**Microsoft Makes rDSN Open Source**

Developed by System Research Group of [Microsoft Research Asia](http://www.msra.cn/zh-cn/default.aspx), Robust Distributed System Nucleus (rDSN), is now open source on [GitHub.com](https://github.com/). rDSN is an open framework for developers, students, and researchers to quickly build and manage high-performance and robust distributed systems, and the latter is critical for the success of many emerging technologies today such as cloud computing, big data, and IoT (Internet of Things).

The idea of this framework arises during the team's past efforts to (semi-) automatically test, debug, optimize, monitor, scale, replicate, compose, and reason about distributed systems. A lot of challenges were encountered in those projects, and most are due to the fact that the initial development process had not consider these goals, resulting later maintenance difficulties.

rDSN provides a coherent framework from which developers build their systems with minor adjustment from traditional development styles. Code built on atop of rDSN conforms to certain principles, and can be upgraded later with little or no cost for achieving the aforementioned goals. An early version of rDSN has been used in Bing for building a distributed data service, and the system has been online and running well. Based on feedbacks from the production teams, rDSN is improved and now made public with an open source license. The goal is to benefit the community especially developers, students, and researchers who are working on distributed systems in various ways.

**For developers**, rDSN enhances development and management experience for system programmability, performance, and robustness. At its simplest form, rDSN can be used as an enhanced RPC library compatible to many others (e.g., Apache Thrift), or a task library where event-driven programming is employed for high throughput. Developers can also configure rDSN into “test” mode, which systematically tests the systems against various failures and scheduling decisions, exposing possible bugs early. Once a bug is exposed, you can switch to a “debug” mode to reproduce it, with all nodes' state in a same process and debug without worrying about false timeouts. When it is online, rDSN provides automatic flow tracing and performance monitoring. If you are not satisfied with the default libraries in rDSN and want to use your own (e.g., logging or networking library), rDSN is open and can be easily modified. Even further, when you need to scale your service and make it reliable under node failure, rDSN can replicate the service with minor further development. In summary, rDSN provides and allows tools/frameworks to be seamlessly integrated with your system, which greatly improves the efficiency of system development and management.

**For students**, rDSN provides a platform where you can easily simplify, understand and manipulate a distributed system. When learning distributed protocols, you can easily implement one atop of rDSN, and test it on its simulator. The simulator can abstract away many practical complexities initially, and you can add them back gradually to evolve your protocol, such as from single-thread to multiple-thread, from constant message delay to variant ones, even with message lost. To understand the running protocol, rDSN provides flow tracing and generates a so-called “event matrix” which records the invocation count among different events, revealing the dependencies with weight inside the system.

**Researchers** usually want to find and build something common to many distributed systems, such as runtime policies and diagnosis tools. rDSN provides a dedicated Tool API for that purpose. The API provides virtualization of all low level components, and exposes all non-deterministic behaviors from the upper applications at the event granularity. With this API, it is much easier to build reliable and effective runtime tools and/or policies. The current release contains a small set of examples. Even better, rDSN ensures that those tools can always be seamlessly integrated with the upper level applications - a big bonus for the research work to make real impact.

With all these possible benefits, it is hoped that the community can together build better distributed systems easily, by adopting rDSN, and contributing back to help others. Visit the project now at <https://github.com/Microsoft/rDSN>.