Defining Bit Stability: A framework for Bit Stability metrics
Micah Beck
University of Tennessee
6/29/12011

Intuitively, bit stability is the property that lies at the basis of all storage and so is fundamental to the long-term preservation of data. However, in order to quantify and measure bit stability, it is necessary to give it a technical definition. This requires us to start with a still more fundamental question: in the context of data preservation, what is a bit?

A bit, or binary digit, is often defined as an abstract numeral, one of two arbitrary choices that, by convention, we name zero and one. Such numerals can then be given concrete expression as states of an observable physical property, usually either matter or electromagnetic particles such as electrons or photons. We think of this state as the modern equivalent of a mark written on paper or chiseled in stone. The bit is embodied as the smallest, most fundamental digital object.

The problem with defining a bit as an observable physical state is it gives rise to a notion of bit stability that is not very useful in the discussion of practical digital preservation. To say that an observable physical state is stable says nothing about what is required in order to observe it. A mark scratched into a metal plate may retain an observable state for a long time, but consider what happens when we attach that plate to a the Voyager spacecraft and set it on a course that will take it out of the solar system. At some point, the mark is no longer observable to any being known to exist, but is in principle observable to a hypothetical alien intelligence, or to future humans who can travel much faster. It may be difficult to say whether the bit so encoded is preserved, but we can say for certain that is not currently part of any preservation system.

The Voyager thought experiment may seem forced, but the world is full of examples of bits that may be stable when considered as potentially observable physical properties which we would typically, but not always, rule out as being preserved. Common examples include the bits on a disk that suffered physical damage but which may be recoverable using forensic techniques to those that are encrypted using a key that may or may not be forgotten, using an algorithm that may or may not be breakable using the supercomputers of today or tomorrow. Can these bits in fact be observed? It all depends on the meaning of the word “can”.

Rather than sinking further into this semantic morass, I propose a more operational view of bits. If I operate an automaton of any kind, it will use a fixed access method to observe a storage bit, or perhaps one of a fixed set of alternative methods. A bit is stable insofar as it is observable unchanged using one of these methods. Another way of saying this is that it is more useful to assign stability as a property of the *service* that delivers the value of the bit to the automaton unchanged.

At this point, the astute reader may have noticed that this definition is quite vague. It doesn't say what property of the service we associate with stability, and it doesn't say what we mean by "unchanged." At this point, we fall back on relativism, the last refuge of scoundrels. We say that stability can be any property that is agreed between the service provider and the client, and the notion of stability is also a matter of mutual agreement. In other words, stability means adhering to a Service Level Agreement (SLA) that applies to the bit access service.

Such relativism may seem like an overly general approach to characterizing stability, when compared to the specific-seeming notion of preserving a physical property. However, this difference only exists because the physical characterization of bit stability is inherently vague about when it means to be able to observe a bit. Is a bit preserved if the originally stored value can be correctly recovered 95% of the time? How about 75% or 55%? At what point is the physical mark considered to have been lost?

So the purpose of this argument has been to present the idea that when we talk about bit stability, we mean that a specific access method will adhere to a specific SLA. This gives us a basis to build an automated system that uses that access method, and enables us to be able to treat the terms of the SLA as invariants about the behavior of that service. These are the foundations of a reliable preservation system.

An example may be useful at this point. Consider a physical medium such as a magnetic disk drive. When a bit is written to the magnetic medium as the polarization of ferrous molecules, that polarization may have a certain probability of physical stability, but we generally think of the access method involving the rotation of the disk, the interpretation of a command by the controller, the positioning of a read head, the transfer of data as an electrical signal, and the delivery of a response by the controller. So we can define an SLA at the controller level. However, if the system we are interested in accessing stored data over a network, we might choose to instead define an SLA in terms of a higher level protocol layered on top of a generic protocol such as IP, and interpreted by a file or database system. A particular preservation system may find it useful to express its requirements at any of these levels, or at some other. One system may require an SLA that delivers "five nines" reliability while another may be able to make due with much lower reliability.

The point of this discussion is to lay a groundwork for a discussion on how to define an appropriate metric for bit stability. My proposal is that for any given SLA that defines stability, the metric should simply be the product of the amount of data stored under that SLA multiplied by the duration for which the SLA has been adhered to. Thus, each SLA would give rise to a different and incomparable metric, although they would all be reported in units of bit-years.

Given such a framework, the difficult task is to agree on SLAs that are meaningful and that organizations that store data are willing and able to measure and report. The details of such an SLA would include matters such as what the access protocol is, how often data is to be scrubbed and using what algorithm, how frequently errors are allowed to occur and how they are reported, how often service outages may occur and for what duration, etc.

There are other ways of defining and measuring bit stability, and it is necessary to come to an agreement. I encourage other working group members to criticize this framework or put forward their own for discussion at the July meeting.