System Complexity and Interdependency

系统复杂性和相互依赖性

Modern online financial systems are some of the most complex systems

in the world today. They process enormous transaction loads

at incredible speeds with high integrity. All of these systems are

interlinked with many other systems in many different organizations,

creating a massively distributed “system of systems” problem

of extreme scale and complexity, with multiple potential points of

failure.

现代的在线金融系统是当今世界最复杂的系统之一。它们以难以置信的速度和高度的完整性处理巨大的事务负载。所有这些系统都是许多不同组织的其它系统互相连通，创建出一个大规模分布的“系统体系”问题，具有极大的规模和复杂性，具有多个潜在的失效点。

While these systems might share common protocols, they were not

necessarily all designed to work with each other. All of these systems

are constantly being changed by different people for different reasons

at different times, and they are rarely tested all together. Failures

can and do happen anywhere along this chain of systems, and

they cascade quickly, taking other systems down as load shifts or as

systems try to handle errors and fail themselves.

虽然这些系统可能使用公共协议，但它们不都是为了彼此合作而设计的。所有这些系统不断地在不同的时间，因为不同的原因被不同的人所修改，并且很少在一起测试。在这个系统链的任何地方都可能发生失败，而且也的确在发生，并且快速蔓延，因为负载转移或者当系统尝试处理错误时导致失败从而将系统拖垮。

It doesn’t matter that all of these systems are designed to handle

something going wrong: hardware or network failures, software failures,

human error. Catastrophic failures—the embarrassing accidents

and outages that make the news—aren’t caused by only one

thing going wrong, one problem or one mistake. They are caused by

a chain of events, mostly minor errors and things that “couldn’t possibly

happen.”4 Something fails. Then a fail-safe fails. Then the process

to handle the failure of a fail-safe fails. This causes problems

with downstream systems, which cascade; systems collapse, eventually

leading to a meltdown.

所有的系统都是为了处理出现的问题：硬件或网络故障、软件故障、人为错误，但这并没有关系。新闻报道中令人尴尬的事故和宕机，产生这些灾难性的故障的原因并不仅仅是一件事出错，一个问题或一个错误。它们是由一连串的事件导致的，主要是小错误和“不可能发生的事情”发生了。一个故障发生了。然后一个故障保护装置失效了。然后处理故障保护装置失效的过程也失效了。这会引起与之关联的下游系统的问题；整个系统系统开始崩塌并最终导致崩溃。

Completing a financial transaction such as a trade on a stock

exchange involves multiple different systems, with multiple network

hops and protocol translations. Financial transactions are also often

closely interlinked: for example, where an investor needs to sell one

or more stocks before buying something else, or cancel an order

before placing a new one; or when executing a portfolio trade

involving a basket of stocks, or simultaneously buying or selling

stocks and options or futures in a multi-leg combination across different

trading venues.

完成一项金融交易，如在股票交易所完成一笔交易，涉及多个不同的系统、多个网络

跳数和协议转换。金融交易也经常紧密互联：例如，投资者需要在卖出一支或者多支股票之前，需要买入其它标的物，或者在发出一个新订单之前需要取消一个订单；或者在执行投资组合交易时涉及一篮子股票，或者进行跨市场的股票、期权或期货的多腿组合交易。

Failures in any of the order management, order routing, execution

management, trade matching, trade reporting, risk management,

clearing, or settlement systems involved, or the high-speed networking

infrastructure that connects all of these systems together, can

make the job of reconciling investment positions and unrolling

transactions a nightmare.

任何涉及订单管理、订单路由、执行管理、成交撮合、成交报告、风险管理、清算和结算等系统，或者将所有这些系统连接在一起的高速网络基础设施的故障，将导致调整投资头寸或者展开交易的工作成为一场噩梦。

Troubleshooting can be almost impossible when something goes

wrong, with thousands of transactions in flight between hundreds of

different systems in different organizations at any point in time,

each of them handling failures in different ways. There can be many

different versions of the truth, all of which will claim to be correct.

Closely synchronized timestamps and sequence accounting are

relied on to identify gaps and replay problems and duplicate messages—

the financial markets spend millions of dollars per year just

trying to keep all of their computer clocks in sync, and millions

more on testing and on reporting to prove that transactions are processed

correctly. But this isn’t always enough when a major accident

occurs.

4 For more on how this happens, read Dr. Richard Cook’s paper, “How Complex Systems

Fail”.

当发生故障时，几乎不可能进行排障。成千上万的交易在数百个不同组织的不同系统中任一节点和任何时间发生，各个组织处理故障的方式也都各不相同。可能有很多不同版本的所谓真相，每个都声称是自己是正确的。依靠紧密同步的时间戳和序列号，可以识别差距、重现问题和重复消息-金融市场每年花费数百万美元仅仅是为了保持他们所有的计算机时钟同步，数目更多的资金则被用于测试和报告以证明交易被正确地处理。但当发生重大故障时，这并不总是足够的。

4 关于这种情况的更多信息，请阅读Richard Cook博士的论文“多复杂的系统”

“失败”。

Nobody in the financial markets wants to “embrace failure” or “celebrate

failure.” They want to confront failure: to understand it, anticipate

it, contain it; to do whatever they can to prevent it; and to minimize

the risks and costs of failure.

金融市场上没有人愿意“拥抱失败”或“庆祝失败”。他们想对抗失败：理解它，预测它，控制它；尽其所能阻止它；并且最小化失败的风险和成本。

With so many systems involved and so many variables changing

constantly (and so many variables that aren’t known between systems),

exhaustive testing isn’t achievable. And without exhaustive

testing, there’s no way to be sure that everything will work together

when changes are made, or to understand what could go wrong

before something does go wrong.

因为涉及的系统太多，变量也在不断地变化（以及系统之间不知道的那么多变量），详尽的测试是无法完成的。而没有详尽的测试，当发生变化时，就没有办法确保所有的东西都能一起工作，或者在问题出现之前理解可能发生的错误。

We’ll look at the problems of testing financial systems—and how to

overcome these problems—in more detail later in this book.

我们将研究测试金融系统的问题以及如何解决这些问题。在本书后面的部分中更详细地进行论述。