化业务产品定制。
- 高度的可扩展能力，这是由于采用 TCM 所具备的。

不足

（仍在开发中，参见《TCM 与流程化的版本计划》）

同行业比较

	TCM Flow	Erdas	ArcGIS
可视化	✓	✓	✓
组件丰富	✗	✓	✓
逻辑丰富	✓	✗	✗
组件可扩展	✓	✗	✓
可管理模型	✓	✓	✓
可导出脚本	✗	✓	✓
可发布执行体	✓	✗	✗
远程调度	✓	✗	✓
跨语言	✓	✓	✓
易用性	✓	✓	✓
灵活性	✓	✗	✗

技术：并行计算

概要

利用多核 CPU、GPU 的硬件优势提高算法执行效率的技术

适用场合

算法具有较高的时间复杂度。

算法处理数据量较大（GB-TB 级别）。

算法中存在 IO、计算交替。

以上任意一个场合适用即可。

要求

具有多核 CPU 或多核 GPU 设备。

过程、数据或代码具有可分性。

每个核心的处理过程互不干涉。

功能

在指定的终端，将任务数据分块，分发到 CPU 和 GPU 的多个核心并行执行，以获得时间效率的提升。这主要使用 OpenCL、CUDA 等技术实现。

进行代码级别的多核并行，获得关键算法的时间效率的提升。这主要使用 PPL、AMP 等技术实现。

将计算与部分 IO 时间重叠，获得总体时间效率的提升。

优势

并行方法丰富，应用场景更全面

效率提升显著

随硬件升级，可以获得效率提升

不足

实现复杂，需要针对具体问题具体分析

技术：分布式计算

概要

利用多个的终端构成的分布式网络执行大规模数据任务的技术

适用场合

数据量极大（不低于 PB 级别），具备多台终端可以构成网络。

简单升级硬件获得的效率提升已不显著。

要求

具有较多的计算终端可以构成网络。

过程、数据或代码具有可分性。

每个核心的处理过程互不干涉。

功能

可以切分、分发任务

可以调度、控制节点

具备容错、容灾能力

可与流程化结合实现业务流

优势

随节点增加，效率提升显著

可构成数据中心，结合虚拟化技术可以扩展成云

不足

受制于网络环境

任务规模不足则效率提升不显著

网络规模不足则效率提升不显著

技术：高性能应用程序框架（HPA）

概要

操作系统层面优化计算、IO 的一类技术构成的开发框架

适用场合

一般软件开发

要求

可能要求特定硬件平台或系统要求

功能

内存池管理，通过托管内存优化内存使用效率

线程池管理，通过托管线程优化线程调度效率

指令集优化，利用 CPU 的高级指令集直接获得性能提升

优势

效率提升显著

粒度小，可以随时使用