Energy and Time Determine Scaling in Biological and Computer Designs

Melanie Moses,[1,2,3*] Stephanie Forrest[1,2,3] George Bezerra,[1]
Benjamin Edwards[1], James Brown,[2,3]

[1]Department of Computer Science
University of New Mexico, Albuquerque, NM, USA.
[2]Department of Biology
The University of New Mexico, Albuquerque, NM, USA.

[3]The Santa Fe Institute, Santa Fe, NM, USA.

[*]To whom correspondence should be addressed
E-mail: melaniem@cs.unm.edu
Address: Department of Computer Science
1 University of New Mexico, Albuquerque, NM USA
Phone: 505-277-3112

Keywords: network scaling, power consumption, metabolic rate, complex systems, energy-time product, organism, computer chip

Classification: Major: ; Minor:

Abstract

Metabolic rate in animals and power consumption in computers are analogous quantities that scale similarly with size. We analyze vascular systems of vertebrates and on-chip networks of microprocessors, where natural selection and human engineering respectively have produced optimized networks. Both network designs simultaneously reduce energy costs and increase flow rates. Using a simple network model, our analysis explains empirically observed trends in the scaling of metabolic rate in organisms and power consumption in chips across several orders of magnitude in size. This result suggests that a single principle governs the designs of complex systems that process energy, materials, and information. Just as fundamental shifts in metabolic energy have accompanied the evolutionary transitions in biology, our models suggest that energy efficiency will change as computer technology becomes more distributed and decentralized in the shift to multi-core and dispersed cyber-physical systems.

1 Introduction

Both organisms and computers have evolved from relatively simple beginnings into complex systems that vary by orders of magnitude in size and number of components. Evolution, by natural selection in organisms and by human engineering in computers, required critical features of architecture and function to be scaled up as size and complexity increased. In Biology, Kleiber's law describes the empirical relation between metabolic rate and many other traits, such as lifespan, heart rate, and number of offspring, with body size [11]. Similarly, computer architecture has Moore's law to describe scaling of transistor density and performance [16], Koomey's law for the energy cost per computation [12], and Rent's rule for the external communication per logic block [7].

We posit that these empirical patterns originate from a common principle: Networks that deliver resources are optimized to reduce energy dissipation and increase flow rates, expressed here as minimizing the energy-time product. That is, both living systems and computer chips are designed to maximize the rate at which resources are delivered to terminal nodes of a network and to minimize the energy cost in the network. In biology, the vascular network of vertebrate animals supplies oxygen and nutrients to every cell, fueling metabolism for maintenance, growth and reproduction. Since energy is a limited resource, we assume that organisms are selected to minimize the energy dissipated in the network [22]. Similarly, computation in microprocessors relies on a network of microscopic wires that transmits bits of information between transistors on a chip. In order to maximize computation speed and minimize power consumption, this network is designed to deliver the maximum information flow at the lowest possible energy cost.

Here, we model vertebrates as composed of regions of tissue that receive oxygen carried by blood via a hierarchical vascular network of pipes, and we model microprocessors as composed

of transistors that perform computation, exchanging information over a modular network of wires. As each system scales up in size, we consider: 1) the rate at which resources are delivered by the network and processed in the nodes; and 2) the energy dissipated during these processes. Despite the obvious differences between organisms and chips, we present a general model and derive energy and time scaling relations from physical principles applicable to each system. Using these relations, we express the optimal network design as a tradeoff between energy cost and processing speed.

Earlier models in biology have assumed either energy minimization, e.g., [22] or optimal resource delivery rate [2], but they have not formalized the tradeoffs between them, as we do here. This formalization helps explain how nature and engineering are able to produce designs that approach pareto-optimal along the energy-time tradeoff. Thus, in biology evolution has produced mammals ranging in size from mice to elephants, rather than converging on a single optimal size, and in computer architecture engineers have designed processors ranging from a few thousand transistors to several billion, each of which fills a specific computational niche.

In the rest of the paper, we first present the unified model of network scaling (Sec. 2 and the basic assumptions underlying the model (Sec. 2.1. We discuss predictions of the model, first for organisms (Sec. 3.1) and then for computers (Sec. 3.2). The model illustrates two important points: integrating energy and time minimization into a single scaling framework; and, adding the contribution of the nodes to that of the network to the scaling analysis. Finally, in Sec. 4 we discuss the implications of these results for evolutionary transitions in nature and engineering.

[MAKE A FIGURE HERE.LIKE GEORGE'S DISSERTATION]

2 A Unified Model of Network Scaling

Vascular systems are hierarchical branching networks where blood vessels (pipes) become thicker and longer as hierarchy increases from the capillaries to the aorta. Similarly, micropro-

Figure 1: Schematic: Panel c illustrates the communication dimension, D_w, which measures how the ratio of internal (intra-module) communication per node to external (inter-module) communication per node scales with the level of the module in the hierarchy. Note, that if we had a biological version, we would have MORE edges.

cessor chips are organized hierarchically into a nested structure of modules and submodules, where wires become longer and thicker as the hierarchical level of a module increases (Figure 1). These wires are organized into metal layers, where short, thin wires are routed on the lowest layers, and long, thick wires are placed on the top layers. We model the scaling of length (l) and thickness (r) of both pipes and wires as

$$l_i = l_0 \lambda^{\frac{i}{D_l}} \tag{1}$$

and

$$r_i = r_0 \lambda^{\frac{i}{D_r}}, \tag{2}$$

where i is the hierarchical level of a branch or module, λ is the branching factor, and D_l and D_r are the length and thickness dimensions. This model is akin to the hierarchical pipe model of vascular systems proposed in [22], $\lambda^{\frac{1}{Dr}}$ and $\lambda^{\frac{1}{Dl}}$ correspond to West et al.'s β and γ respectively. In vascular networks, r represents the radius of cylindrical pipes, and in computer interconnects, r represents the width of wires with aspect ratio 1. The smallest edges occur at $i = 0$, and have constant radius, r_0, and length, l_0, that scales with system size [2].

The length parameter D_l determines the spatial dimension occupied by the nodes of the network [15]. For chips, $D_l = 2$, since transistors are placed on a single two-dimensional layer [10]. For three-dimensional organisms, $D_l = 3$.

Digital circuits scale in a third way in addition to length and radius, which has no direct analog in organisms. Although vascular networks have a single pipe per branch, chips have multiple

wires per module, and their number increases with the hierarchical level. This difference arises from the fact that digital circuits are decentralized networks connecting multiple sources and destinations, while vascular networks are centralized, with blood flowing from a single heart. To account for this difference, we introduce a new equation, in which the communication (or number of wires) per module increases with the hierarchical level as

$$w_i = w_0 \lambda^{i/D_w}, \tag{3}$$

where D_w is the communication dimension and w_0 is the average number of wires per node. This hierarchical scaling of communication is a well-known pattern in circuit design called Rent's rule [7], where $p = \frac{1}{D_w}$ is the Rent's exponent.* This pattern is not unique to circuits and has been shown to occur in many biological networks [20, 3]. Vascular systems correspond to a special case where $w_i = 1$ for all i.

2.1 Assumptions of the Unified Model

Before deriving scaling predictions from the model, we make explicit its assumptions and how they relate to earlier models, both in computation and biology:

1. **Time and energy are equally important constraints:** System designs seek to deliver the maximum amount of resource per unit time for the minimum amount of energy overhead. In computer architecture this relationship is expressed as the *energy-delay product*, which formalizes the insight that a chip that is ten times faster or ten times more energy efficient is ten times better.

*Rent's rule is typically expressed as $C(n) = kn^p$, where C_n is the external communication of a module, n is the size of the module (number of nodes), k is the average external communication of a module with size 1, and p is the Rents exponent. For a hierarchy with branching factor of λ, the size of a module is given as $n = \lambda^i$, where i is the hierarchical level. Therefore, we can rewrite Rents rule as $c_i = c_0 \times \lambda^{ip}$, where $c_0 = w_0$ and $p = \frac{1}{D_w}$.

2. **Steady state**: Resource supply matches processing demand [2]. That is, the network supplies resources continually to the terminal nodes and is always filled to capacity. This avoids network delays and the need to store resources in the system. Specifically,

   (a) System designs balance network delivery rates with node processing speeds, so that resources are delivered at exactly the same rate that they are processed.

   (b) Pipelining: A concept from computer architecture in which resources, e.g., computer instructions, leave the source at the same rate that they are delivered to the terminal nodes, and the network is always full. In the hierarchical networks considered here, the source is the highest branch (root), and the principle holds within every level of the hierarchy. Consequently, resources flow through the network continually at maximum speed, and they do not accumulate at source, sink, or intermediate locations.

3. **Terminal units and service volumes:** In contrast to Ref. [22] we do not assume that terminal units have fixed size. In chips it is well known that transistor size has shrunk over many orders of magnitude. Previous scaling models of biology posit that the service volume (region served by terminal units of the network) actually increases with system size [22, 2]. Following Ref. [2], we assume that capillaries have fixed radius but that their length is proportional to the radius of the service volume. Similarly, the radius of the isochronic region (service volume) for chips scales proportionally with decreasing transistor size.

In addition to these general assumptions, we make the following refinements to accommodate salient differences between biology and computer architecture.

1. In biology, the energy processed by a service volume, E_{node}, is invariant with system

size. That is, as the service volume increases with body size, the total amount of energy processed remains constant. We do not make this assumption for chips.

2. Component packing: In chips, we assume that total chip area is constant, and the number of transistors N is the square of the process size, i.e., the length of one side of a transistor.

In biology it is known that blood flow slows by several orders of magnitude as it travels from the aorta to the capillaries (CITE somebody). Scaling models have generally ignored this slowing [22, 2], but we leave this variable in our equations to highlight where velocity affects time and energy scaling.

3 Predictions of the Model for Organisms and Computers

We define E_{net} and T_{net} respectively to be the energy dissipated and the time taken by the network to deliver a unit of resource to a node. For organisms the resource is oxygen (in mammals, carried by a unit volume of blood), and for computers the fundamental resource is a bit of information. Similarly, we define E_{node} and T_{node} as the energy dissipated and the time taken by a node to process that resource. For organisms, the node is the service volume corresponding to a region of tissue supplied by a single capillary, E_{net} is the energy required to deliver oxygen to the cells (as studied by [22]), and E_{node} is the energy dissipated by cells processing incoming oxygen. T_{net} and T_{node} are the time delay between delivering new resources to the cell and the time taken for the cell to process those resources respectively. The pipelining and steady-state assumptions indicate that $T_{net} = T_{node}$, i.e., supply matches demand.

For computers, the nodes are transistors, and E_{net} and E_{node} represent the energy dissipated delivering bits to transistors and the energy required to process the bits at the node, and T_{net} and T_{node} are the times required to deliver and process a bit at the node (i.e., network and transistor switching delay). In computers the time taken to deliver and process bits is in principle bounded

by the maximum of the communication time $\max(T_{net}, T_{node})$, i.e. a node cannot process another bit until it is delivered, and a node cannot process a new bit until it is done processing the previous bit. Because of the steady state assumption T_{net} and T_{node} are equal. Thus, for both organisms and computers we define the total energy as $E_{sys} = E_{net} + E_{node}$ and the total delay as $T_{sys} = T_{net} = T_{node}$.

In the following, we derive general scaling relationships between these quantities and the number of nodes N, assuming that the energy time product is minimized. That is, system designs seek to deliver the maximum amount of resource per unit time for the minimum amount of energy overhead. This is equivalent to minimizing the energy dissipated to deliver and process a unit of resource, while simultaneously minimizing the time to deliver and process that resource. Assuming that time and energy are equally important, this quantity becomes the energy-time product, a well-known concept in computer architecture referred to as the energy-delay product.

We express the optimal network design as a constraint optimization problem in which the whole system's energy-time product is minimized:

$$\min_{D_r, D_w, D_l} (E_{sys} \times T_{sys}) \tag{4}$$

We derive expressions for E_{sys} and T_{sys} for organisms (Sec. 3.1) and computers (Sec. 3.2) in terms of the dimensions D_r, D_w, and D_l where D_l is fixed by the external dimensions of the system.

3.1 Organisms

In this section, we derive general energy and time scaling relations for the network and nodes, in organisms in order to minimize Eq. 4.

We first derive the energy required to both distribute and process oxygen in biological networks. From basic principles of hydraulics, E_{net} is given by the loss in pressure from the

aorta to the capillaries multiplied by the volume being transported. Pressure is the product between hydraulic resistance (R) and flow (Q), so the loss in pressure $\Delta P = RQ$. Thus $E_{net} \propto \Delta P \propto RQ$.

The resistance of a pipe is given by given by the well-known Hagen-Poiseuille's equation, where R at hierarchical level i is $R_i = \frac{8\mu l_i}{\pi r_i^4}$, where μ is the viscosity constant. The total network resistance R is given by [22]:

$$R = \sum_{i=0}^{H} \frac{8\mu l_i}{\pi r_i^4} \frac{1}{n_i} = \frac{8\mu l_0}{\pi r_0^4} \lambda^{-H} \sum_{i=0}^{H} \lambda^{i\left(\frac{1}{D_l} - \frac{4}{D_r} + 1\right)} \tag{5}$$

where there are $H + 1$ hierarchical levels, and $n_i = \lambda^{H-i}$ is the total number of pipes at hierarchical level i.

West et al. predict that R and therefore E_{net} are minimized by $D_r = 3$ in the lower part of the network, where Eq. 5 dominates. However, this prediction does not account for delivery time, which slows dramatically when $D_r = 3$[†].

We derive upper and lower bounds for D_r in the network given that the goal is to minimize the energy-time product. Recalling that $\lambda^{-H} = N^{-1}$ and that $D_l = 3$ for organisms, in the case where $D_r \leq 3$, the summation in Eq. 5 converges to a constant ($\log(N)$ in the case of $D_r = 3$), and $R \propto l_0 N^{-1}$. As D_r increases above 3, R increases from $\propto l_0 N^{-1}$ to $R \propto l_0 N^{\frac{1}{3} - \frac{4}{D_r}}$. See Sec. 7 for details of the calculation.

Flow through a pipe is defined as $Q = u\pi r^2$, where u is the fluid velocity. Therefore, flow through the aorta equals $Q = u_H \pi r_H^2$, and substituting from Eq. 2, $Q = u_0 \pi r_0^2 \lambda^{\frac{2H}{D_r}} = u_0 \pi r_0^2 N^{\frac{2}{D_r}}$. We do not assume that u_H is independent of N, and therefore we include u_0 in our scaling equations. We assume Q is equal at all levels of the network (steady state assumption) giving: $Q \propto u_0 N^{\frac{2}{D_r}}$.

[†]To understand how pulsitile flow affects resistance in the upper part of the network, see [22], which shows that $D_r = 2$, i.e., the network is area preserving, in this region.

Having derived R and Q it is evident that E_{net} scales differently depending on the value of D_r:

Case 1: $2 \leq D_r \leq 3$: $E_{net} \propto l_0 u_0 N^{\frac{2}{D_r}-1}$

Case 2: $3 \leq D_r \leq 12$: $E_{net} \propto l_0 u_0 N^{\frac{1}{3}-\frac{2}{D_r}}$

We assume that the quantity of energy dissipated to metabolize a fixed amount of nutrients is independent of metabolic rate and, therefore, E_{node} is proportional to the number of nodes in the system, $E_{node} \propto N$.

We next derive the time delay, T_{net}, to deliver a fixed number of oxygen molecules to the node. T_{net} is given by the volume of blood being transported divided by the flow (Q). Since flow is conserved (the pipelining assumption), the flow rate is constant and Q is defined in Eq XX.

We assume that each service volume processes a constant number of oxygen molecules per unit time.

The total number of oxygen molecules processed in the nodes is therefore proportional to the number of nodes, N. This gives us $T_{net} \propto \frac{N}{Q} \propto u_0^{-1} N^{1-\frac{2}{D_r}}$. One might ask why there is no distance term in the previous equation. Because of the steady-state assumption, T_{net} is the time it takes to get the 'next' oxygen molecule from a capillary and not the time it takes a single molecule to traverse the network.

By the steady state assumption we assume that resources are absorbed into the service volume at the same rate as they are delivered and therefore $T_{net} = T_{node}$.

In Section 7, we show how the energy time product scales for all values of D_r. Here we show the case ($D_r \leq 3$) that minimizes the scaling of the energy time product (Eq. 4):

$$\min_{D_r}(RQ+N)\times\left(\frac{N}{Q}\right)\propto RN+\frac{N^2}{Q}\propto l_0+u_0^{-1}N^{2-\frac{2}{D_r}} \tag{6}$$

where N is the number of terminal units. [NEED TO TALK ABOUT WHY THIS LEADS TO THE J CURVE IN SCALING]

In both cases the energy time product is dominated by the second term in Eq. 6 which is minimized by minimizing D_r. We note that $D_r < 2$ is physiologically unrealistic because the network would then be area decreasing, causing blood velocity to increase through the network rather than slowing down to allow oxygen absorption at the nodes. To accommodate the necessary slowing of blood in the capillaries D_r must be greater than 2 [22].

This derivation in Section 7 provides a straightforward theoretical explanation for why $D_r \leq 3$ in the lower region of the network by identifying a discontinuity in the scaling exponent at $D_r = 3$.

The final energy and time scaling relations for organisms are (NOTE: putting these here to keep track for now, and noting that they don't make any sense!):

$E_{net} \propto RQ$ where R is Eq. 6, $Q \propto N^{\frac{2}{D_r}}$ under the minimization condition that $D_r = 3$. Substituting into Eq 6 gives: $R \propto N^{-1}$ and

$E_{net} \propto N^{-1}N^{\frac{2}{D_r}} \propto N^{\frac{2}{D_r}-1}$

$E_{node} \propto N$

$T_{net} \propto N/Q \propto N^{1-\frac{2}{D_r}}$ where $D_r = 2$ in the upper pulsatile area preserving part of the network so $T_{net} \propto N^{1-\frac{2}{D_r}}$ and $D_r = 3$ in the lower Poiseuillle area increasing part of the network so $T_{net} \propto N^{\frac{1}{3}}$.

$T_{node} \propto N^{1-2/D_r}$ matching T_{net}.

3.2 Computers

We now apply the same reasoning to computer chips. In computers, unlike biology, the nodes (transistors) are not of constant size and have shrunk by many orders of magnitude over 40 years of micro architecture evolution. During this time, chip area has grown much more slowly; we assume chip area to be constant for our calculations. Putting these two constraints together, the linear dimensions of transistors and wires decrease with transistor count as $N^{-1/2}$ (or, more generally, N^{-1/D_l}) [17]. This miniaturization process is well understood and is accurately modeled by Dennard's scaling theory [8]. Thus, $r_0 \propto l_0 \propto N^{\frac{-1}{D_l}}$. Intuitively, this means that the number of nodes increases by placing smaller transistors closer together and connecting them with smaller and shorter wires.

In the following, we assume that all wires carry the same flow and that information is transferred synchronously. We first calculate how E_{net} scales with N. From basic principles of electronics, the energy dissipated to transmit a bit over a wire is given by the formula $\frac{CV^2}{2}$, where C is capacitance and V is voltage. Although energy depends on V^2, voltage has remained approximately constant over the last four decades (decreasing only by a factor of five while transistor count increased by six orders of magnitude [18]), so we estimate that the total energy to transmit all bits over the network scales as C [5]. Ignoring fringe effects and for an aspect ratio of 1, wire capacitance is $C_i = \epsilon l_i$ [23], where ϵ is the dielectric constant. The network capacitance is the sum of the capacitances of all wires, which is proportional to the total wire length of the network [9]:

$$C \propto Length \propto \sum_{i=0}^{H} l_i w_i n_i \propto l_0 w_0 \lambda^H \sum_{i=0}^{H} \lambda^{i\left(\frac{1}{D_l}+\frac{1}{D_w}-1\right)} \tag{7}$$

where l_i is the length of wire, w_i is the number of wires per module, and n_i is the number of modules, all at level i.

Recall that $l_0 \propto N^{-1/D_l}$ and $\lambda^H \propto N$ giving

$$C \propto N^{(1-\frac{1}{D_l})} \sum_{i=0}^{H} \lambda^{i\left(\frac{1}{D_l}+\frac{1}{D_w}-1\right)} \tag{8}$$

Similarly to energy scaling in organisms, how C scales depends on whether the exponent $\frac{1}{D_l} + \frac{1}{D_w} - 1$ is positive or negative. If $D_w \geq \frac{D_l}{D_l-1}$ the exponent is negative and the summand converges to a constant ($\log(N)$ in the exact equality case), leaving $E_{net} \propto N^{1-\frac{1}{D_l}}$. When $D_w < \frac{D_l}{D_l-1}$, $C \propto N^{\frac{1}{D_w}}$. See section 8 for details.

This indicates that E_{net} grows sub-linearly with the number of terminal units in the network. This makes sense intuitively because we assume chip area is constant, so that as N increases the distance between nodes is reduced. Additionally, Rent's rule, reflected in D_w, means that most bits move locally so that the distance between nearest nodes affects the average distance that bits are transmitted.

We now calculate the scaling of E_{node}. For a single node, computation energy is given by the transistor's (dynamic) energy dissipation as $\frac{CV^2}{2}$, and summing across all nodes,

$$E_{node} = N\frac{CV^2}{2} \propto NC_0 \propto Nl_0 \propto N^{1-\frac{1}{D_l}} \tag{9}$$

We now calculate how T_{net}, the time to transmit a bit over a wire scales with N. T_{net} is equivalent to the wire latency, L_i, in each wire at each level of the network. The delay in delivering new bits in the network is bounded by the slowest level, implying:

$$T_{net} \propto \max_i L_i \tag{10}$$

This latency is proportional to the product of resistance and capacitance, usually referred to as the RC time constant [1], where R is the wire resistance. For wires with aspect ratio 1, $R_i = \rho l_i / r_i^2$, where ρ is the resistivity of the material. We can then calculate:

$$L_i = RC = \frac{\rho \epsilon l_i^2}{r_i^2} = \frac{\rho \epsilon l_0^2}{r_0^2} \lambda^{i\left(\frac{2}{D_l} - \frac{2}{D_r}\right)} \tag{11}$$

Here if $D_r > D_l$ the largest latency will be at the top of the network L_H and $T_{net} \propto N^{\frac{2}{D_l} - \frac{2}{D_r}}$. However, it is unrealistic for the length of a wire to grow more slowly than its thickness. If $D_r \leq D_l$ the highest latency will be at the bottom of the network, and $T_{net} \propto \frac{l_0^2}{r_0^2} \propto N^0$. Under our steady state assumption, $D_r = D_l$, so that all levels of the network have the same latency, and no time is wasted at any level. For more details see 8.

Computation time for a single node, T_{node}, is calculated as the transistor delay, $\frac{CV}{I}$, at the nodes [1].

$$T_{node} \propto C_0 \frac{V}{I} \propto N^{-1/D_l} \tag{12}$$

XX

Because D_w is not a factor in any other term of the energy time product, we set

Noting that $D_l = 2$, case XXX minimizes E

$D_w = \frac{D_l}{D_l - 1}$. Given a two dimensional chip, $D_l = 2$ and so $D_w = 2$

XX

Summarizing the scaling relationships:

$E_{net} \propto N^{1-1/D_l}$

$E_{node} \propto N^{1-1/D_l}$

$T_{net} \propto N^0$

$T_{node} \propto N^0$

Thus, for 2 dimensional chips in which $D_l = 2$, three components of the energy time product (E_{net}, E_{node}, and T_{node}) scales sub linearly as $N^{1/2}$ but T_{net} and T_{node} are constant with respect to N.

Putting it all together,

$E_{sys} = E_{net} + E_{node} \propto N^{1-1/D_l} \propto N^{1/2}$

$T_{sys} = max(T_{net}, T_{node}) \propto N^0.$

$E_{sys} \times T_sys \propto N^{\frac{1}{2}}$

4 Results and Discussion

The following 2 biology paragraphs need a total rewrite!!

Analysis of Equation **??** determines the optimal value of D_r for organisms. Although energy dissipated by the network decreases as D_r increases, we observe that for $D_r \geq 2$ the energy consumed by the nodes (N) dominates the energy dissipated by the network ($l_0 u_0 N^{\frac{2}{D_r}-1}$). Therefore, there is no additional benefit to setting $D_r > 2$. Similarly, time is reduced by having $D_r = 2$ and, therefore, the minimum energy-time product occurs at $D_r = 2$. This result shows that the optimal network design is area preserving, i.e., when the aggregate cross-sectional area of pipes is the same at all hierarchical levels. Various derivations show that this design leads to the 3/4 power scaling of metabolic rate known as Kleiber's law (e.g., [22, 2]).

The model predicts that, for the optimal network design (i.e., $D_l = 3$ and $D_r = 2$), the system's energy-time product is scales with l_0 and N. This result has important implications for the energetic basis of fitness. Some have proposed that biological fitness maximizes metabolic power (energy/time) [14, 19], whereas others have proposed that it minimizes biological times (e.g., generation times, which is equivalent to maximizing vital rates) [13, 21]. The invariance of the energy-time product is consistent with the fact that fitness of organisms is largely independent of body mass. Organisms of all sizes, from small, fast, low-power microbes to large, slow, powerful mammals, coexist and, therefore, are likely nearly equally fit. This implies a direct trade-off between maximizing metabolic power and minimizing generation times that holds over the many orders of magnitude variation in body mass. The energy-time product reflects

powerful geometric, physical and biological constraints on the evolution of organism designs.

For computers, minimizing the energy time product is trivial: also leading to $D_l = D_r = D_w = 2$,. This result corresponds to ideal scaling, as suggested by Dennard [8], where the linear dimensions of transistors and wires scale at the same rate, and wire delay is constant and the Rent's exponent is 1/2, consistent with observations (NEED citations.)

From this result, we make two important predictions. First, power consumption in chips (total energy dissipated per unit of time) scales as $Power = (N \cdot E_{sys})/T_{sys} \propto N^{1/2}$. Performance is given by the number of computations executed per unit of time, or throughput. Second, assuming that a constant fraction of the transistors is active at each cycle, throughput is predicted to scale linearly with N, i.e. $Throughput \propto N/T_{sys} \propto N$.

We compared the predictions for power consumption with data obtained for 523 different microprocessors over a range of approximately 6 orders of magnitude in transistor count. The data are shown in Figure 2, where the measured exponent was 0.495, which agrees closely with the prediction of 0.5. Consistent data on performance across many technology generations is difficult to obtain because the standards have changed over the years and their adoption by different vendors is not uniform. We were able to obtain normalized performance data for 16 Intel processors from different generations over a range of 6 orders of magnitude in transistor count. The exponent obtained for these data was 1.0, as shown in Figure 3, which agrees exactly with the predictions from our analysis. This suggests that engineered designs approach the theoretical optimal defined by the model. The final energy-time product predicted by the model scales as $N^{-1/2}$, showing that, unlike organisms, as size increases, the energy-delay product decreases systematically. This is a consequence of process technology improvement, since newer, larger processors use hardware that is smaller, faster and more energy efficient.

Our model provides a simple theoretical explanation for the scaling of power and performance in computers over 40 years of microprocessor technology improvements. The excellent

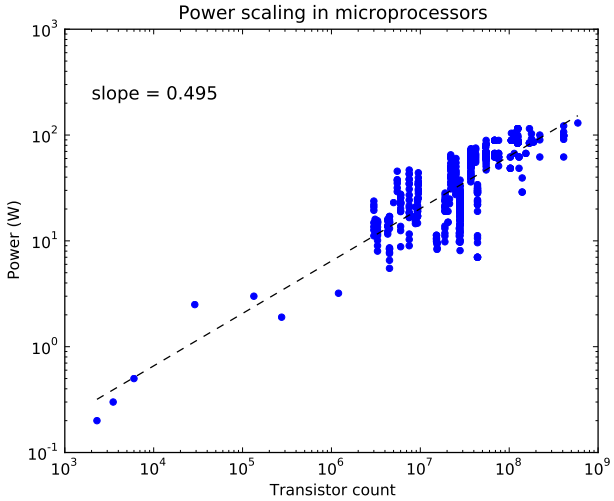

Figure 2:

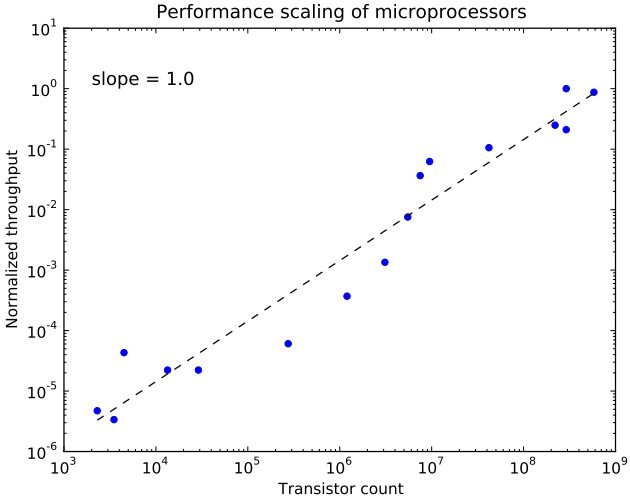

Figure 3:

agreement between the theoretical optimal and experimental data suggests that through successive generations of trial-and-error, innovation and optimization, engineered designs are highly successful, approaching the theoretical limit predicted by the model.

These results also provide an explanation for Koomey's Law, which states roughly: "The number of computations per joule of energy dissipated has been doubling approximately every

1.57 years." This follows directly from minimizing the energy time product and Moore's Law, the empirical observation that the number of transistors (N) doubles every 2 years. NEED a figure that combines data from Fig 2 and Fig 3 to show this?

5 Discussion

- Technological evolution is undergoing a major evolutionary transition as distributed computing changes the metabolic landscape of technology and its interaction with the environment.

- By unifying time and energy into a single model, the model accounts for the wide variation in size of organisms and computers, e.g., mouse to elephant and arm to multe-core.

- We provide an explanation for Koomey's Law (p. 13)

- The model shows why computers have given up on reducing clock speeds and gone to multi-core. Once you reach a minimum component size, then you are at a transition and a new scaling regime. What is the regime and why is biology the guide?

- The joule-second as a new fundamental unit of life and information.

- Theoretical explanation for well-known empirical patterns in computer architecture.

- Nature dealt with network design constraints by slowing metabolism as animal size increases. Computer architecture, at least until recently, focused on increase clock speeds by minimizing component size and incresing energy per chip. Also, using the third dimension to accommodate more wires,

- Computer transions: transistors; transistors to integrated circuits; single core to multicore; desk tops to data centers. Multicore is an evolutionary transition. IBM currently

making 10 nm chips which are the limit for silicon. IBM recently announced a 7 nm transistor (more than 1000 times smaller than the diameter of a red blood cell) and only three times larger than a strand of DNS, using silicon germanium targeting a 50% power improvement. NYT July 9, 2015. Then going to carbon nanofibers.

6 Conclusion

Our analysis provides a unifying explanation for the origin of scaling laws in biology and computing. Despite obvious differences in form and function, the scaling of organisms and computers is governed by the same simple principle. By minimizing the energy-time product, whether through natural selection or engineering, existing designs manage the trade-off between cost and performance, leading to general scaling patterns observed over several orders of magnitude in size. Moreover, power laws as a function of size are not unique to organisms and computers but are widely observed across a large variety of complex systems in nature, society and technology. The scaling of white and grey matter in the brain [24], of energy use and GDP in countries [6], and the pace of life and population in cities [4] are additional examples where a unifying explanation is still lacking. Because cost and performance, i.e., energy and time, impose universal constraints, we suggest that a common design principle governs the scaling of complex systems that process energy, materials and information.

7 Appendix A: Details of Scaling in Organisms

In this section we give a detailed analysis of the derivation of the scaling of the total network resistance discussed in Sec. 3.1. Recall that $D_l = 3$ for 3 dimensional organisms and that $\lambda^{-H} = N^{-1}$. Using these values and simplifying, equation 5 is transformed.

$$R = \frac{8\mu l_0}{\pi r_0^4} N^{-1} \sum_{i=0}^{H} \lambda^{i(\frac{4}{3}-\frac{4}{D_r})} \tag{13}$$

Let the summand $S = \sum_{i=0}^{H} \lambda^{i(\frac{4}{3}-\frac{4}{D_r})}$. $R \propto N^{-1}S$. How S scales with N is dependent on the exponent $\frac{4}{3} - \frac{4}{D_r}$, and reduces to four different cases:

Case 1: $D_r = 3$: In this case the exponent is equal to 0, and the $S = H + 1 \propto \log(N)$, and $R \propto \frac{\log(N)}{N}$, because $\log(N)$ in this case grows much more slowly than N, it is reasonable to conclude that $R \propto N^{-1}$

Case 2: $D_r < 3$: Here (and in subsequent cases) we can use the geometric series to calculate the exact value of S. In particular

$$\begin{aligned} S &= \frac{\left(1 - \lambda^{(\frac{4}{3}-\frac{4}{D_r})(H+1)}\right)}{1 - \lambda^{\frac{4}{3}-\frac{4}{D_r}}} \\ &= \frac{1 - \left(\lambda^H\right)^{(\frac{4}{3}-\frac{4}{D_r})} \lambda^{(\frac{4}{3}-\frac{4}{D_r}))}}{1 - \lambda^{\frac{4}{3}-\frac{4}{D_r}}} \\ &= \frac{1 - N^{(\frac{4}{3}-\frac{4}{D_r})} \lambda^{(\frac{4}{3}-\frac{4}{D_r}))}}{1 - \lambda^{\frac{4}{3}-\frac{4}{D_r}}} \end{aligned}$$

If we let $c = \lambda^{(\frac{4}{3}-\frac{4}{D_r})}$ we see that

$$S = \frac{1 - cN^{(\frac{4}{3}-\frac{4}{D_r})}}{1 - c} \tag{14}$$

Because $\frac{4}{3} - \frac{4}{D_r} < 0$ is negative in this case and N is large in practice, $cN^{(\frac{4}{3}-\frac{4}{D_r})}$ is small, and S is proportional to a constant ($S \approx \frac{1}{1-c}$). This implies that $R \propto N^{-1}$.

Case 3: $3 < D_r < 12$: In this case the exponent in S is positive, meaning that S scales directly with N. Note that $c > 1$ in this case and we can write

$$S = \frac{cN^{(\frac{4}{3}-\frac{4}{D_r})} - 1}{c - 1} \tag{15}$$

This means that $S \propto N^{(\frac{4}{3}-\frac{4}{D_r})}$. This implies that $R \propto N^{-1}S \propto N^{(\frac{1}{3}-\frac{4}{D_r})}$. This means that resistance still scales inversely with size, but at a faster rate than if $D_r \leq 3$.

Case 4: $D_r \geq 12$: This final case is analogous to the one above, except that now resistance scales positively with N, implying that the energy in the network would scale positively with N. This is likely a non-physical possibility, but we include it here for completeness.

8 Appendix B: Details of Scaling in Electronics

In this section we give a detailed analysis of the derivation of the scaling of the network capacitance and network latency discussed in Sec. 3.2.

8.1 Capacitance

Recall that $D_l = 2$ for 2 dimensional computer chips and that $\lambda^{-H} = N^{-1}$. We can then calculate capacitance as:

$$C \propto N^{(1-\frac{1}{D_l}} \sum_{i=0}^{H} \lambda^{i\left(\frac{1}{D_l}+\frac{1}{D_w}-1\right)} \tag{16}$$

Similar to how we handled organsims we are interested in whether the exponent $\frac{1}{D_l}+\frac{1}{D_w}-1$ is positive or negative.

Let the summand $S = \sum_{i=0}^{H} \lambda^{i(\frac{1}{D_l}+\frac{1}{D_w}-1)}$. $C \propto N^{1-\frac{1}{D_l}}S$.

Case 1: $D_r = \frac{D_l}{D_l-1}$: In this case the exponent is equal to 0, and the $S = H + 1 \propto \log(N)$, and $C \propto \log(N)N^{1-\frac{1}{D_l}}$, because $\log(N)$ in this case grows much more slowly than $N^{1-\frac{1}{D_l}}$ and we know $D_l = 2$ for 2 dimensional chips, it is reasonable to conclude that $C \propto N^{\frac{1}{2}}$

Case 2: $D_r > \frac{D_l}{D_l-1}$: Here (and in subsequent cases) we can use the geometric series to calculate the exact value of S, using a similar approach to **??**. In this case the exponent is negative and S

is a small constant, leaving $C \propto N^{\frac{1}{2}}$

Case 3: $D_r < \frac{D_l}{D_l - 1}$: In this case the exponent in S is positive, meaning that S scales directly with N. Now the summand contributes an $N^{\frac{1}{D_l} + \frac{1}{D_w} - 1}$ and $C \propto N^{\frac{1}{D_w}}$.

8.2 Network Delay

Recall that we wish to determine the network latency L which is defined as:

$$T_{net} \propto \max_i L_i \tag{17}$$

with

$$L_i \propto RC = \frac{\rho \epsilon l i^2}{r_i^2} = \frac{\rho \epsilon l_0^2}{r_0^2} \lambda^{i\left(\frac{2}{D_l} - \frac{2}{D_r}\right)} \tag{18}$$

L_i will scale differently depending on the relative values of D_r and D_l.

Case 1: $D_r > D_l$: In this case the fraction in the exponent is greater than 0 and the latency will be highest when $i = H$, resulting in $L \propto N^{\frac{2}{D_l} - \frac{2}{D_r}}$.

Case 2: $D_r < D_l$: In this case the exponent is negative and the highest latency occurs at the bottom of the network $i = 0$, leaving $L \propto \frac{l_0^2}{r_0^2} \propto N^0$

Case 3: $D_r = D_l$: In this case the exponent is 0 and there is equal latency at all levels and $L \propto N^0$.

References and Notes

[1] H. Bakoglu. *Circuits, Interconnections, and Packaging for VLSI.* 1990.

[2] J. Banavar, M. Moses, J. Brown, J. Damuth, A. Rinaldo, R. Sibly, and A. Maritan. A general basis for quarter-power scaling in animals. *Proceedings of the National Academy of Sciences*, 107(36):15816–15820, 2010.

[3] D. Bassett, D. Greenfield, A. Meyer-Lindenberg, D. Weinberger, S. Moore, and E. Bullmore. Efficient physical embedding of topologically complex information processing networks in brains and computer circuits. *PLoS computational biology*, 6(4):e1000748, 2010.

[4] L. Bettencourt, J. Lobo, D. Helbing, C. K"uhnert, and G. West. Growth, innovation, scaling, and the pace of life in cities. *Proceedings of the National Academy of Sciences*, 104(17):7301, 2007.

[5] B. Bingham and M. Greenstreet. Computation with energy-time trade-offs: Models, algorithms and lower-bounds. In *Parallel and Distributed Processing with Applications, 2008. ISPA'08. International Symposium on*, pages 143–152. IEEE, 2008.

[6] J. Brown, W. Burnside, A. Davidson, J. Delong, W. Dunn, M. Hamilton, N. Mercado-Silva, J. Nekola, J. Okie, W. Woodruff, et al. Energetic limits to economic growth. *BioScience*, 61(1):19–26, 2011.

[7] P. Christie and D. Stroobandt. The interpretation and application of rent's rule. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(6):639–648, 2000.

[8] R. Dennard, F. Gaensslen, V. Rideout, E. Bassous, and A. LeBlanc. Design of ion-implanted mosfet's with very small physical dimensions. *Solid-State Circuits, IEEE Journal of*, 9(5):256–268, 1974.

[9] W. Donath. Placement and average interconnection lengths of computer logic. *Circuits and Systems, IEEE Transactions on*, 26(4):272–277, 1979.

[10] W. E. Donath. Wire length distribution for placements on computer logic. *IBM J. Res. and Development*, 25:152–155, 1981.

[11] M. Kleiber. Body size and metabolic rate. *Physiological Reviews*, 27(4):511, 1947.

[12] J. Koomey, S. Berard, M. Sanchez, and H. Wong. Implications of historical trends in the electrical efficiency of computing. *Annals of the History of Computing, IEEE*, 33(3):46–54, 2011.

[13] S. Lindstedt and W. Calder III. Body size, physiological time, and longevity of homeothermic animals. *Quarterly Review of Biology*, pages 1–16, 1981.

[14] A. Lotka. *Elements of mathematical biology*. Dover Publications New York, 1956.

[15] B. Mandelbrot. *The fractal geometry of nature*. Wh Freeman, 1983.

[16] G. Moore et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.

[17] M. Moses, S. Forrest, A. Davis, M. Lodder, and J. Brown. Scaling theory for information networks. *Journal of the Royal Society Interface*, 5(29):1469, 2008.

[18] T. H. Ning. A perspective on the theory of MOSFET scaling and its impact. *IEEE Solid State Circuits Newsletter*, 12(1):27–30, 2007.

[19] H. Odum et al. *Environment, power and society.* New York, USA, Wiley-Interscience, 1971.

[20] S. Reda. Using circuit structural analysis techniques for networks in systems biology. In *Proceedings of the 11th international workshop on System level interconnect prediction*, pages 37–44. ACM, 2009.

[21] R. Sibly. The life-history approach to physiological ecology. *Functional Ecology*, 5(2):184–191, 1991.

[22] G. West, J. Brown, and B. Enquist. A general model for the origin of allometric scaling laws in biology. *Science*, 276(5309):122, 1997.

[23] N. Wilhelm. *Why Wire Delays Will No Longer Scale for VLSI Chips*. Sun Microsystems Laboratories, 1995.

[24] K. Zhang and T. Sejnowski. A universal scaling law between gray matter and white matter of cerebral cortex. *Proceedings of the National Academy of Sciences*, 97(10):5621, 2000.

Figure legends

Figure 1: Log-log plot of power consumption as a function of transistor count for 523 computer microprocessors from different vendors and technological generations. The linear regression slope is 0.495 with correlation coefficient of 0.81. A regression that removes the first seven data points and focuses on the main cloud of more modern chips produces a slope of 0.487.

Figure 2: Log-log plot of normalized throughput as a function of transistor count for 16 Intel microprocessors from different technological generations. The linear regression slope is 1.0 with correlation coefficient of 0.97.