**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 provides an open framework for developers, students, and researchers to quickly build, understand, and analyze high-performance and robust distributed systems.

rDSN leverages and enhances the existing tools such as Google Protocol Buffer and Apache Thrift for programming agility; it adopts the event-based architecture to achieve high performance. The difference of rDSN, is that it also pays a lot of attention to robustness, for which it provides systematic supports spanning the whole life-cycle of a distributed system, by providing, enabling, and integrating better programming models, distributed service frameworks, as well as development and operation tools, into a single coherent platform. The current version of rDSN is in C++, and can run on multiple platforms include Linux, OS X, and Windows. By open sourcing this framework, we think we may help the developers, students, and researchers who are working on distributed systems.

**For developers** who writes distributed applications and/or infrastructures, rDSN can greatly enhance the development and operation experience, and at the same time improve the system performance, robustness, reliability etc. with much less cost! For example, by simply configuring the framework into “test” mode, a distributed system atop of rDSN can be automatically tested against many different combinations of system scheduling and various faults, exposing early the possible bugs in your system without really putting them into deployment environment. Once there is a bug exposed, you can switch to a “debug” mode, and rDSN will reproduce the bug for you. When it is online, rDSN provides automatic flow tracing and performance monitoring. Even further, when you need to make your service more reliable, you can use our code generation tool and rDSN replicates your service with very minor further development cost. rDSN provides many tools that can be seamlessly integrated with your distributed systems, and they are growing!

**For students** who study distributed systems, rDSN provides a platform where you can easily understand and manipulate a distributed system. For instance, rDSN has a pre-defined execution model following the event-driven architecture, and the computation of an application in rDSN is decomposed into many named events. rDSN generates a so-called “event matrix” which records the invocation count among named events for a given system instance. This gives you the high level information about how the given system is organized and what are the flows there. As another example, rDSN follows the “micro-kernel” architecture, where all the low level components can be easily replaced, with which you can try to change one each time and observe to understand what its contribution in this framework. When you are learning some distributed protocols, you can easily implement one atop of rDSN, and test it on its simulator. The simulator can abstract away many practical difficulties initially, and you can add them back gradually to improve your protocol, such as from single-thread to multiple-thread, from constant message delay to variant ones, even with message lost. And we think there are a lot more potentials.

**For researchers** who usually want to find and build something common to many distributed systems, such as runtime policies and diagnosis tools, rDSN is just a perfect platform as it provides a dedicated Tool API for that purpose! The API ensures that all non-deterministic behaviors from the upper applications are exposed, and can be easily captured as well as controlled by the underlying tools built with this API. It also introduces hooks on all asynchronous flow points and capability of attaching the state to the events and messages, enabling many interesting scenarios such as taint analysis of the events. With this API, we actually have already built a handful set of development tools and runtime policies. Even better, rDSN ensures that the tools can always be seamlessly integrated with the upper applications. The research results therefore can usually be taken into production adoption without further effort - a big bonus for the research work.

With all these possible benefit, we believe we can together build better distributed systems easily, by not only adopting rDSN, but also contributing back whatever you can build with rDSN to help the others. The original version of rDSN has already been successfully implemented and used inside Bing. With feedbacks from product teams, the current updated version improves a lot. We hope it brings great value to those developers, students, and researchers who are working on distributed systems. Visit our project now at <https://github.com/Microsoft/rDSN>.