Chapter 6

Reversibility and physical scaling laws

In this chapter we analyze how the use of reversibility can improve how well various measures of computational cost-efficiency will scale as we increase the size of our machines, or the size of the problems we are trying to solve. Our analysis establishes that *only* reversible computers are capable of realizing the maximum level of computational scalability that is afforded by the laws of physics.

Moreover, even with today's relatively primitive level of technology, substantially reversible computing can already be the most cost-effective solution in contexts where energy dissipation is a dominant concern, such as in portable devices, or large supercomputing systems. Also, we expect that as device technology improves, reversible operation will become more and more favored.

In chapter 3 we discussed how existing reversible models of computation compare with irreversible models when using a variety of non-physical measures of cost, such as are used in traditional computational complexity theory. Using those measures, we saw that reversibility did not improve efficiency, and in some models could be proven to actually degrade efficiency (§3.4), when carried to the extreme of total reversibility. But the problem with taking those results at face value is that, as we saw in the previous chapter, traditional computation models and cost measures are not realistic; they do not reflect real costs and the physical constraints on computation that we discussed in chapter 2.

In section 3.2.2, p. 54, we introduced some new cost measures which we proposed were more physically appropriate than are the quantities that are traditionally measured in computational complexity theory. In sec. 6.2 (p. 124) we will perform an analysis of physically realistic models of computation using various such physical cost measures, and show that using those models and measures, reversibility can be seen to increase overall efficiency.

Let us now introduce the general classes of models which we will analyze.

6.1 Types of architectures studied

For convenience, in this chapter we will use the term *architecture* to denote the concept of a model of a family of physically-implementable machines, such as those we proposed in ch. 5. Section 6.2 of this chapter analyzes the properties of several very general classes of architectures, which we will define below: fully irreversible architectures (FIA), time-proportionally reversible architectures (TPRA), and ballistically reversible architectures (BRA). All three of these classes will have a number of features in common.

6.1.1 Shared properties

The machine classes we study will all be imagined to be implemented in some fixed underlying technology, which we take to mean that several quantities are fixed across all three classes of machines:

1. There is a fixed minimum physical size (mass and volume) for storing a bit of computational state.

2. There is a fixed maximum physical entropy density allowable within the machines in question, including in their cooling systems.

   The above two items can be justified on the basis of the limits presented in §2.2, along with the argument that mass densities, energies, and temperatures will not be able to be increased indefinitely in any computer technology realizable in the foreseeable future.

3. There is a fixed maximum rate at which bit-operations can be performed per unit of mass and per unit of volume in the machine.

   This limit follows from the fundamental Margolus-Levitin bound we mentioned in §2.4; much tighter bounds than this will certainly hold for all technologies through the foreseeable future.

Now let us distinguish the three classes of architectures that we will study.

6.1.2 Fully irreversible architecture

A *fully irreversible architecture* FIA is one in which there is a fixed constant lower bound, independent of the machine size or of any adjustable parameters of the architecture, on the average number of bits of *computational information* that are lost (converted to entropy) per primitive computational operation that is performed. Note that this does not count the mere conversion of bits that may *already* be entropy from a controlled digital form to an uncontrolled physical form. We are concerned here only with the amount of *new* entropy that is generated per operation due to the architecture.

An architecture is fully irreversible if, for example, it routinely uses ordinary irreversible logic gates, which must produce entropy every time they erase a bit, according to Landauer's principle (§2.5).

6.1.3 Time-proportionately reversible architecture

A *time-proportionately reversible architecture* TPRA is one that provides the option to reduce the average entropy S generated per primitive operation to an arbitrarily small amount that is asymptotically proportional to the inverse of the amount of time t_{op} over which individual operations are performed; that is, $S \sim 1/t_{\mathrm{op}}$. In such architectures, the "degree of irreversibility" (entropy generated per operation) is inverse to this time, so the "degree of reversibility" can be considered *proportionate* to time. Thus we use the adjective "time-proportionately reversible," to describe these machines, the motivation being that this is much more precise than alternative adjectives such as "adiabatic," "asymptotically reversible," and "quasistatic" which have often been used in the past when referring to technologies that have this particular property. (See §7.3 for further discussion of this terminology issue.)

As we will see in chapters 7 and 8, a large number of existing and proposed logic-device technologies are capable of implementing time-proportionate reversibility; so the TPRA model is certainly realistic for purposes of an asymptotic scaling analysis. However, the constant of proportionality (which we call the "entropy coefficient") varies greatly across different technologies, so the range of validity of the asymptotic analysis depends significantly on the technology in question.

6.1.4 Ballistic reversible architecture

This next class of "architectures" may or may not actually be realistic, but it will be a useful point of comparison, which will help us interpret the results of our analysis of the TPRA. The ballistically reversible architecture BRA is a model based on an imagined technology where the entropy generated per constant-time operation can be made exactly zero, or at least so close to zero that the difference does not matter for any achievable scale of machines.

A BRA is the conceptual limit of a TPRA in which the entropy coefficient becomes arbitrarily small. It is appropriate to consider this limit because we do not yet know of

Symbol	Name of architecture class	Entropy generated per operation
FIA	Fully irreversible architecture	$\mathbf{\Theta}(1)$
TPRA	Time-proportionally reversible architecture	$\mathbf{\Theta}(1/t_{\mathrm{op}})$
BRA	Ballistically reversible architecture	0

Table 6.1: The three classes of physical machine models that are compared in this chapter. The defining difference between them is in how the average entropy generated per computational operation scales in relation to the length of time t_{op} over which the operation is performed.

---

any fundamental physical restrictions on how low the entropy coefficient can actually be made to be.

Note that in both the TPRA and the BRA we specify that *fully* logically reversible operation is permitted, but not required. In these models, we also provide the option to perform logically irreversible operations which generate constant entropy. (Moreover, the type of operation to use should be selectable at run time.) This allows these models to use the external universe as a garbage-information dump, just like the FIA does; this option ensures that our reversible machines will be *at least* as powerful as the FIA, since it will be subsumed as a special case, one in which the time-proportionate reversibility feature is effectively unused.

Table 6.1 summarizes the three classes of architectures we will compare in this chapter.

A general feature of these analyses will be attention to some of the subtle ways in which several kinds of physical constraints, such as limits on entropy density and propagation speed, interact with each other to determine the form of the most cost-efficient possible machines.

The structure of the rest of this chapter will be, roughly, to proceed from the simpler, less compelling physical cost measures and analyses to more sophisticated and realistic ones.

6.2 Analyses under various physical costs

In this section we determine, for various cost measures $\$$, the *reversible advantage* $\mathcal{A}_{\mathrm{r}}$ under the given cost measure. We define $\mathcal{A}_{\mathrm{r}}$ as the asymptotically fastest-growing value of the cost-efficiency ratio $\%_{\$\mathrm{rev}}/\%_{\$\mathrm{irr}}$, as a function of cost, for any class of computational tasks. Equation (3.1), p. 54 defined cost-efficiency as

$$\%_{\$} = \frac{\$_{\mathrm{min}}}{\$}.$$

Therefore, letting $\$_{\mathrm{i}}$ and $\$_{\mathrm{r}}$ be the costs on an irreversible and reversible machine, respectively,

$$\begin{aligned} \mathcal{A}_{\mathrm{r}} &= \%_{\$\mathrm{rev}} / \%_{\$\mathrm{irr}} \\ &= (\$_{\mathrm{min}}/\$_{\mathrm{r}})/(\$_{\mathrm{min}}/\$_{\mathrm{i}}) \\ &= \$_{\mathrm{i}}/\$_{\mathrm{r}}, \end{aligned}$$

that is, the reversible advantage is equal to the ratio of the cost on an irreversible machine to the cost on a reversible machine. (And similarly for the ballistic advantage $\mathcal{A}_{\mathrm{b}}$.)

We will often normalize $\mathcal{A}_{\mathrm{r}}$ by expressing it as a function of $\$_{\mathrm{r}}$, the cost on the reversible machine. So, if we write $\mathcal{A}_{\mathrm{r}} \sim f(\$_{\mathrm{r}})$, this means there are classes of computations such that, for instances that cost $\$_{\mathrm{r}}$ to perform on a TPRA (reversible) machine, the cost to perform them on an FIA (irreversible) machine in general is $\Theta(f(\$_{\mathrm{r}}))$ times larger. If $f \sim 1$ this indicates no reversible advantage; any $f \succ 1$ indicates an asymptotically unbounded reversible advantage, as the cost level increases.

It is important to keep in mind that the true reversible advantage is determined by the *best possible* efficiency of each of the two classes of machines on the problem in question. To show a reversible advantage greater than $\Theta(1)$ (no asymptotic advantage), we have to show that *no* FIA machine can perform a given computation with less than a given asymptotic cost that *is* achievable on a TPRA.

Moreover, throughout this section we will be concerned only with *sustainable* costs; that is, an assessment of a computation's cost will only be considered to be fair if a long series of $N \gg 1$ repetitions of computations like it could be performed on the given machine class with no more than N times the cost. This will allow us to marginalize factors such as the time required for set-up of the initial state and read-out of the result. It is assumed that this is fair to do, because there are many useful computations that *are* of a form that requires numerous sequential iterations of a procedure.

Some of the analyses and results in this section were first reported in our earlier publications [70, 71].

6.2.1 Entropy cost

Perhaps the simplest physical measure of cost, which also gets us away from the bias towards the abstract time and space cost-measures featured in traditional complexity theory, is the idea of the cost of a computation being proportional to just the amount of new entropy that it generates.

This measure makes sense for several reasons:

1. Entropy takes up space, and when too much of it accumulates within a fixed-size system, it causes the system to become disordered in uncontrollable ways. For example, a computer might melt if it produces too much entropy without removing it.

2. As we saw in §2.5.3, energy is required to support the existence of entropy in any system at non-zero temperature. Therefore it costs us free energy whenever entropy is generated. As we mentioned in §2.5.4, the coolest accessible place to dump large amounts of entropy is the cosmic microwave background at ≈ 3 K, so each bit's worth of sustained entropy generation costs us at least $\approx 3{\times}10^{-23}$ J (≈ 0.2 meV) of energy which cannot be recovered. (Except maybe by waiting for the universe to cool further, which will take a while!)

3. Even in the distant-future limit, if there is a finite upper bound to the maximum entropy of the universe, then negentropy ($S_{\rm max}{-}S_{\rm current}$) is a truly non-renewable resource; once we use it up, no further entropy-producing operations will be possible. (There's a cost measure for you!) So the efficiency of our use of entropy is crucial if we wish to maximize the total amount of computational work that we accomplish throughout all time.

Scaling comparison. With entropy alone as the cost measure, $\$ = S$, the comparison is of course straightforward. The irreversible FIA machine by definition produces constant entropy per operation, so the cost of any computation scales as the number of primitive operations, $\$ \sim N_{\rm ops}$.

The ordinary reversible machine TPRA, given unbounded space, can be run in fully logically reversible fashion using Bennett's 1973 algorithm (with the same order $N_{\rm ops}$ as the FIA), and still produce no computational entropy other than, at most, the size $n_{\rm in}$ of the input problem, and that only if the input is no longer needed after the computation. The entropy generation due to friction can be made arbitrarily smaller than $n_{\rm in}$, by extending the computation over a sufficiently long period of time. Thus the total entropic cost is at most equal to the input size, $\$ \lesssim n_{\rm in}$.

Similarly for the ballistic BRA, except that we do not have to run the machine indefinitely slowly to achieve that low of a cost.

Thus, unsurprisingly, when entropy is the cost measure, reversible machines completely dominate irreversible ones in their cost-efficiency. Since for arbitrary problem classes, $N_{\rm ops}$ may scale arbitrarily quickly with $n_{\rm in}$, the reversible advantage factor may be an arbitrarily fast-growing function of the input size.

Of course, using entropy as the sole cost measure is not particularly compelling, because it ignores the opportunity cost of using up some amount of physical space for the amount of time required by the computation. This is particularly apparent for the case of the TPRA which may consume a large amount of space (for example, proportional to $N_{\rm ops}$) for a large amount of time ($\boldsymbol{\Omega}(N_{\rm ops}^2/n_{\rm in})$ to get the physical entropy generation below $\mathcal{O}(n_{\rm in})$). When minimizing entropy only, total spacetime resources for a computation will likely be polynomially larger for the reversible computation. Thus it behooves us to consider those costs as well.

Before we study true spacetime costs, let us first consider another measure of cost that is easier to analyze, but still takes into account measures of both run-time and machine size.

6.2.2 Area-time product

For purposes of this section, we will characterize machine size as the surface area A of the least-area surface that encloses all of the computer's active information-processing components. Note that if the "computer" happens to consist of many independent components that are spread far apart from each other over a large surface, then under our definition, the least-area surface enclosing the system may actually consist of many separate small surfaces, one around each component.

In any case, one reason to think of area as a component of a cost measure is that it measures quantities such as desktop footprint, floor space, and land (planetary surface), which have everyday significance as resources. Moreover, present computer manufacturing technology, based on building up structures on the surfaces of silicon wafers, is geared towards building dense circuitry in only two dimensions, so area is a frequent cost measure in that arena as well.

More fundamentally, due to the limits on entropy density we assumed in §6.1, we will see in a moment that minimum surface area determines the maximum sustained rate at which entropy can be produced within the surface. If a system actually does produce entropy at this rate, it thereby subtracts correspondingly from the maximum rate at which entropy can be produced by the remainder of any larger system within which it is enclosed. So, area makes sense as a component of computational cost.

Multiplying the area by time converts it to a measure of the *rent* that the area would yield over the course of the computation if it were rented out for other purposes (such as for alternative computations). This makes sense in intuitive economic terms, and it also corresponds to a bound on the total amount of entropy that the given system could have produced over the given amount of time.

6.2.2.1 Rate of computation as a function of area

For computing the area-time product, let us first ask, how does the maximum rate of computation scale as a function of area?

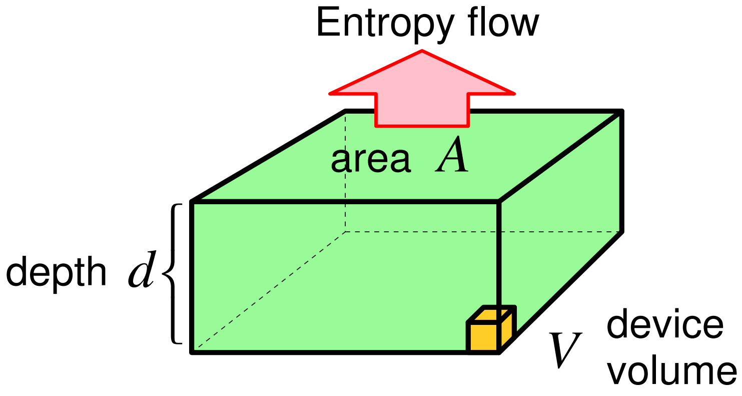
For now, we will characterize the raw processing rate $\mathcal{R}_{\text{op}}$ in terms of the number of primitive computational operations (such as logic gate operations) performed per unit of real time. We also assume, for now, that the computation being performed is an inherently logically reversible one that does not require asymptotically more computational steps or memory on a reversible processor; this will allow us to treat time-proportionally reversible operations as equivalent to irreversible operations for our purposes. An example of such a computation would be a simulation of a logically reversible system; we will see other examples in ch. 9.

Irreversible machine. The FIA machine by definition produces $\mathbf{\Theta}(1)$ entropy per operation, and we assume as always that entropy densities are limited. As per our arguments in §2.3, the rate of entropy removal per unit area is therefore also limited. Since the total volume within the given area is limited (it's at most $\mathcal{V} \leq \frac{1}{6}\pi^{-1/2}A^{3/2}$), it follows that for a long computation, the highest rate of entropy generation that is sustainable is just equal to the maximum rate at which the entropy may leave through the surface. This rate is bounded by the fixed maximum entropy flux F_{S} times the minimal enclosing area A. Thus $\mathcal{R}_{\text{op}} \precsim A$.

Reversible machine. Let the TPRA contain logic devices at constant average density, so that the total number of logic devices N_{dev} is proportional to the TPRA's volume $\mathcal{V}$, and let the TPRA also be roughly spherical (a cube would suffice) so that $\mathcal{V} \sim A^{3/2}$. Then, the number of devices $N_{\text{dev}} \sim A^{3/2}$. If each device operation takes time t_{op}, and all operations are reversible, the entropy per operation is $\mathbf{\Theta}(1/t_{\text{op}})$ and so the total rate of entropy generation is $\mathcal{R}_{\text{S}} \sim N_{\text{dev}}/t_{\text{op}}^2 \sim A^{3/2}/t_{\text{op}}^2$. Since $\mathcal{R}_{\text{S}}$ must be no greater than the rate $\mathcal{O}(A)$ of entropy removal, we have that $t_{\text{op}} \succsim A^{1/4}$. Letting $t_{\text{op}} \sim A^{1/4}$, we have $\mathcal{R}_{\text{op}} = N_{\text{dev}}/t_{\text{op}} \sim A^{5/4}$.

Thus, within area A, the TPRA can run $\mathbf{\Theta}(\sqrt[4]{A})$ times faster than the FIA, on computations that involve only logically reversible operations. For an approximate sphere/cube of diameter $d \sim \sqrt{A}$, the speed advantage of the reversible machines scales as $\mathbf{\Theta}(\sqrt{d})$. Many such cubes can be arranged beside each other in a plane without changing the asymptotic area available to each one, forming a flat slab of material of thickness (depth) d, which can perform at a per-area rate of $\mathbf{\Theta}(\sqrt{d})$. (See figure 6.1.) An irreversible machine, in contrast, would be capable of only a constant rate per unit area.

Ballistic machine. In this case there is no entropy production, so the maximum rate of operation scales with volume, $\mathcal{R}_{\text{op}} \sim A^{3/2}$. This is a factor of $\sqrt{A}$ times faster than the irreversible machine and $\sqrt[4]{A}$ times faster than the time-proportional



$$\left.\begin{array}{lll}\text{Number of devices} & N_{\text{dev}} & = Ad/V \\ \text{Entropy generation rate} & \mathcal{R}_{\text{S}} & = N_{\text{dev}}S/t_{\text{op}} \\ \text{Entropy removal rate} & \mathcal{R}_{\text{S}} & = F_{\text{S}}A \\ \text{Entropy per operation} & S & = k_{\text{S}}/t_{\text{op}} \\ \text{Total rate of operation} & \mathcal{R}_{\text{op}} & = N_{\text{dev}}/t_{\text{op}}\end{array}\right\} \mathcal{R}_{\text{op}} = A\sqrt{\frac{dF_{\text{S}}}{k_{\text{S}}V}}$$

Max rate achieved when $t_{\text{op}} = \sqrt{dk_{\text{S}}/F_{\text{S}}V}$.

Beats irreversible $\mathcal{R}_{\text{op}} = A\dfrac{F_{\text{S}}}{S}$ whenever $d > \dfrac{F_{\text{S}}k_{\text{S}}V}{S^2}$.

Figure 6.1: Speed limit for reversible machines of minimum-surface area $\mathbf{\Theta}(A)$ and thickness $d \precsim A^{1/2}$. The maximum rate of computation scales as $\mathbf{\Theta}(A\sqrt{d})$.

reversible machine. For a sphere or slab of thickness d the ballistic machine is $\Theta(d)$ times faster than the irreversible machine, and $\Theta(\sqrt{d})$ times faster than the TPRA.

So the TPRA is, in a sense, "halfway" between irreversible and ballistic machines in terms of rate per unit area; its benefit factor above the irreversible machine is the square root of that for the ballistic machine.

6.2.2.2 Minimum area-time product

Now that we know how speed scales with area, let us see how to choose the shape of a machine so as minimize the area-time product AT for a given computation requiring N_{ops} operations, under our three classes of architectures.

Let us assume we are dealing with a restricted class of computational tasks in which no communication is required between processors during the course of the computation: the task can be expressed as $\Theta(N_{\text{ops}})$ separate computational tasks that can be performed entirely independently of each other. This is approximated by, for example, a brute-force search problem in which a very large number of independent possibilities need to be checked for a solution, and checking each one takes roughly constant time, independent of the number checked. (Remember, we can amortize away the set-up and read-out times, because we are concerned with determining a sustainable rate for many iterated repetitions of the given computation.)

Irreversible. If all the area can be used effectively, $\mathcal{R}_{\text{op}} \sim A$, so the time $\mathcal{T}$ for N_{ops} operations is $\Theta(1/A)$, and so $A\mathcal{T} \sim A(1/A) = 1$. Thus the choice of the area of the machine does not affect the asymptotic area-time product. To see what the area-time product is as a function of N_{ops}, consider spreading the processors arbitrarily far apart over a 2-dimensional plane. The minimum-area surface will then consist of a collection of small surfaces, one enclosing each processor, thus the total surface area will be proportional to the number of processors ($A \sim N_{\text{proc}}$), and if we give each processor a constant-size, constant-time piece of the total problem, the number of processors N_{proc} scales as N_{ops}, and the whole computation takes constant time, and $A\mathcal{T} \sim N_{\text{ops}}$.

Reversible. Let the N_{ops} operations again be performed in parallel on $N_{\text{proc}} \sim N_{\text{ops}}$ processors, but this time in a compact structure with area $A \sim N_{\text{proc}}^{2/3}$. In §6.2.2.1 we already derived that the maximum rate of computation for this TPRA structure is $\Theta(A^{5/4})$, so the minimum time $\mathcal{T}$ for N_{ops} operations is $\Theta(N_{\text{ops}}/A^{5/4})$, or $\Theta(N_{\text{ops}}/N_{\text{ops}}^{5/6}) \sim N_{\text{ops}}^{1/6}$. Thus $A\mathcal{T} \sim N_{\text{ops}}^{5/6}$ in this configuration. Can we do better by spreading the processors out? No, because when we decrease the thickness by a factor of x, the area increases by a factor of $\Theta(x)$, but the time only scales down by $\Theta(\sqrt{x})$, so the area-time product increases by $\Theta(\sqrt{x})$. So the optimal configuration is the one we chose, where the diameter is asymptotically minimal.

Ballistic. In the ballistic machine we perform the N_{ops} operations in parallel on $N_{\mathrm{proc}} \sim N_{\mathrm{ops}}$ processing elements in constant time, and because they produce no entropy we can cram them inside the minimal surface area $A \sim N_{\mathrm{proc}}^{2/3}$ without worrying about entropy removal, and so the area-time product for the whole computation is $\mathbf{\Theta}(N_{\mathrm{ops}}^{2/3})$.

Thus for these inherently reversible, completely parallelizable computations, composed of $\mathbf{\Theta}(N_{\mathrm{ops}})$ independent constant-time sub-computations, the TPRA reversible model provides an area-time cost-efficiency advantage of $\sqrt[6]{N_{\mathrm{ops}}}$, again the square root of the benefit of $\sqrt[3]{N_{\mathrm{ops}}}$ that would be provided by a perfectly reversible ballistic computer.

In terms of the cost on the reversible machine, the reversible advantage grows as $\$_{\mathrm{r}}^{1/5}$ for this type of problem. This is the highest scaling possible for this cost measure, because for both FIA and TPRA models, the optimal solution for this problem could be achieved using the same structure: a compact, maximally-parallelized structure. This is already the structure that favors reversible operation the most, since structures that are smaller or more spread out will be less limited by entropy removal; and computations that are less parallelizable will require smaller machines for a given N_{ops} to minimize the area-time product.

Thus, we need not consider other types of computational tasks; we have established that the best reversible advantage $\mathcal{A}_{\mathrm{r}}$ for the area-time cost measure is exactly $\mathbf{\Theta}(\$_{\mathrm{r}}^{1/5})$. This area-time advantage does not grow as quickly as the reversible entropy advantage of §6.2.1 did, but it still becomes unboundedly large as we compare machines at larger and larger cost levels.

6.2.3 Time cost

We have seen how, given a measure of cost consisting of area times time, reversibility yields a scaling advantage. But what if we don't agree that there should be a surface area factor in the cost? Can reversibility provide any benefits for optimizing run-time, by itself?

For the sort of problem considered in the previous section, in which no communication is required between parts of the computation (during the computation itself), it is clear that reversibility provides no asymptotic speed benefit. To minimize the run-time, the processors performing the independent pieces of the computation can simply be spread far enough apart so that the minimal enclosing area becomes proportional to the number of processors, and then entropy removal no longer constrains the asymptotic minimum time, even in the fully irreversible case. The run-time in all models is then $\mathbf{\Theta}(1)$, the time for each individual processor to complete its piece of the computation. (Again, we amortize away set-up time by assuming that many sequential iterations of this computation are required.)

Therefore, in order to show a reversible advantage for time-efficiency, we must consider a different class of sustainable computations, namely one that requires frequent communication between processing elements. This will imply that processing elements cannot be spread arbitrarily far apart without adversely affecting the time for the computation (due to the lightspeed limit). The requirement for a relatively compact structure will then lead to a tradeoff between entropy generation and speed which, as we will show, will favor the reversible machines.

Fortunately, many real computations of interest are indeed of the sort that requires frequent communication. Our canonical example will be the simulation of physical systems; in particular, reversible 3-dimensional lattice simulations (*cf.* [119, 164, 117]; [117] contains many more references). In such computations, each update of a computational cell depends on the results of the updates of its nearest neighbors from the previous time step.

6.2.3.1 Time for 3-D local array simulations

Irreversible time. There is a simple proof of a lower bound on the average time per step for performing 3-D local array computations on an FIA. Consider the problem of simulating a locally-connected $N_{\mathrm{D}} \times N_{\mathrm{D}} \times N_{\mathrm{D}}$ array of cells for a number of steps $N_{\mathrm{st}} \gg N_{\mathrm{D}}$. Consider a segment of this computation consisting of a series of $\mathbf{\Theta}(N_{\mathrm{D}})$ consecutive steps. An element's value at the end of this segment will in general depend on the values (at the start of the segment) of all the elements less than $\mathbf{\Theta}(N_{\mathrm{D}})$ positions away from it, that is, $\mathbf{\Theta}(N_{\mathrm{D}}^3)$ different elements, and on the results of $\mathbf{\Theta}(N_{\mathrm{D}})$ updates of those elements, for a total of $\mathbf{\Theta}(N_{\mathrm{D}}^4)$ operations involved in determining the final value.

If the series of steps is performed within a time $\mathcal{T}$, then all those $\mathbf{\Theta}(N_{\mathrm{D}}^4)$ operations must occur within a sphere of radius $R = c\mathcal{T} \sim \mathcal{T}$ of the final result, in order to possibly affect the final result, given that information propagates no faster than light. This sphere is contained within a surface of area $A \sim \mathcal{T}^2$. By the arguments in §6.2.2.1, the maximum rate $\mathcal{R}_{\mathrm{i}}$ of fully irreversible computation that can be sustained within this region is then bounded by $\mathcal{O}(A) \sim \mathcal{T}^2$. (We care about the sustainable rate because the block of N_{D} steps in question is performed in series with $N_{\mathrm{st}}/N_{\mathrm{D}} \gg 1$ other similar segments operating over the same cells.)

Running at the rate $\mathcal{R}_{\mathrm{i}} \precsim \mathcal{T}^2$ for time $\mathcal{T}$ means that only $N_{\mathrm{ops}} \precsim \mathcal{T}^3$ total operations affecting the result can be performed within that time. For this N_{ops} to be equal to the needed $\mathbf{\Theta}(N_{\mathrm{D}}^4)$, $\mathcal{T}$ must then be $\mathbf{\Omega}(N_{\mathrm{D}}^{4/3})$. If it takes $\mathbf{\Omega}(N_{\mathrm{D}}^{4/3})$ time to perform $\mathbf{\Theta}(N_{\mathrm{D}})$ steps, then the average time per step is $t_{\mathrm{op}} \succsim N_{\mathrm{D}}^{1/3}$.

If we assume that some means is available for ballistic constant-speed communication between neighboring processors over arbitrary distances, then this bound can actually be achieved, using, for example, an array of $N_{\mathrm{D}} \times N_{\mathrm{D}} \times N_{\mathrm{D}}$ processing elements spaced a distance of $\mathbf{\Theta}(N_{\mathrm{D}}^{1/3})$ apart from their neighbors, each updating its cell once every $t_{\mathrm{op}} \sim N_{\mathrm{D}}^{1/3}$ time units, and spending the $\mathbf{\Theta}(N_{\mathrm{D}}^{1/3})$ time before its next update exchanging results with its neighbors. See figure 6.2.

Each processor produces $S = \mathbf{\Theta}(1)$ entropy per step, so a single column of N_{D} processors produces entropy at the rate $S/t_{\mathrm{op}} \sim N_{\mathrm{D}}^{2/3}$. Fortunately, the cross-sectional area of the column is $\mathbf{\Theta}(N_{\mathrm{D}}^{1/3}) \times \mathbf{\Theta}(N_{\mathrm{D}}^{1/3}) \sim N_{\mathrm{D}}^{2/3}$ and so the flow of entropy can move along the column with no more than constant flux. And if it can be moved ballistically, no additional entropy is generated by this flow.

Ballistic inter-processor communication and ballistic entropy transport seem to be reasonable assumptions because they are very closely approximated by, for example, propagation of photons or information-carrying matter through vacuum, and by propagation of electrons through superconductors.

Ballistic computation, in contrast, may well be more difficult because the need for frequent interactions between information-carrying components may sap energy or introduce exponentially-increasing error; these issues would need to be addressed to build a substantially ballistic computational system. But for purposes of communication only, no interactions need occur during flight, and so those particular problems do not arise.

In any case, it seems a reasonable approximation to conclude that a time per step of $\mathbf{\Theta}(N_{\mathrm{D}}^{1/3})$ for simulation of diameter-N_{D} 3-d arrays can actually be achieved on fully irreversible machines. Can we beat this when running in a time-proportionate reversible fashion?

Reversible time. The answer is yes. Consider a TPRA implementation using a similar $N_{\mathrm{D}} \times N_{\mathrm{D}} \times N_{\mathrm{D}}$ array. This time, spread the processors only $\ell = \mathbf{\Theta}(N_{\mathrm{D}}^{1/4})$ distance apart from their neighbors, and let them take $t_{\mathrm{op}} \sim \ell$ time for each update computation. (See fig. 6.3.) Then the entropy generated per update is $S \sim 1/t_{\mathrm{op}} \sim N_{\mathrm{D}}^{-1/4}$, and the rate of entropy generation per processor is $\mathcal{R}_{\mathrm{S}} = S/t_{\mathrm{op}} \sim 1/t_{\mathrm{op}}^2 \sim (N_{\mathrm{D}}^{-1/4})^2 = N_{\mathrm{D}}^{-1/2}$. Thus the rate of entropy generation for a column of N_{D} processors is $\mathbf{\Theta}(N_{\mathrm{D}} \cdot N_{\mathrm{D}}^{-1/2}) \sim N_{\mathrm{D}}^{1/2}$. The cross-sectional area of the column is $\mathbf{\Theta}(N_{\mathrm{D}}^{1/4}) \times \mathbf{\Theta}(N_{\mathrm{D}}^{1/4}) \sim N_{\mathrm{D}}^{1/2}$, so this rate of entropy generation is sustainable, and the time per step of $\mathbf{\Theta}(N_{\mathrm{D}}^{1/4})$ is not prevented by entropy removal.

We can show that this asymptotic time of $N_{\mathrm{D}}^{1/4}$ is minimal for a TPRA, just as $N_{\mathrm{D}}^{1/3}$ was minimal in the irreversible case. Suppose the average time per step in a sustained TPRA implementation is t_{op}. The average entropy generated per op is then $S \gtrsim 1/t_{\mathrm{op}}$. Performing the $\mathbf{\Theta}(N_{\mathrm{D}}^4)$ operations that affect a cell during an N_{D}-step computation then generates $\mathbf{\Omega}(N_{\mathrm{D}}^4/t_{\mathrm{op}})$ entropy, and since the N_{D} steps take exactly time $\mathcal{T} = t_{\mathrm{op}}N_{\mathrm{D}}$, the average rate of entropy generation is $\mathbf{\Omega}(N_{\mathrm{D}}^3/t_{\mathrm{op}}^2)$. Suppose $t_{\mathrm{op}} \prec N_{\mathrm{D}}^{1/4}$: then the rate $\mathcal{R}_{\mathrm{S}}$ of entropy generation would be $\mathbf{\Omega}(N_{\mathrm{D}}^3/\mathbf{o}(N_{\mathrm{D}}^{1/4})^2) \succ$

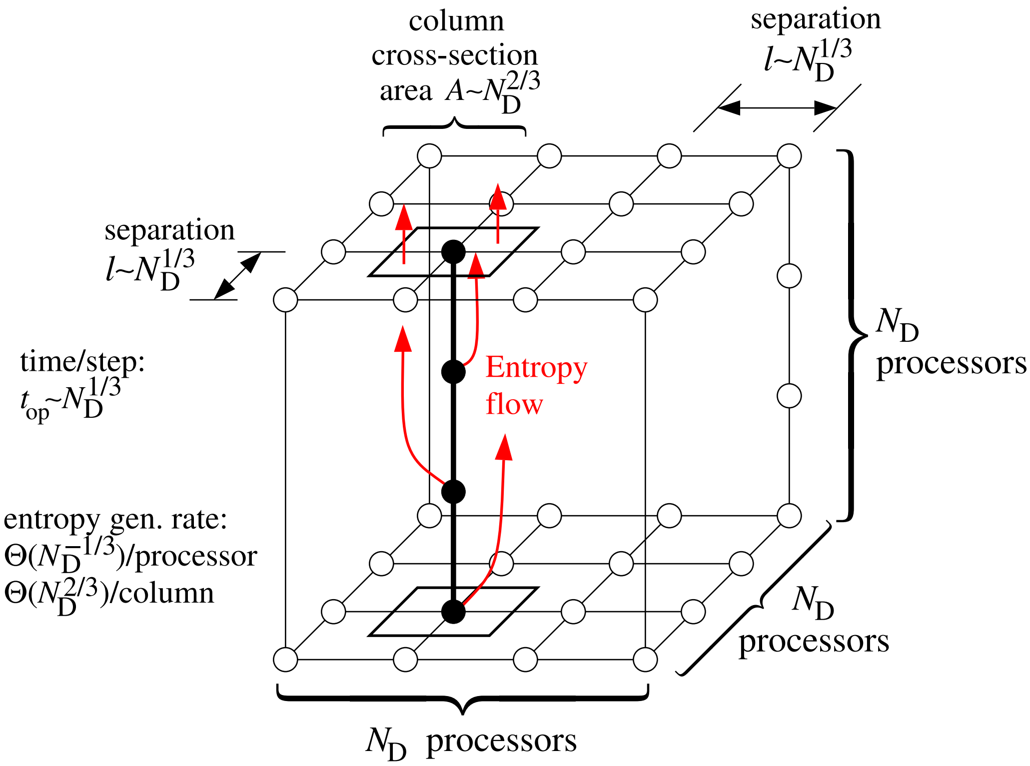


Figure 6.2: A machine configuration that achieves the asymptotically optimal FIA performance of $\mathbf{\Theta}(N_{\mathrm{D}}^{1/3})$ time per step on 3-D local cell-array simulations. The top and bottom layers of a locally-connected $N_{\mathrm{D}} \times N_{\mathrm{D}} \times N_{\mathrm{D}}$ mesh of processors are shown, and a single column of processors through the machine is emphasized in black. Spacing the processors $\mathbf{\Theta}(N_{\mathrm{D}}^{1/3})$ apart gives enough room for the entropy produced by the column to be removed with no more than the maximum achievable flux $F_{\mathrm{S}} = \mathbf{\Theta}(1)$, while still allowing neighbors to communicate with each other within $\mathbf{\Theta}(1)$ steps. Closer spacing would increase the time for entropy removal; sparser spacing would increase the communication time.

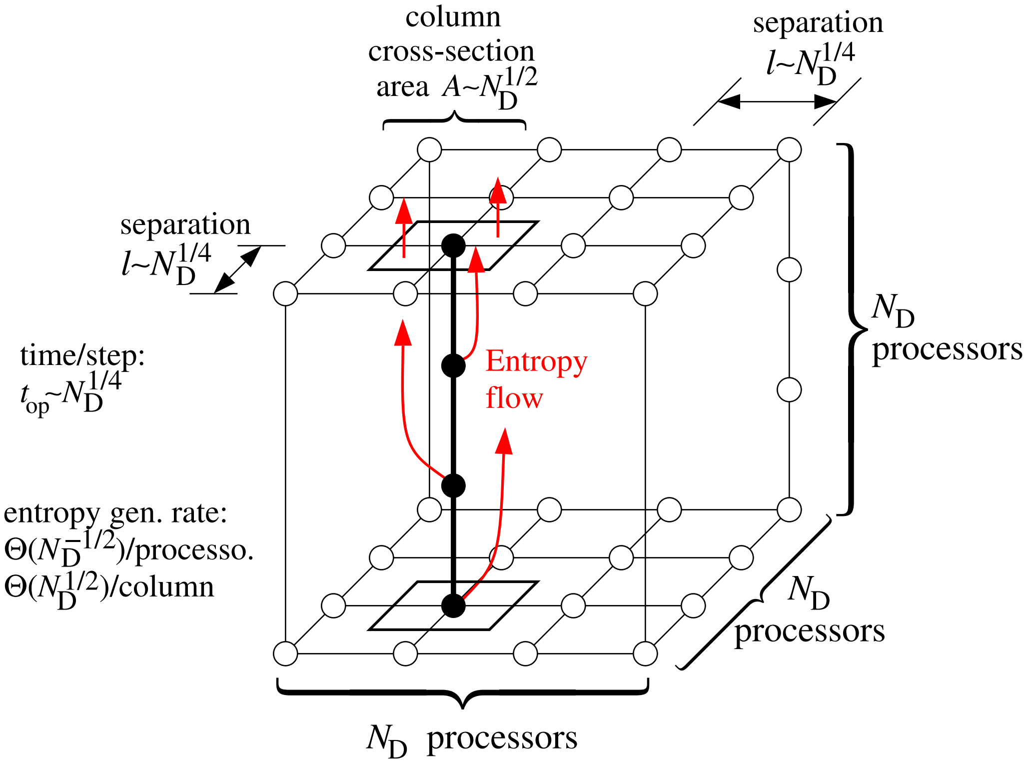


Figure 6.3: A TPRA configuration that is asymptotically strictly faster than the fastest FIA (fig. 6.2) for 3-D simulations of reversible locally-connected cell arrays. The speedup is possible because the lower TPRA entropy per operation, $\boldsymbol{\Theta}(1/t_{\mathrm{op}})$, permits the processors to be packed closer together, and run at a correspondingly faster rate, without the fixed maximum entropy flux F_{S} being exceeded. An inter-neighbor spacing and time per step of $\boldsymbol{\Theta}(N_{\mathrm{D}}^{1/4})$ is optimal among TPRA structures. In contrast, an idealized, perfectly ballistic machine, generating no entropy, could achieve $\boldsymbol{\Theta}(1)$ time per step.

$N_D^3 \cdot N_D^{-1/2} \sim N_D^{5/2}$, that is, $\mathcal{R}_S \succ N_D^{5/2}$. But if $t_{op} \prec N_D^{1/4}$, then the total time $t_{op}N_D \prec N_D^{5/4}$, and so the operations must be performed within a sphere of radius $R \prec N_D^{5/4}$, which has area $A \sim R^2 \prec N_D^{5/2}$. Supporting a sustained rate of entropy generation of $\mathcal{R}_S \succ N_D^{5/2}$ within an area $A \prec N_D^{5/2}$ would require an average entropy flux $F_S = \mathcal{R}_S/A \succ 1$, which violates our basic technological assumption of a fixed upper bound on entropy flux. Therefore an average time per step $t_{op} \prec N_D^{1/4}$ on this problem is actually not possible for a TPRA.

Therefore, for this class of computations, relevant to simulation of physical systems, a time-proportional reversible machine is faster than a fully irreversible machine by a factor of exactly $\Theta(N_D^{1/3})/\Theta(N_D^{1/4}) \sim N_D^{1/12}$. In terms of $\$_r \sim \mathcal{T}$, the reversible advantage is $\mathcal{A}_r \sim \$_r^{1/3}$.

Ballistic time. This situation is trivial. The ballistic machine produces no entropy to remove, so N_D^3 processing elements can just be packed together with minimal separation no matter what the value of N_D, and so the communication time and the time per step can be made constant, independent of N_D.

6.2.3.2 Time cost with non-local communication

In section 6.2.3.1 we saw that on 3-D array simulations with local communication, reversible machines were faster than irreversible machines by a factor of $\Theta(N_D^{1/12})$ where N_D was the number of elements across the array in each dimension. Are there other kinds of problems where the reversible advantage is greater as a function of N_D? What problems have the highest asymptotic reversible advantage as a function of N_D?

One idea to try to improve the reversible advantage is to pose a problem that requires non-local communication between cells, to try to force the machines to be more compact, giving the reversible machine more of an advantage. For an array of cells of diameter $\Theta(N_D)$, obviously the farthest we can require a signal to travel before being processed is $\Theta(N_D)$ inter-cell distances. However, if we make this logical communication distance be as large as $\Theta(N_D)$, then reversibility will confer *no* speed advantage, because the communication time $\Theta(N_D)$ will be sufficient for all entropy to be removed even from the irreversible machine in the most compact configuration! So the required communication distance must actually be $\mathbf{o}(N_D)$ if we are to achieve any reversible advantage.

An analysis (not detailed here) indicates that the optimal scaling relation d_c between logical communication distance (distance in terms of cells) and array size to achieve maximal reversible advantage is $d_c \sim N_D^{1/2}$. For this problem, the optimal configuration for the irreversible machine turns out to be with distance $\Theta(N_D^{1/6})$ between processors, which gives a minimum time per step of $\Theta(N_D^{2/3})$; the optimal

TPRA and BRA are both packed at fixed density and run with a time per step of $\Theta(N_{\mathrm{D}}^{1/2})$, that is, $\Theta(N_{\mathrm{D}}^{1/6})$ times faster than the irreversible machine. In terms of $\$_{\mathrm{r}} \sim \mathcal{T}$, the reversible advantage is $\mathcal{A}_{\mathrm{r}} \sim \$_{\mathrm{r}}^{1/3}$. This appears to be the maximum speedup possible using time-proportionate reversibility. Still, using the advanced technologies mentioned in chapter 8, we expect that even this rather slow scaling is sufficient to yield significant speedups for reversible machines over irreversible ones at reasonable scales. (However, more detailed analysis is needed.)

Now, as we already discussed in 3.2.2.2, we generally cannot assume that time complexity alone is a good measure of cost. Let us now see what happens to our scaling arguments when we factor in other components of cost as well.

6.2.4 Spacetime cost

The results derived above for the minimum time for array situations might at first appear to be inapplicable to the problem of minimizing the spacetime product, since many of our solutions involved spreading neighboring processors apart with ever-increasing distances between them; the total volume of the computer must thus be enormous!

However, this is actually not the case: all the machines discussed above can be easily converted to configurations in which the total volume of the machine scales no faster than the volume just to store the data being manipulated.

The way this is done is by simply *folding up* each column of processors (normally aligned parallel to the entropy flow) to fill up the entire $\ell \times \ell$ area available between the neighboring columns, thus reducing the thickness of the machine in the direction of entropy flow, to the point where the machine has some fixed density independent of scaling. (See fig. 6.4.)

This transformation changes nothing in our earlier analyses; nothing prevents operation exactly the same as before. We have one additional construction requirement, however; namely that throughout each period that is reserved for signal propagation, each processing element must vacate the paths across the plane (perpendicular to entropy flow) along which interprocessor signals propagate, so as not to the impede the ballistic propagation of signals to and from the processors in neighboring columns.

Therefore, our solutions from the previous section, so reconfigured, optimize volume as well as time, and thus also optimize their product. So for a given problem size, reversibility provides the same asymptotic benefits for spacetime cost (namely, $\mathcal{A}_{\mathrm{r}} \sim N_{\mathrm{D}}^{1/6}$) as it does for time cost alone. Expressed in terms of the number of processors or volume $N_{\mathrm{proc}} \sim \mathcal{V} \sim N_{\mathrm{D}}^3$, we have $\mathcal{A}_{\mathrm{r}} \sim N_{\mathrm{proc}}^{1/18}$. Expressed in terms of the spacetime cost $\$_{\mathrm{r}} = \mathcal{V}\mathcal{T} \sim N_{\mathrm{D}}^3 \cdot N_{\mathrm{D}}^{1/2} \sim N_{\mathrm{D}}^{7/2}$ on the reversible machine, the advantage is $\mathcal{A}_{\mathrm{r}} \sim N_{\mathrm{D}}^{1/6} \sim (\$_{\mathrm{r}}^{2/7})^{1/6} = \$_{\mathrm{r}}^{1/21}$.

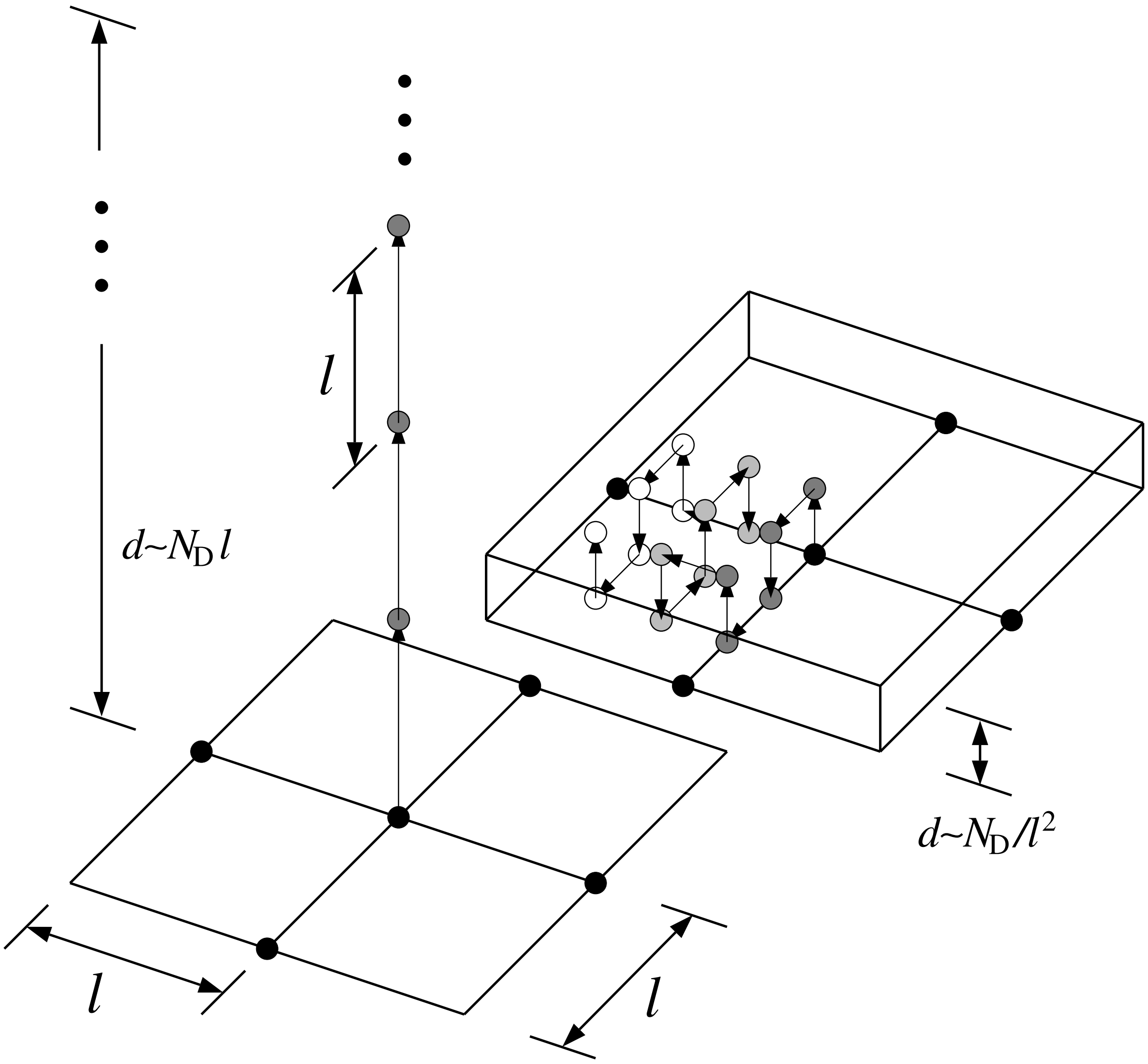


Figure 6.4: How to "fold up" a column of processors to convert a space-inefficient mesh into another structure with the same asymptotic speed but minimum volume. Initially a column of N_D (here, 18) processors extends straight up through the machine (full height not shown) from its lowest plane. We take this column, and fold it up at maximum density within the $\ell \times \ell$ area between it and its neighboring columns. The entropy flux through that area does not increase, nor does the distance between any two logically neighboring processors. (Indeed it decreases for neighbors in the same column.) But the thickness d of the machine is decreased by a factor of $\mathbf{\Theta}(\ell^3)$, from $\mathbf{\Theta}(N_\mathrm{D}\ell)$ to $\mathbf{\Theta}(N_\mathrm{D}/\ell^2)$; and the volume decreases by the same factor.

6.2.5 Mass-time product

For the array-simulation problem, the mass of our solutions scaled no faster than the mass necessary just to represent the raw information in the array, so our solutions optimized the mass-time product as well. Thus reversibility gives at least the same advantage for mass-time product efficiency as well.

Together with our observations above about spacetime cost, reversibility also minimizes the combined cost measure $(M + \mathcal{V})\mathcal{T}$.

6.2.6 (Area + mass) × time

This case, too, is identical, because for the machines discussed above, $A + M \sim M$ in all machine configurations discussed. The minimum area scaled no faster than the mass, and so for sufficiently large problems made at most a mass-proportionate contribution to the total cost.

6.2.7 Entropy + mass-time

For problems where no communication is required between processors, the mass-time product is proportional to the number of operations, and the entropy production never grows faster than this anyway, so the cost in all models reduces to $\mathbf{\Theta}(N_{\mathrm{ops}})$.

For problems such as the array simulations where communication is required, again the total cost is dominated by the mass-time cost in all cases, so reversibility again improves efficiency by the same factor of $\mathbf{\Theta}(N_{\mathrm{proc}}^{1/18})$ in the $\sqrt{N_{\mathrm{D}}}$ communication case.

Similarly, we get the same asymptotic results for the comprehensive cost measure $\$_{\mathrm{c}} = S + (A + M + \mathcal{V})T$ from p. 58. (We drop the integral here because we are considering problems where the resource usage does not change significantly over time.)

6.3 Generalizing the results

The scaling results of the previous section were derived under the assumption that the computational task being performed was one that inherently required only reversible operations, so that time-proportionally reversible operations could be considered equivalent in power to logically irreversible operations.

We also depended on the computation being parallelizable, and in the more sophisticated cost measures, we depended on a requirement for frequent communication between relatively nearby processors, and on the absence of a requirement for frequent communication between arbitrarily distant processors. An example of such a computation is the simulation of a spatially and temporally discretized reversible physical system.

Given all these assumptions, just how general are the reversible scaling advantages? Do they cover very many practical applications in large problem classes, other than just physical simulations?

The complete answer to this question is uncertain, but one observation is that Bennett's 1989 algorithm [19, 103] can be utilized to remove the requirement for the *reversibility* of the underlying task, while still permitting almost the same polynomial speedups and cost-efficiency benefits. (However, the assumptions regarding parallelizability and communication requirements remain.)

6.3.1 Speedups for irreversible computations on reversible machines

Bennett's technique [19] allows one to transform a logically irreversible algorithm that requires S memory cells ("space") and T primitive operations ("time") into a reversible algorithm that leaves behind no garbage information (other than input and output) and takes $\mathsf{T}' \sim \mathsf{T}(\mathsf{T}/\mathsf{S})^{\varepsilon}$ operations, and $\mathsf{S}' \sim \mathsf{S}\log(\mathsf{T}/\mathsf{S})$ memory, for any $\varepsilon > 0$. (See Levine & Sherman 1990 [103] for the derivation.)

A finite irreversible processing element running for N_{st} steps performs $\mathsf{T} \sim N_{\mathrm{st}}$ operations, using $\mathsf{S} \sim 1$ space. Therefore, using Bennett's algorithm, such a run can be simulated reversibly with $\mathsf{T}' \sim N_{\mathrm{st}}^{1+\varepsilon}$, $\mathsf{S}' \sim \log N_{\mathrm{st}}$, accumulating no garbage except for the input, that is, the state of the simulated processor prior to the run.

If we then irreversibly erase this $\Theta(1)$-size input, we generate $\Theta(1)$ entropy, and we can proceed to simulate arbitrarily many consecutive blocks of N_{st} steps in this way, with an average entropy generation per reversible operation of $S_{\mathrm{op}} \sim 1/N_{\mathrm{st}}^{1+\varepsilon}$, and a memory requirement of only $\mathsf{S}' \sim \log N_{\mathrm{st}}$, which is constant in the number of *blocks* of N_{st} steps that are simulated.

If we wish this algorithm to be time-proportionately reversible, the average entropy generated per operation must be $\mathcal{O}(1/t_{\mathrm{op}})$. So we must have $N_{\mathrm{st}}^{1+\varepsilon} \succsim t_{\mathrm{op}}$, or $N_{\mathrm{st}} \succsim t_{\mathrm{op}}^{1/(1+\varepsilon)}$. With the smallest choice, $N_{\mathrm{st}} \sim t_{\mathrm{op}}^{1/(1+\varepsilon)}$, the memory requirement of this algorithm then scales as $\mathsf{S} \sim \log t_{\mathrm{op}}^{1/(1+\varepsilon)} \sim \log t_{\mathrm{op}}$, given constant ε.

By running this algorithm simultaneously on a 3-D array of reversible processors of memory $\Theta(\log t_{\mathrm{op}})$ each, we can sustainably simulate an entire 3-D array of fixed-size irreversible processors in TPRA fashion. Furthermore, we saw in §6.2.3.1 that a 3-D TPRA can run with $t_{\mathrm{op}} \sim N_{\mathrm{D}}^{1/4}$, so a memory per processor of $\mathsf{S} \sim \log N_{\mathrm{D}}^{1/4} \sim \log N_{\mathrm{D}}$ will suffice. Given that the computer must fit within the finite observable universe, $\log N_{\mathrm{D}}$ is bounded by a reasonably small constant, so we may approximate S as

$\mathbf{\Theta}(1)$ for all practical purposes. (Although this is cheating from a pure theoretical perspective.)

With $t_{\rm op} \sim N_{\rm D}^{1/4}$, we have that $N_{\rm st} \sim N_{\rm D}^{1/[4(1+\varepsilon)]}$, and the Bennett simulation of this many steps takes $N_{\rm ops} \sim N_{\rm st}^{1+\varepsilon} \sim [t_{\rm op}^{1/(1+\varepsilon)}]^{1+\varepsilon} = t_{\rm op} \sim N_{\rm D}^{1/4}$ reversible operations, for an average real time, per irreversible step simulated, of

$$\begin{aligned}\mathcal{T} &= t_{\rm op} N_{\rm ops}/N_{\rm st} \\ &\sim N_{\rm D}^{1/4} N_{\rm D}^{1/4} / N_{\rm D}^{1/[4(1+\varepsilon)]} \\ &= N_{\rm D}^{\frac{1}{2} - \frac{1}{4(1+\varepsilon)}} \\ &= N_{\rm D}^{\frac{1}{4}\left(\frac{1+2\varepsilon}{1+\varepsilon}\right)} \\ &= N_{\rm D}^{\frac{1}{4}(1+\varepsilon')} \qquad (6.1)\end{aligned}$$

where $\varepsilon' = \varepsilon/(1+\varepsilon)$. The exponent of $N_{\rm D}$ in eq. (6.1) can be made as close to $1/4$ as desired, by taking ε close to 0.

In contrast, as we saw earlier, the 3-D irreversible array being simulated must itself take at least $\mathbf{\Omega}(N_{\rm D}^{1/3})$ time per step. So a reversible machine can simulate even an *irreversible* 3-D array faster than that array can run by itself! This improved asymptotic speed also leads to improved asymptotic cost-efficiency by the various other measures we have covered. Moreover, the reversible advantages can become arbitrarily asymptotically close to those we calculated in the previous section for the case of simulating 3-D reversible systems.

However, as pointed out by Levine and Sherman [103], one must be careful when using Bennett's algorithm not to take ε *too* close to zero, because the constant factor in the memory requirement increases *exponentially* in $1/\varepsilon$, specifically as $\varepsilon 2^{1/\varepsilon}$. But to beat the irreversible 3-D array's asymptotic performance, we only require $\varepsilon < 1/2$, so the constant factor increase in memory size due to the choice of ε only needs to be more than 2.

The upshot of all this is that, apparently, for *any* class of computations that are sufficiently parallelizable and require the right amount of communication, a TPRA reversible machine family, such as the R3M, can perform that class of computations strictly faster, asymptotically, than any FIA machine family. The class of computations in question does *not* have to be "inherently" reversible in order for this to be true.

6.4 Summary of scaling results

We conclude this chapter with a summary of our discoveries about the asymptotic scaling advantages that can be gained by the use of time-proportionate reversibility.

		Cost in each model			Advantage factors	
Cost measure	Task type	FIA ($\$_\mathrm{i}$)	TPRA ($\$_\mathrm{r}$)	BRA ($\$_\mathrm{b}$)	Reversible $\mathcal{A}_\mathrm{r} = \$_\mathrm{i}/\$_\mathrm{r}$	Ballistic $\mathcal{A}_\mathrm{b} = \$_\mathrm{i}/\$_\mathrm{b}$
$\$ = S$	any	N_ops	n_in	n_in	$N_\mathrm{ops}/n_\mathrm{in}$	$N_\mathrm{ops}/n_\mathrm{in}$
$\$ = \mathcal{T}$	no comm.	1	1	1	1	1
	local comm.	$N_\mathrm{D}^{1/3}$	$N_\mathrm{D}^{1/4}$	1	$N_\mathrm{D}^{1/12}, \$_\mathrm{r}^{1/3}$	$N_\mathrm{D}^{1/3}$
	$\sqrt{N_\mathrm{D}}$ comm.	$N_\mathrm{D}^{2/3}$	$N_\mathrm{D}^{1/2}$	$N_\mathrm{D}^{1/2}$	$N_\mathrm{D}^{1/6}, \$_\mathrm{r}^{1/3}$	$N_\mathrm{D}^{1/6}, \$_\mathrm{b}^{1/3}$
$\$ = A\mathcal{T}$	no comm.	N_ops	$N_\mathrm{ops}^{5/6}$	$N_\mathrm{ops}^{2/3}$	$N_\mathrm{ops}^{1/6}, \$_\mathrm{r}^{1/5}$	$N_\mathrm{ops}^{1/3}, \$_\mathrm{b}^{1/2}$
$\$ = \mathcal{V}\mathcal{T}$,	no comm.	N_D^3	N_D^3	N_D^3	1	1
$M\mathcal{T}$,	local comm.	$N_\mathrm{D}^{3+1/3}$	$N_\mathrm{D}^{3+1/4}$	N_D^3	$N_\mathrm{D}^{1/12}, \$_\mathrm{r}^{1/39}$	$N_\mathrm{D}^{1/3}, \$_\mathrm{b}^{1/9}$
$\ldots, \$_\mathrm{c}$	$\sqrt{N_\mathrm{D}}$ comm.	$N_\mathrm{D}^{3+2/3}$	$N_\mathrm{D}^{3+1/2}$	$N_\mathrm{D}^{3+1/2}$	$N_\mathrm{D}^{1/6}, \$_\mathrm{r}^{1/21}$	$N_\mathrm{D}^{1/6}, \$_\mathrm{b}^{1/21}$

Table 6.2: Summary of asymptotic scaling results for reversible versus irreversible machines. The first column indicates the type of cost measure being used; S being entropy, A surface area, $\mathcal{T}$ real time, $\mathcal{V}$ physical volume, M total gravitating mass, $\$_\mathrm{c}$ the comprehensive cost measure from p. 58. The second column indicates restrictions on the type of computational task for which the results hold, in particular on the communication involved. The quantity N_D refers to the number of elements along each dimension of a 3-D array of finite-state cells.

Under the most comprehensive cost measures, such as $\$_\mathrm{c}$, the reversible advantage $\mathcal{A}_\mathrm{r}$ can scale with factors as high as the 21st root of the reversible cost (18th root of physical machine size), but no more than that. With fully ballistic machines, the comprehensive advantage (in the local communication case) would scale better, $\mathbf{\Theta}(\$_\mathrm{b}^{1/9})$.

See table 6.2.

The best asymptotic cost-efficiency advantage for reversible machines is of course gained in the case where total entropy generation is the sole measure of cost. The ratio between irreversible and reversible entropy costs in this case may be an arbitrarily fast-growing function of problem size or reversible entropy cost. But this cost measure is not very satisfying because it ignores the time taken and the opportunity cost due to the temporary use of other resources (A, $\mathcal{V}$, M).

In contrast, considering time costs alone gives a rate of growth for the reversible-to-irreversible speed ratio that, for suitable problem classes, is limited to at most the cube root of the reversible time cost. Thus, a computation on a TPRA machine (such as the R3M of 5.4.1, p. 116) that takes time $\mathcal{T}$ will in some cases require as much as $\Omega(\mathcal{T} \cdot \mathcal{T}^{1/3})$ time in the *fastest possible* fully irreversible (FIA) implementation. In other words, in terms of speed, the class of architectures that permits time-proportional reversible operation strictly dominates the class of architectures that does not.

Of course, in general, time alone is not the only factor in the cost of computation, so we also studied the case where various measures of machine size that influence "rental cost" were included as well. With surface area as the size measure, the best reversible advantage $\mathcal{A}_{\mathrm{r}}$ grows as the 5th root of cost, or the 4th root of area, or the square root of diameter. When mass and/or volume are included as components of the machine size, the best reversible advantage scales as the 21st root of total cost, or 18th root of the machine's mass or volume. This advantage occurs in computations in which communication distances are proportional to the square root of the logical diameter of the machine.

Such scaling may not appear to be very significant, but we estimate that the constant factors work out so that even given the relatively poor performance of the reversible logic devices available today (which we will discuss in the next chapter), at the extremely high end of machines buildable with current technology, reversible operation is apparently required for optimal efficiency.

Such a machine would be rather large and very expensive (we estimate tens of billions of dollars), but as the underlying device technology improves, machines that gain a cost-efficiency advantage through reversibility will become buildable at lower and lower cost levels. In chapter 8 we will show that if certain proposals for future reversible logic devices work as predicted, any computer larger than about a micron in diameter will require reversibility in order to achieve optimal efficiency.