**微软开源rDSN分布式系统开发框架**

今天，由微软亚洲研究院系统组开发的分布式系统开发框架——Robust Distributed System Nucleus（rDSN）通过[GitHub](https://github.com/microsoft) 平台开源。rDSN旨在为广大基于分布式系统应用和框架的开发者，研究人员，以及学生提供一个开放式的框架，用于快速搭建高性能和高鲁棒的分布式系统。

rDSN首先利用和扩展了已有的RPC框架生成工具(如Google Protocol Buffer, Apache Thrift)来保证**可编程性**；和很多高并发系统一样(如nginx, Node.js, lighttpd)，采用事件驱动架构来达到对系统资源的充分利用来实现**高性能**。rDSN的不同之处在于，它同时还尝试解决分布式系统开发和运维的整个生命周期里的其他一些和鲁棒性相关的问题，比如**测试**，**调试**，**部署**，**自动扩展**，和**高可用**等等。开发者一开始通常对这些问题缺乏整体规划(如缺乏资源)，但是到后期它们会暴露并造成比较大的损失(如几个小时甚至几天服务不可用)，而开发者在系统代码基本完成后(Post-Programming)修复这些问题也经常事倍功半。rDSN因此提供了一个统一的开发框架，在尽量不增加开发者负担的前提下，帮助开发者**系统和（半）自动地解决**这些问题。

开发rDSN这个框架的想法来源于我们过去基于已有分布式系统的各种自动化工作，包括自动测试，调试，优化，运维，复制(Replication)，组合(Composition)等。在这其中我们遇到了很多困难，很多都是由于我们的工具没办法处理所有已有代码的状况 - 导致没法真正投入生产使用。我们尝试总结这些工具的共同需求，结果发现我们需要回答一个很基本的问题: **分布式系统带来什么样的困难，如何(半)透明地解决这些困难**。相比单机单线程系统，我们认为分布式系统的困难之处在于它引入了各种各样的系统复杂度，比如并发(Concurrency)，异步(Asynchony)，网络延迟(Network Delay), 丢包(Message Lost)，宕机(Machine Crash), 以及其它形式各异的故障(Fault)等。这些不确定性和它们的叠加决定了开发一个鲁棒的分布式系统相比单机单线程系统要困难许多，开发如上这些自动工具则更加困难。rDSN于是引入了两个方法来解决这个问题。

首先在单机方面，通过仔细定义编程模型和执行模型，rDSN能够**观察和操纵这些系统不确定性**。rDSN提供了两个开发API: 服务API(Service API)和工具API(Tool API)，前者用开发基于事件驱动的高并发应用，而后者则在事件驱动模型的基础上暴露各种系统不确定性来开发各种工具。比如，为了解决分布式系统调试超时的问题，我们有一个工具来模拟时间。另外一个工具模拟各种各样的调度决策和系统错误，来测试系统在各种实际部署环境中的鲁棒性。当错误发生时，它能够实施相同的不确定性决策，重播发生的错误，来帮助开发者更快地找到问题的根源。再比如，我们的自动复制工具(Replication)录制了在一台机器上发生的不确定性并通过复制协议把不确定性在另外几台机器上重播，保证了系统状态的一致性和发生故障时的高可用性。

第二在分布式方面，rDSN定义了一个一致的系统架构来**简化系统的复杂度**。在rDSN中，分布式系统由一组自包含的基础服务，以及在它们之上的用来处理端到端系统输入(比如用户请求)的一组工作流组成。开发者使用IDL(Interface Definition Language, 比如Google Protocol Buffer) 来描述基础服务的接口，和另外一个类SQL的语言来描述如何调用这些接口组成工作流。rDSN提供了工具来自动翻译这两个语言并生成基于事件的代码框架。开发者需要简单地把具体的应用逻辑写成一个个简单的事件，并使用rDSN提供的服务API来实现所有跨事件的以及不确定的操作，以保证相关工具能够正常工作。rDSN另外还提供了一组分布式系统框架，比如load blancer, partition manager, replication, 和workflow controller，来支持上层应用的分布式执行。这些方法使得rDSN对上层应用在跨事件，线程，和机器的所有依赖都有足够的了解和控制，这使得我们有可能实现可靠的和高端的开发和运维工具/框架/策略，来进一步保障应用的鲁棒性。

尽管提供了预定义的框架和编程模型，rDSN尽量和现有的常用分布式系统编程框架保持一致。我们移植了5个比较流行的库，发现代码的改动都在200~500行之间。另外，rDSN采用了"微内核“的架构，使得整个系统非常开发。比如， 所有基于工具API开发的工具，都可以和基于服务API开发的应用**无缝集成**。这带来了几方面的好处：一是开发者可以很方便地把自己开发的工具共享给社区，帮助其他应用变得更加鲁棒和高效。二是由于在rDSN中，实际部署使用的运行时库(比如网络，文件，锁等)也是一个特殊的工具，开发者因此可以很容易地集成他们自己喜欢的库或者在新的硬件平台上开发新的运行时插件(如基于RDMA的网络库)。rDSN目前已经包含了一组开发工具，一些运行时工具，和一些分布式系统框架。

此外，由于早期rDSN版本已经在Bing内成功使用和上线，现在的新版本也在产品组反馈的基础上进一步做了很多改进。相信一定会为正在开发，学习和研究分布式系统的人员带来很大的便利！