;login:

THE MAGAZINE OF USENIX & SAGE

June 2001 • Volume 26 • Number 3

inside:

NEEDLES IN THE CRAYSTACK: WHEN MACHINES GET SICK

PART 4: ENTROPY

by Mark Burgess

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

needles in the craystack: when machines get sick

Part 4: Entropy: The Good, the Bad, and the Aged

by Mark Burgess

Mark is an associate professor at Oslo College and is the program chair for LISA 2001.

<Mark.Burgess@iu.hio.no>

DEDICATED TO THE MEMORY OF CLAUDE SHANNON (1916-2001)

You and I, and everyone on the planet, are doomed to die because of a memory leak in the human genome. For better or for worse, whether bug or a feature, DNA contains a sequence of repeated molecular material called telomeres which is used in the unzipping and replication of the DNA strand. Each time DNA is replicated, one of these telomeres is used up and does not get transferred to the copy. Finally, after 50 or so forks, all the telomeres have been used up, and the cell replication program crashes. It is a classic case of unlimited use of limited resources.

Enzyme telomerase is a garbage collection agent which returns this resource to the resource pool. In the growing fetus, it is hard at work, rejuvenating the stem cells which provide the clay for the developing body. But in time it ceases to be produced and the telometer starts clocking up our fare.

Runaway resource usage is nothing to write home about. It is happening all around us. Until the recent trend toward recycling made a small dent in a huge problem, most of the Earth's resources were harvested and disposed of in such a way that they were unrecoverable. We think little about waste. We consume and we abandon. What we abandon is even toxic to the system: fuel emissions, for example, produce poisonous gases, upset the greenhouse balance and even the protective ozone layer. Klondike Pete with his trusty mule searched the hills for years to dig up a few grams of gold, only for future generations to spread it thinly over electronic devices, which are now being buried under piles of dirt, when we are bored with them so that the gold can never be recovered. Burying nuclear waste might frighten some, but burying precious resources is a more certain threat to our future.

With computers we see the same phenomenon not only in the disposal of hardware, the circuitry, and the cases, but also with the resources of the software: disk space is used wastefully (lack of tidying, growing log files), memory leaks in buggy software (never frees RAM or disk), creeping featurism in software (placing ever greater demands on resources). DOS attacks and spam take advantage of the limitation of finite resources and show us the folly of presumption. The idea that we should reduce our consumption of precious resources is not a popular paradigm in contemporary Western society, but it will be a necessary one.

Environmentally conscious observers have long pointed out the negative effects of resource abuse on the environment, but it is less common to point out the steady decline of our resource pool. It is not just fossil fuels, but metals, forests, and biodiversity itself which are at risk. This familiar problem has always existed and will always exist, because resources are inevitably finite.

Availability of resources has been discussed in many contexts, but all examples of resource depletion are essentially symptomatic of a fundamental phenomenon: the build-up of useless information, of waste. In the safe abstract world of physics, this phenomenon acquired the name of entropy. The concept was originally used in the study of ideal gases, but it was later extended to many other phenomena, as general principles were understood. It was many years before the full meaning of entropy was revealed. Many of us have acquired a mythological understanding of entropy, through accounts of popular physics, as being the expression of what we all know in our guts: that everything eventually goes to hell. Disorder increases. Things break down. We grow old and fall apart.

Entropy: Information's Lost+Found

Although there is nothing wrong with the essence of this mythological entropy, it is imprecise and doesn't help us to understand why resource consumption has inevitable consequences, nor what it might have to do with computers. Entropy is a useful measure, and its increase is an important principle, so understanding it is key to all science. What makes entropy a poorly understood concept is its subtlety. The exuberant mathematician John Von Neumann is reputed to have told Claude Shannon, the founder of information theory, that he should call his quantity H *informational entropy*, because it would give him a great advantage at conferences where no one really understood what entropy was anyway.

Entropy is a useful measure, and its increase is an important principle, so understanding it is key to all science.

Before statistical methods were introduced by Boltzmann and others, entropy was defined to be *the amount of energy in a system which is unavailable for conversion into useful work*, i.e., the amount of resources which are already reduced to waste. This had a clear meaning to the designers of steam engines and cooling towers but did not seem to have much meaning to mathematicians. Physicists like Boltzmann and Brillouin, and later Shannon, made the idea of entropy gradually more precise so that, today, entropy has a precise definition, based on the idea of *digitization* – discussed in the last part of this series. The definition turns out to encompass the old physical idea of entropy as unusable resources while providing a microscopic picture of its meaning.

Think of a digitized signal over time. At each time interval, the signal is sampled, and the value measured is one of a number of classes or digits C, so that a changing signal is classified into a sequence of digits. Over time, we can count the occurrences of each digit. If the number of each type i is n_i, and the total number is N, we can speak of the probability of measuring each type of digit since measurement started. It is just the fraction of all the digits in each class $p_i=n_i/N$. Shannon then showed that the entropy could be defined by

$$H = - p_1 \log_2 p_1 - p_2 \log_2 p_2 \ldots - p_C \log_2 p_C$$

where the base 2 logarithm is used so that the measurement turns out to be measured in "bits." This quantity has many deep and useful properties that we don't have to go into here. Shannon showed that the quantity H was a measure of the amount of information in the original signal, in the sense that it measured the amount of its variation. He also showed that it is a lower limit on the length of an equivalent message (in bits) to which a signal can be compressed.

The scale of entropy tells us about the distribution of numbers of digits. It has a minimum value of zero if all of the p_i are zero except for one, i.e., if all of the signal lies in the same class, or is composed of the same single digit for all time, e.g., "AAAAAAAAA...." This corresponds to a minimum amount of information or a maximum amount of order in the signal. Entropy has a maximum value if all of the p_i are the same. This means that the digitized signal wanders over all possible classes evenly and thus contains the maximum amount of variation, or information, i.e., it is a very disordered signal such as "QWERTYUIOPASD...."

Noise is indeed information: in fact, it is very rich information.

The entropy is just a number: it does not "remember" the sequence of events which led to the value describing state of the system, because it involves an implicit averaging over time (we cannot recover a sequence of changes from a single number). But it does record how much average information was present in the sequence since measurements began.

As a metaphor, entropy is discussed with three distinct interpretations: gradual degradation (the ugly), total information content (the good), and loss of certainty (the bad). Let's consider these in turn.

Ugly: this interpretation is based on the assumption that there are many random changes in a system (due to unexpected external factors, like users, for instance), which cause the measured signal (or state of the system) to gradually wander randomly over all possible states. Entropy will tend to increase due to random influence from the environment (noise). This corresponds to a gradual degradation from its state of order at the initial time. This interpretation comes from physics and is perhaps more appropriate in physics than in computer science, because nature has more hidden complexity than do computers; still, it carries some truth because users introduce a lot of randomness into computer systems. Thus, entropy is decay or disorder.

Good: information can only be coded into a signal by variation, so the greater the variation, the greater the amount of information which it could contain. Information is a good thing, some would say, but this view does not have any prejudice about what kind of information is being coded. Thus entropy is information.

Bad: if there is a lot of information, distributed evenly over every possibility, then it is hard to find any meaning in the data. Thus entropy is uncertainty, because uncertainty is conflicting information.

It is not hard to see that these viewpoints all amount to the same thing. It is simply a matter of interpretation: whether we consider information to be good or bad, wanted or unwanted; change is information, and information can only be represented by a pattern of change. Our prejudicial values might tend to favor an interpretation where a noisy radio transmission contains less information than a clear signal, but that is not objectively true.

Noise is indeed information: in fact, it is very rich information. It contains information about all the unrelated things which are happening in the environment. It is not desired information, but it is information. Much of the confusion about entropy comes from confusing information with *meaning*. We cannot derive meaning from noise, because it contains too much information without a context to decipher it. Meaning is found by restricting and isolating information strings and attaching significance to them, with context. It is about looking for order in chaos, i.e., a subset of the total information.

In fact, at the deepest level, the meaning of entropy or information is simple: when you put the same labels on a whole bunch of objects, you can't tell the difference between them anymore. That means you can shuffle them however you like, and you won't be able to reverse the process.

Entropy grows when distinctions lose their meaning and the system spreads into every possible configuration. Entropy is reduced when only one of many possibilities is prevalent. What does this have to do with forgetfulness and wastefulness? There are two principles at work.

Memory fragmentation of resources can be discussed in terms of entropy.

The first, *grouping by digitization* (ignoring detail), involves reducing the number of classes or distinctions into fewer, larger groups called digits. By assimilating individual classes into collective groups, the number of types of digits is fewer, but the numbers of each type increase and thus the entropy is smaller. This, of course, is the reason for our current obsession with the digital: digital signals are more stable than continuous signals, because the difference between 1 and 0 is coarse and robust, whereas continuous (analog) signals are sensitive to every little variation. The second principle is about *repeated occurrences* of the same type of digit. One digit in the same class is as good as the next, and this means that repeated occurrences yield no more information than can be gleaned from counting. Similarity and distinction can be judged by many criteria and this is where the subtlety arises. When we measure basic resources in terms of abstract definitions (car, computer, sector, variable, etc.) there are often a number of overlapping alternative interpretations, which means the entropy of one abstract classification differs from that of a different abstract classification.

Consider an example: fragmentation of memory resources can be discussed in terms of entropy. In memory management, one region of memory is as good as the next and thus it can be used and reused freely. The random way in which allocation and de-allocation of memory occurs leads to a fragmented array of usage. As memory is freed, holes appear amidst blocks of used memory; these are available for reuse, provided there are enough of them for the task at hand. What tends to happen, however, is that memory is allocated and de-allocated in random amounts, which leaves patches of random sizes, some of which are too small to be reused. Eventually, there is so much randomization of gap size and usage that the memory is of little use. This is an increase of entropy.

Several small patches are not the same as one large patch. There is *fragmentation* of memory, or wasted resources. One solution is to defragment, or shunt all of the allocated memory together, to close the gaps, leaving one large contiguous pool of resources. This is expensive to do all the time. Another solution is quantization of the resources into fixed-size containers. By making memory blocks of a fixed size (e.g., pages or sectors), recycling old resources is made easier. If every unit of information is allocated in fixed amounts of the same size, then any new unit will slot nicely into an old hole, and nothing will go to waste.

This is essentially the principle of car parks (aka, parking lots in the US). Imagine a makeshift car park in a yard. As new cars come and park, they park at random, leaving random gaps with a distribution of sizes (non-zero entropy). New cars may or may not fit into these gaps. There is disorder and this ends up in wastefulness. The lack of discipline soon means that the random gaps are all mixed up in a form which means that they cannot be put to useful work. The solution to this problem is to buy a can of paint and mark out digital parking spaces. This means that the use of space is now standardized: all the spaces are placed into one size/space category (zero entropy). If one car leaves, another will fit into its space.

The reason for dividing memory up into pages and disks into sectors is precisely to lower the entropy; by placing all the spaces into the same class, one has zero entropy and less wastage. C's union construction seems like an oddball data type, until one understands fragmentation; then, it is clear that it was intended for the purpose of making standard containers.

Standardization of resource transactions is a feature which allows for an efficient use of memory, but the downside of this is that it results in increased difficulty of location. Since the containers all look the same, we have to go and open every one to see what is

The accumulated entropy of a change is a measure of the amount of work which would be needed to remember how the change was made.

inside. In order to keep track of the data stored, different labels have to be coded into them which distinguish them from one another. If these labels are ordered in a simple structure, this is easy. But if they are spread at random, the search time required to recover those resources begins to increase. This is also entropy at work, but now entropy of the physical distribution of data, rather than the size distribution. These problems haunt everyone who designs computer storage.

The accumulated entropy of a change is a measure of the amount of work which would be needed to remember how the change was made. It is therefore also that amount of information which is required to *undo* a change. In the car parking example, it was a measure of the amount of resources which were lost because of disorder. In sector fragmentation it is related to the average seek time. Entropy of the universe is a measure of the amount of energy which is evenly spread and therefore cannot be used to do work. We begin to see a pattern of principle: inefficient resource usage due to the variability of change with respect to our own classification scheme. Entropy reflects these qualities and is often used as a compact way of describing them. It is not essential, but it is precise and lends a unifying idea to the notion of order and disorder.

Rendezvous with Ram

In classifying data we coarse grain, or reduce resolution. This means actively forgetting, or discarding the details. Is this forgetfulness deadly or life-giving?

If we recycle the old, we use less resources but are prevented from undoing changes and going back. The earlier state of order is lost forever to change, by erasure. We could choose to remember every change, accumulate every bit of data, keep the packaging from everything we buy, keep all of the garbage, in which case we drown in our own waste. This is not a sustainable option, but it is the price of posterity.

Forgetting the past is an important principle. In a state of equilibrium, the past is unimportant. As long as things are chugging along the same with no prospect of change, it doesn't matter how that condition arose. Even in periods of change, the distant past is less important than the recent past. Imagine how insane we would be if we were unable to forget. One theory of dreaming is based on the idea that dreams are used for short-term memory erasure, collating and integrating with long-term experience.

In *Star Trek: The Next Generation* it was suggested that the android Data never forgets. That being the case, one might ask how come he hasn't exploded already? Where do all those memories go? If elephants never forget, no wonder they are so big! Taxation has long been a way of opposing the accumulation of material wealth or potential resources (money). Throughout the centuries, all manners of scheme have been introduced in order to measure that wealth: hearth (fireplace) tax, window tax, poll tax, income tax, road toll, and entrance fees. Since income tax was introduced, records in the UK have been kept on all citizens' activities for years at a time, although there is an uncanny feeling that tax inspectors might pursue them to the grave for lost revenue, replacing the accumulation of wealth with an accumulation of records. In fact, after 12 years, tax records are destroyed in the UK. A sliding-window sampling model, rather than a cumulative model is the essence of recycling.

Queuing and Entropy

Memory is about recording patterns in space, but entropy of spatial classification is not the only way that resources get used up. Another way is through entropy of time resources. Everyone is familiar with the problem of scheduling time to different tasks. Interruption is the system administrator's lot. As one reader commented to me, his company often insists: drop what you are doing and do *this* instead! It results in fragmentation of process: only a small piece of each task gets done. In computer systems it is algorithmic complexity which is responsible for sharing time amongst different tasks. Context switching is the algorithm multi-tasking computers use for sharing time strictly between different tasks. This sharing of time implies some kind of queuing with all its attendant problems: starvation and priorities. Context switching places tasks in a round-robin queue, in which the system goes through each task and spends a little time on it, by assigning it to a CPU. This is an efficient way of getting jobs done, because the number of objects or classes is approximately constant, and thus the parking lot principle applies to the time fragments. It does not work if the number of jobs grows out of control. But if one simply piles new jobs on top of one another, then none of the jobs will get finished. Anyone who has played strategy games like Risk knows that it does not pay to spread one's army of resources too thinly.

This is much the same problem that is considered in traffic analysis (cars and network packets). At a junction, cars or packets are arriving at a certain rate. The junction allows a certain number to flow through from each adjoining route, but if the junction capacity is too slow, then the entropy of the total resources grows to infinity because the number of different digits (cars or packets) is growing. No algorithm can solve this problem, because it has to focus on smaller and smaller parts of the whole. This is the essence of the expression *a watched kettle never boils* taken to extremes. Spamming or denial of service attacks succeed because resources are used up without replacement. This leads to "starvation" of time resources and/or memory resources.

It was once suggested to me that cars should not have to drive more slowly on the motorway when lanes are closed for repair: according to hydrodynamics, everyone should drive much faster when they pass through the constricted channel, to keep up the flow. Unfortunately, unlike ideal fluids, cars do not have purely elastic collisions. Queues build up because there is a limit to how fast transactions can be expedited by a communications channel.

Queues build up because there is a limit to how fast transactions can be expedited by a communications channel.

Reversible Health and Its Escape Velocity

The message I have been underlining above is that there is a fundamental problem where limited resources are involved. The problem is that reversibility (the ability to undo) depends on information stored, but that information stored is information lost in terms of resources. There is an exception to this idea, however. Some corrections occur in spite of no log being made.

What goes up must come down. Common colds and other mild diseases are not fatal to otherwise healthy individuals. The pinball will end up in its firing slot. A compass always points to magnetic North. Moths fly toward a flame. Adults prefer to look younger. Follow the light at the end of the tunnel. Ideals.

If you drop a ball to the ground, it does not usually land at your feet and remain there: the ground is seldom flat, so it begins to roll downhill until it finds a suitable local minimum. It tries to minimize its potential energy under the incentive of gravitation. Now energy is just a fictitious book-keeping parameter which keeps track of how much it cost to lift the ball up earlier and how much will be repaid by letting it roll down again. Like entropy, energy is a summary of information about how resources are distributed in the face of an unevenness. In thermodynamics, energy U and entropy S appear in relationships with the opposite sign: $dF = dU - TdS$

A potential is an anti-entropy device. It makes a difference by weighting the possibilities.

F is the free energy, or the amount of energy available for conversion into useful work, while U is the total energy and S is the entropy (the amount of energy which is spread about in a useless form). T is the temperature, which acts as an essentially irrelevant integrating factor for the entropy in this formulation.

A potential is an anti-entropy device. It makes a difference by weighting the possibilities. It tips the balance from pure randomness, to favor one or few possibilities or ideals. Think of it as a subsidy, according to some accounting principle, which makes certain configurations cheaper and therefore more likely. A potential can guide us into right or wrong, healthy or unhealthy states if it is chosen appropriately. While entropy is simply a result of statistical likelihood (deviation from ideal due to random change), a potential actually makes a choice.

Potentials are all around us. Indeed, they are the only thing that make the world an interesting place. Without these constraints on behavior, the Earth would not go around the sun; in fact it would never have formed. The universe would just be a bad tasting soup. Emotions are potentials which guide animal behavior and help us to survive. I have often wondered why the writers of *Star Trek* made such an issue about why the android Data supposedly has no emotions. For an android without emotions, he exhibits constantly emotional behavior. He is motivated, makes decisions, shows complex "state" behavior, worries about friends and has "ethical subroutines." The fact that he does not have much of a sense of humor could just as well mean that he originated from somewhere in Scandinavia (this would perhaps explain the pale skin, too).

Probably, complex animals could not develop adaptive behavior (intelligence) without emotions. Whatever we call complex-state information, it amounts to an emotional condition. Emotions may have a regulatory function or a motivational function. They provide a potential landscape which drives us in particular directions at different times. They alter the path of least resistance by enticing us to roll into their local minima. It's pinball with our heads. There is probably no other way of building complex adaptive behavior than through this kind of internal condition. We think of our emotions as being fairly coarse: happy, sad, depressed, aroused, but in fact, they have complex shades. We just don't have names for them, because as was noted in the last issue, we digitize.

Reversal of state can be accomplished without memory of the details, if there is an independent notion of what ideal state is: a potential well, like an emotional context, driving the system towards some tempting goal. That is because a potential curves the pathways and effectively makes them distinguishable from one another, labeling them with the value of the potential function at every point. Health is such a state: a complex multifaceted state whose potential is implemented in terms of a dynamical immune system rather than a static force. Ecologies and societies also have emergent goals and preferences. These represent a highly compressed form of information, which perhaps summarizes a great deal of complexity, just as a curve through a set of points approximates possibly more complex behavior.

If one could make a computer's ideal condition, its path of least resistance, how simple it would be to maintain its stability. Usually, it's not that simple, however. The playing field is rather flat, and armies battle for their position. It is as though hordes of barbaric changes are trying to escape into the system, while administrators representing the forces of law and order try to annex them. Neither one has any particular advantage other than surprise, unless the enemy can be placed in a pit.

Computer systems remain healthy and alive when they recycle their resources and do not drift from their ideal state. This was the idea behind *cfengine* when I started writing it eight years ago. The ideal computer state is decided by system policy, and this weights the probabilities p_n so that randomness favors a part of the good, rather than the bad or the ugly. Although a potential can only guide the behavior on average (there will always be some escape velocity which will allow a system to break its bounds), the likelihood of its long-term survival, in a world of limited resources, can only be increased by compressing complex information into simple potentials. This was the message of *cfengine* and the message of my papers at LISA 1998 and 2000. Now all we need is to design the pinball table.

So p_n – you feelin' lucky?