Computation and Heat

Dexter Kim*

jwkonline@gmail.com

Abstract

This is a summary of Landauer's 1961 paper, "Irreversibility and Heat Generation in the Computing Process" (R. Landauer, **IBM Journal**, **3**, *183 (1961)). The main points of the paper and further refinements from Bennett are outlined.*

I. Introduction

It was believed that to overcome thermal noise in calculation a minimal energy of kT must be provided per elementary logic step on a binary degree of freedom [1]. This belief was overturned when Landauer showed irreversible operations generate heat, and error from thermal fluctuation is negligible compared to other sources [2]. Landauer concluded heat of order kT at minimal is generated per machine cycle due to inevitability of irreversible operations during computation. The rough sketch of Landauer's argument of heat generation [2] will be followed by findings of Bennett [3] that calculation itself is not the true culprit of heat.

II. Landauer's argument on irreversibility and heat

Heat is generated during computation when changing the values stored in memory is done in an irreversible fashion [2]. For a binary computing device, either **zero** or **one** can be stored in a bit of memory. The two possibilities merge into one after a specific value is stored by an operation, such as **restore to one**. During this process, the accessible states in the phase space of memory is compressed by a factor of two, thus the entropy – a state variable – of the memory is reduced by $k \ln 2$. This reduction in entropy must be countered by increase in entropy of somewhere else by the second law of thermodynamics, thus heat of $kT \ln 2$ must be dissipated per bit of memory. Unfortunately, Landauer considered such irreversible operations to be indispensable in computation, leading to a wrong conclusion that computation fundamentally generates heat.

All computational operations act on some degree of freedom in the computing device, which are memories for calculations. The accessible stable states are used to denote a particular value. Landauer considered binary devices.[1] A typical example is a bistable potential well with a particle. The particle being in the left well can be labelled as **zero**, while the other – the particle in the right – can be labelled **one**. Another example is one dimensional free particle with slow Brownian motion, the value represented by its position. A coin on a table can be thought as a mechanical analogue for the former, and an abacus for the latter.

Irreversible operations denote operations for which input cannot be determined from the output of the operation. An archetypal example is **restore to one**, where **one** is stored in the memory regardless of its prior value. This operation must dissipate heat when the operation is blind of initial value, i.e. when the same mechanism is used for all possible initial values. This results from properties of conservative systems; a conservative system is

*Blog: http://dexterstory.tistory.com

[1] Three classes of one-bit memory were inspected in the paper, but it seems sufficient to analyze only one of them (non-dissipative memory) in detail.

fully deterministic and can be reversed in time. When **restore to one** is reversed the product can be both **zero** and **one**, thus the inverse operation is indeterministic. Heat is generated in nonconservative systems, so irreversible operations generate heat.

How can one estimate the lower bound of heat dissipated in the process? The irreversible operation **restore to one** has the property that it does not depend on the initial state of the memory. In other words, even if the initial state of the memory system was not known, the system becomes totally known after **restore to one** has been operated on the system. As entropy is the degree of ignorance, this also means that memory system's entropy reduced after **restore to one** had been applied. The typical value of reduced entropy is $k \ln 2$ per bit of memory, since a memory could have been storing any of two values **zero** and **one**.

By the second law of thermodynamics, the lowering of entropy for some system always accompanies an increase in entropy of another. Landauer argued that this entropy is released to surroundings, the degrees of freedom unrelated to memory. These other degrees of freedom can be thought as a heat reservoir, since they generally greatly outnumber the memory degrees of freedom. The entropy increase in non-memory degrees of freedom is produced by isothermal process, thus requiring minimal heat dissipation of $kT \ln 2$ per bit of memory.

However, one must note that **restore to one** may not dissipate heat when information of initial state can be utilized. If the memory is already **one**, it can be left in that state. Restoring **zero** to **one** can be done reversibly. For memory of bistable potential well type, a force that restores **zero** to **one** does work while pushing the particle up the well. When the particle reaches the unstable equilibrium point, retarding force can be used to retrieve work while going downhill. All work done on the system can be recovered, thus generation of heat can be avoided. For memory of abacus type, dissipation of heat can be made arbitrarily small if work is done slow enough. Of course, this does not mean that avoiding generation of entropy is always possible. This non-dissipating mechanism operates differently for different initial values, so previous memory can be inferred from the state of the mechanism right after the operation; this is just postponing dissipation of heat. On the other hand, this implies that it could be possible to circumvent heat generation when information is not lost. This is where Bennett tackled the original conclusion that heat is generated per machine cycle.

III. Bennett's analysis of computation

The main cause of inevitable heat generation lies in irreversibility of computing systems. Circumvention of heat generation is possible if computation can be done in a fully reversible manner. Bennett showed any calculation can be done reversibly by leaving footprints [3]. Furthermore, the seemingly random history of calculation can be reversibly erased by backtracking the footprints of calculation. The output will be restored to the original input, but the outcome of computation can be copied before retracing the history. No message was scrambled, so entropy did not increase in the process.

A computation returns an output according to the input. When different inputs always yield different outputs, the computation is reversible; the input can be inferred from the output alone. If not, one can always add an extra label to make a computation undoable. However, the term *reversible computation* is reserved for a more restricted class of computation; every step in a reversible computation must be retraceable. This condition makes sure no irreversible operation were done, and obviates any generation of entropy and heat. This also means every step of reversible computation leaves a footprint, and we would have a rather longer history than ordinary computation.

The scrambled history from reversible computation may seem random, but it is not; it can be erased reversibly by retracing the footprints of calculation one by one. The outcome of the computation would be erased at the end, but it can be copied onto some blank memory reversibly before erasure. In the end, only the input and the output of computation remain. Entropy was not generated; the memory used as input has returned to its original state, and we have full information of the output as well.[2] Bennett concluded computation itself does not require heat dissipation, but it is the erasing the information of previous computation that requires heat dissipation.

Is there a physical model for reversible computation? Although elementary, copying is a computation; a new space of memory is filled with some symbol according to the internal state and symbol read from the memory. Therefore a biological system that transcripts genetic code is also a computer. This elementary code-copying machine consists of an enzyme (RNA polymerase) attached to a DNA strand that acts as a template [4]. The enzyme takes a molecule of nucleotide pyrophosphate (complementary to the base of the DNA strand to be copied), forms a covalent band between nucleotides, and then releases a free pyrophosphate molecule. This is a chemical process and is driven by relative concentrations of reactants and products. In other words, the whole process can run backward by raising the concentration of products. In such a process, dissipation of heat can be made arbitrarily small by running the system near equilibrium. Further bizarre forms of reversible computers are treated in [4].[3]

IV. Conclusion

Landauer has shown that irreversible operation during calculation dissipates heat by generating entropy. For a binary device, an irreversible operation generates entropy of $k \ln 2$ and heat of $kT \ln 2$ per bit of memory. Although it was thought that such irreversible operations are unavoidable during computation, Bennett showed the converse; all computation can be made reversibly. It is the irreversible erasure of information (possibly during computation) that necessitates heat generation.

V. Appendix

I. Lifetime of information

Landauer analyzed a bit of memory of bistable potential well type and showed thermal noise cannot be the dominant contributor of errors in computation. His analysis includes lifetime of information stored in memory [2]. The results will be used in Appendix 2 as well.

Let n_L and n_R denote the number of ensembles having the particle at the left and the right well, respectively. The energy at the bottom of the well is given as u_L and u_R, while the height of the well is given as u. The rate of particles jumping from the left well to the right would be of the form $Cn_L \exp[-(u-u_L)/kT]$. The same applies to the contrary case, i.e. $Cn_R \exp[-(u-u_R)/kT]$.

$$\begin{aligned}\frac{dn_L}{dt} &= -Cn_L \exp[-(u-u_L)/kT] \\ &\quad +Cn_R \exp[-(u-u_R)/kT] \\ \frac{dn_R}{dt} &= -Cn_R \exp[-(u-u_R)/kT] \\ &\quad +Cn_L \exp[-(u-u_L)/kT]\end{aligned} \tag{1}$$

This equation can be viewed as a linear transformation of (n_L, n_R) yielding $(dn_L/dt, dn_R/dt)$. The eigenvalues give the characteristic evolution of eigenvectors. The first eigenvalue is $\lambda_1 = 0$, and gives the equilibrium distribution.

$$n_L = n_R \exp[(u_R - u_L)/kT] \tag{2}$$

The other eigenvalue λ_2 gives the characteristic decay time of deviation from equilib-

[2]In some restricted conditions where some output has a unique input, Bennett [3] demonstrated that input can be reversibly erased as well. In this case only the outcome of the computation remains as the result of computation.

[3]The reversible computers considered run according to classical mechanics. This observation triggered the question "how small can computers get?", leading to quantum computers [5].

rium.

$$\lambda_2 = -\frac{1}{\tau} = -C\exp[(u-u_L)/kT] \\ -C\exp[(u-u_R)/kT] \quad (3)$$

In terms of $\Delta = \frac{1}{2}(u_L - u_R)$ and $u_0 = \frac{1}{2}(u_L + u_R)$, equation (3) can be recast in a more convenient form.

$$\frac{1}{\tau} = 2C\exp[(u-u_0)/kT]\cosh[\Delta/kT] \quad (4)$$

The average well bottom u_0 will remain unaffected to first order of switching force. This means equation (4) can be written as

$$\frac{1}{\tau} = \frac{1}{\tau_0}\cosh[\Delta/kT] \quad (5)$$

where τ_0 is the relaxation time for $\Delta = 0$. During this characteristic time, the message in the memory is lost due to thermal fluctuations, decaying with rate $\exp[-t/\tau_0]$; τ_0 is the lifetime of information stored in memory. For a usable memory, this lifetime of information should be very large. Note that τ is the minimum time needed for switching.

II. Thermal noise

Landauer considered thermal noise as unswitched memory in switching process; a fraction $\exp[-2\Delta/kT]$ will resist switching due to thermal excitation. Landauer called this the Boltzmann error, and showed this cannot be a dominant cause of error [2].

During switching time T_s, a fraction T_s/τ_0 of memory is lost from decay of information. For Boltzmann error to be dominant,

$$e^{-2\Delta/kT} > \frac{T_s}{\tau_0} = \frac{T_s}{\tau\cosh[\Delta/kT]} \quad (6)$$

which follows from (5). Assuming $\Delta \gg kT$, equation (6) is modified to

$$e^{-2\Delta/kT} > \frac{2T_s}{\tau}\exp[-\Delta/kT] \\ \therefore \frac{1}{2}e^{-\Delta/kT} > \frac{T_s}{\tau}. \quad (7)$$

On the other hand, the error due to incomplete switching is $\exp[-T_s/\tau]$. It follows that

$$e^{-T_s/\tau} > \exp[-\frac{1}{2}e^{-\Delta/kT}] > e^{-2\Delta/kT}$$

where he last inequality stems from $\Delta \gg kT$. Consequently, the condition that information decay is negligible compared to Boltzmann error leads to the conclusion that incomplete switching is the main cause of error; thermal noise cannot be a dominant source of error.

III. Maxwell's demon

Many have argued Maxwell's demon cannot violate the second law of thermodynamics since measuring the velocity of molecules will require heat generation [6, 7]. In fact, measurement can be made reversibly; there is no need for heat generation in measurements [4]. It is therefore the very act of *erasing information about the molecules* that forbids the demon from violating the second law; measurement is irreversible when some information has to be overwritten [8].

References

[1] R. Landauer, "Zig-zag path to understanding [physical limits of information handling]," *Proceedings of the Workshop on Physics and Computation: PhysComp '94*, IEEE Computer Society Press (1994)

[2] R. Landauer, **IBM Journal**, **3**, 183 (1961)

[3] C. H. Bennett, **IBM Journal of Research and Development**, **17**, 525 (1973)

[4] C. H. Bennett, **International Journal of Theoretical Physics**, **21**, 905 (1982)

[5] R. Feynman, *Feynman Lectures on Computation*, Addison-Wesley (1996) p. 182

[6] C. H. Bennett, **IBM Journal of Research and Development**, **32**, 270 (1988)

[7] M. V. Volkenstein, *Entropy and Information*, Birkäuser Verlag (2009) pp. 154-157

[8] C. H. Bennett, "Notes on Landauers Principle, Reversible Computation, and Maxwell's Demon" (2002) arXiv:physics/0210005v1