Section 4: Week 8: Enhance a Distributed System Architecture

Nate Bachmeier

TIM-8120: Distributed Systems

November 10th, 2019

North Central University

# Enhance a Distributed System Architecture

Contoso Retail and Manufacturing is similar to many large enterprises in their need to manage and support heterogeneous networks that span multiple datacenters and branch offices. Their workloads include standard services such as authentication, name resolution, and distributed storage system, along with proprietary systems. This topology naturally stears them towards distributed system designs, which come with their own sets of challenges, like balancing availability, reliability, and performance. Many of these challenges are NP-hard, meaning that the precies answers are complex to derive but easy to verify. Therefore they require efficient approximation algorithms or revolutionary changes in approach. Other aspects of the system need architectural patterns that enable Recovery Oriented Programming (ROP)*.*

# What is NP Complexity

If the solution to a decision problem takes *polynomial time* to solve, then it would have a complexity of *P*. These questions would include addition, sorting a list, and performing many procedural transformations. Other problems take *nondeterministic polynomial time* to solve and fall into the category of *NP*. If the solution to an NP question is verifable in polynomial time, then it is *NP-hard*. Examples include hardware verification, multiprocessor scheduling, and even Super Mario Brothers (Mann, 2017) (Aloupis, Demaine, & Guo, 2012). When both the answer and verification take nondeterministic polynomial time, these challenges are said to be *NP-complete*.

According to Aloupis et al., if and only if, a decision problem can be proven to be as tough as an existing NP-hard problem, can it be considered NP-hard. A common strategy is to map the new challenge to the classic *Satisfaction Problem* (SAT). SAT is known to be NP-hard as its “combination of Boolean variables, negations, disjunictions, and conjustions […] cannot be solved more efficiently than exhaustive search (2^n steps) (Mann, 2017, p. 76).” Consider a Mario level (see Figure 1), where the player needs to collect (true) and use (false) power-ups in a precise sequence of events to reach the finish line. A model of these decisions can be perfectly maped to SAT and therefore said with certainty, to solve Mario is at least equally hard to solving SAT.

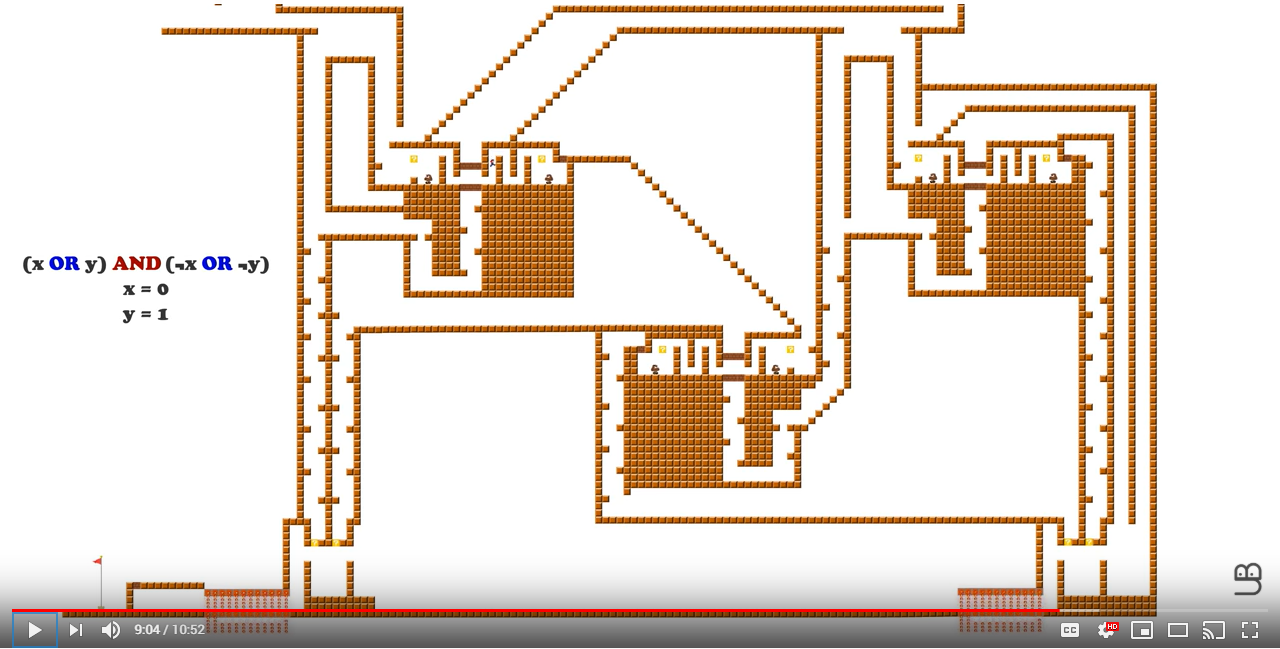


Figure 1 Super Mario as SAT (Undefined Behavior, 2019)

Mann states that there are many misconceptions around the notion of NP-hard and that literature frequently misrepresents the establishment of this assignment. Many scenarios, such as combinatorials, require exponential but finite time making them P. He also highlights that many problems *appear* similar, but can through changes to the model no longer be map precisely. These changes might include additional constraints to define the scope of the solution. Imagine the previous Mario level, where the player can walk through walls or use a ladder (oracle) to skip the maze and directly reach the end. Lastly, he causions against the presumption that NP-hard can only solutions only exist through exhaustive search and heuristics. Many complex problems, such as bin packing, are addressable on a practical basis with a consistent selection strategy.