

Hot Data Centers vs. Cool Peers

Sergiu Nedeveschi[†], Sylvia Ratnasamy[†] and Jitu Padhye[‡]

[†]Intel Reserch, [‡]Microsoft Research

1 Introduction

Most networked applications use either a largely centralized architecture (e.g. iTunes) or a p2p architecture (e.g. BitTorrent). The popularity of centralized Internet applications such as Search and web portals has fueled the growth of large data centers. Modeling and minimizing the power consumption of large data centers is the hot new research area (pun intended).

However, no attention has been paid to the power consumption of p2p systems. This despite the fact that programs like BitTorrent account for up to 95% of Internet traffic today. There are two main reasons why the power consumption of p2p applications has not yet been studied. First, these applications do not consume a large amount of power at a single location. Second, no central entity pays for the power consumed by applications like BitTorrent.

Yet, it is undeniable that systems like BitTorrent do consume power, and quite likely, a lot of power. End hosts consume more power as they do more work and routers consume more power as they route additional traffic [3]. Thus one may wonder: is it more energy efficient to download a song from BitTorrent rather than iTunes?

To answer such questions, one needs to build models that can predict the power consumption of p2p applications. In this paper, we present an outline of a model that allows us to compare the energy consumed by a p2p system to its centralized counterpart. Note that it is not our aim to advocate one system architecture over another. Many issues such as manageability, reliability and ease of deployment must be taken into account when making high-level architectural decisions. Indeed, it may be entirely infeasible to deploy applications such as web search in a p2p manner.

Thus, this model is best viewed as an exercise driven primarily by academic curiosity. And yet, this exercise is not entirely without practical interest. Today energy costs represents a significant component of the capital and operational costs of data centers, and hence it is important to understand whether, and when, alternate system architectures might prove more cost effective. For example, perhaps in controlled environments such as enterprise networks it is reasonable to harness user desktops to take on some of the tasks traditionally assigned to server rooms and data centers? More broadly, models like this help us understand the energy impact of the different design decisions that go into building a large scale system. For example, our model shows that numerous factors such as power consumption of routers, whether the application is communication *vs.* computation heavy and the efficiency of the protocol design affect the overall power consumption.

To the best of our knowledge, this is the first attempt to model the energy consumption of complex networked

systems in their entirety. We hope our initial exploration spurs further research on this topic.

2 Background

Before we formalize the details of the model, we discuss some of the overarching factors that impact the power consumption of p2p systems and data centers.

P2P systems: Several studies have shown that end-systems such as enterprise desktops, home PCs and laptops typically spend a significant portion of the day fully powered-up (*i.e.*, neither turned off, nor in low-power hibernation). Moreover, the average utilization of these machines is typically very low. Similar to prior studies, in a recent measurement we performed at 300+ enterprise hosts, we found an average “fully on” time of 14.4 hours/day per machine, with an average utilization below 5%. In this paper, we assume that a p2p application makes use of such already-on, but underutilized, end-systems. In other words, we assume that only peers that are (for whatever reason) already powered up will participate in a p2p service and that users will *not* leave their machines powered on for the sole purpose of participating in the p2p service.¹ This assumption will play a key role in the power consumption we attribute to a p2p architecture.

Needless to say, the use of peers has certain drawbacks as well. The primary overhead is that p2p systems make heavier use of the network, by generating overhead traffic for tasks such as membership maintenance and peer discovery. Furthermore, peers are at the edges of the network, and the average path length between two peers is likely to be more than the average path length from the client to the central server.

Data Centers: Two factors reduce the energy efficiency of data centers. First, because data centers consolidate thousands of computers in a single facility, cooling becomes a gargantuan issue. Several studies report that as much as 50% [2, 4] of the power consumed by today’s data centers is spent on cooling. By contrast, cooling is less of an issue for more decentralized systems such as p2p systems. For example, in its heyday, Napster routinely supported upto 25 million users with no special concern for cooling simply because its “servers” were geographically dispersed. And while one may argue that enterprise and homes may also require air-conditioning systems, we note that this need arises regardless of the p2p service. *I.e.*, for an enterprise, the presence of a large number of employees (and their machines) necessitates cooling whether or not their machines participate in p2p applica-

¹Strategies that incentivize users to cut down on the up-time of their machines will admittedly diminish the pool of potential peers in which case we might have to revisit this assumption.

tions. Moreover, since we assume a p2p application only uses machines that were already fully powered on, the *additional* cooling, if any, required to run a p2p application is negligible.

Another factor is the high baseline energy consumption of computers. As noted in [1], the power consumed by a computer is roughly linear in its CPU utilization, *with an offset*. This offset represents the power required simply to have the machine powered up and ready to process work – we term this the *baseline* energy consumption. Several empirical studies [3, 1, 4] reveal that this baseline consumption typically dominates the incremental consumption due to increased utilization.

In a data center, machines exist solely to run the service(s) in question and hence the service(s) can be viewed as responsible for the entire power consumption of the servers. By contrast, if a p2p service makes use of peers that are already powered-on, then it is only responsible for the resultant *increase* in the peer’s power consumption (since the peer was already consuming at least the baseline power draw just by virtue of being fully powered on). Hence, from the standpoint of energy consumption, the p2p option may be preferable, especially when the baseline power consumption is the dominant component of a machine’s overall power consumption. This is essentially an argument for reusing machine deployments.

In the rest of the paper, we shall see that the assumptions we make about the issues mentioned above are critical in determining which architecture is more energy efficient.

3 Model components

In this paper, we focus on a simple file transfer application. Both centralized (e.g. iTunes) and p2p (e.g. BitTorrent) versions of this application are in popular use. Even for the simple file transfer application, creating a good model is a difficult exercise. First, we must decide what it means for a centralized architecture and a p2p architecture to deliver equivalent functionality. Only then can one compare their power consumption. Second, we must account for the energy consumption due to the network/routers involved. Finally, the various servers, peers and routers might all be handling more than just the particular service under evaluation and hence we must be careful in deciding what fraction of the energy consumption at (for example) a router should be attributed to the particular service we’re considering.

We now describe the key components of our model. The model contains many simplifications since our goal is to roughly estimate how various parameters influence the energy consumption in the two architectures.

Consumers of energy: There are three primary consumers of energy in the scenarios we consider: servers, peers and the routers along the path. The number of routers depends on the distribution of path lengths in the Internet. For the purpose of this paper, we assume Internet paths are linearly distributed with an average path length of d_s and d_p hops to (data-center) servers and peers respectively.² Finally, for a p2p system, we use n (≥ 1) to

denote the number of peers that would be required to service a request at the same level as a single server.

Components of energy: We divide the energy consumption in each of the above consumers – servers, peers and routers – into two components. The first is the energy consumed just from having the equipment be powered up and ready to process work – we refer to this as the *baseline* energy of the equipment. In addition, we have the energy required to actually process work; this *work-induced* energy is directly dependent on the equipment’s level of utilization when processing the offered workload. In keeping with recent empirical findings[1], we assume that the work-induced component of energy scales linearly with equipment utilization. Thus, the energy consumed by (for example) a server over time t as:

$$E = (S_{base} + (S_{max} - S_{base}) \cdot u_s) \cdot t \quad (1)$$

where S_{base} is the server’s baseline power draw, S_{max} is its power draw when serving at maximum capacity and u_s is the average server utilization. The consumption at peers and routers could be expressed similarly.

Attributing energy components: We must determine what portion of the energy consumed at each of the servers, routers and peers can be attributed to the service in question. This is where a key distinction arises between peers and servers. In the case of data center servers, since they are machines dedicated only to serving client requests, we hold the service responsible for some fraction of *both* the server’s baseline and work-induced energy consumption. The p2p architecture, however, however, makes use of peers that were already powered on, and would have been up anyway, regardless of whether or not they were processing service requests. We thus do not hold the service accountable for the baseline energy consumption at a peer and only hold it responsible for some fraction of the peer’s work-induced energy consumption.

For routers, the question of whether or not to attribute a portion of a router’s baseline consumption to our service is arguable. On the one hand, routers run regardless of our particular service but, on the other hand, routers’ baseline consumptions must eventually be accounted for and a fair way to do so would be to split the charge across all the traffic they serve. We thus consider both possibilities.

Cooling overhead: Data centers require significant amount of energy for cooling the systems. On the other hand, cooling is typically not required for p2p systems. We capture this overhead with a multiplicative factor c (≥ 1) that we apply only to server consumption. I.e., if we were to attribute an amount E_s of a server’s energy consumption to our service, then we say that cE_s is the energy consumption to be attributed when taking cooling costs into account.

P2P communication Overhead: Most p2p systems incur some additional computation and communication overhead due to various factors: the redundancy typically required to compensate for poor quality peers, protocol overheads due to communicating with multiple peers, tracking peer membership and so forth. This overhead re-

² We can incorporate more realistic models to improve accuracy.

sults in an increase in the workload seen by both peers and routers and again we model these with multiplicative factors w_p and w_r (≥ 1) that capture the overall overhead at peers and routers respectively.

4 Model

Our approach is to compute the energy consumption for a single service request under both the data-center and the p2p architectures. Let E_s , E_p and E_r denote the energy consumption due to a single request at each of a server, peer and router respectively. Then, we can express E_{dc} , the energy consumption for a single request in the data-center scenario as:

$$E_{dc} = cE_s + d_s E_r \quad (2)$$

Similarly, the energy consumption for a p2p-based system, E_{p2p} , can be expressed as:

$$E_{p2p} = n w_p E_p + n w_r d_p E_r \quad (3)$$

where n is the number of peers, d_s and d_p the path length to servers and peers and c , w_p and w_r the cooling and p2p overheads respectively.

Now, to estimate the per-request energy consumptions E_s , E_p and E_r we must decide what fraction of the baseline and work-induced energy at each of the servers, routers and peers can be attributed to our service request. Computing this is non-trivial because all three entities handle more than just this particular service – servers handle multiple client requests simultaneously, routers handle other network traffic and peers might be running additional user tasks unrelated to our service.

We proceed as follows: we assume that server and peer workloads are *linearly proportional* to the number of requests. The assumption is reasonable for moderate utilizations, especially for simple tasks like file transfer. Thus, if a request involves the transfer of an average of B bits, then we can view server (and peer) workloads as linear in the number of bits transferred over the network. We therefore introduce the notion of the energy consumed *per bit* transferred by each of servers, peers and routers and then measure the per-request consumption by multiplying the per-bit consumption with the number of bits transferred per request. Define:

δ_s , δ_p , and δ_r : these denote the *work-induced* energy consumed per additional bit transferred by a server, peer and router respectively.

γ_s , γ_p , and γ_r : these denote the *baseline* energy consumed per processed bit. For completeness, we refer to γ_p even though, as argued earlier, we set $\gamma_p = 0$.

We can now expand the energy equations (2, 3) as:

$$\begin{aligned} E_{dc} &= cE_s + d_s E_r \\ &= c(\delta_s + \gamma_s) \cdot B + d_s(\delta_r + \gamma_r)B \end{aligned} \quad (4)$$

$$\begin{aligned} E_{p2p} &= n w_p E_p + n w_r d_p E_r \\ &= n w_p(\delta_p + \gamma_p) \cdot B/n + n w_r d_p(\delta_r + \gamma_r) \cdot B/n \\ &= w_p(\delta_p + \gamma_p) \cdot B + w_r d_p(\delta_r + \gamma_r) \cdot B \end{aligned} \quad (5)$$

where B is the total number of bits transferred per

request served at a server. Note that the above assumes that the bits transferred per peer is inversely proportional to n , the number of peers. This is reasonable since we're using bits transferred as indicative of workload. While peers may see additional overheads, recall that these are reflected in the values of w_p and w_r .

We now describe how to express these per-bit *work-induced* and *baseline* consumptions in terms of measurable quantities. Define:

M_s , M_p , M_r : these denote the maximum capacity measured in *bits-per-second* for servers, peers and routers respectively. *I.e.*, this is the network transfer rate corresponding to the point at which the machine is processing the task to its fullest capability.

S_{base} and S_{max} : these denote a server's baseline power consumption and its consumption when operating at maximum capacity M_s . S_{base} is independent of workload but S_{max} will again depend on workload. For example, for a computation-intensive task, we'd expect higher levels of CPU utilization and since the CPU is a major contributor to system consumption, we'd expect a higher gap between S_{base} and S_{max} . Communication-intensive tasks tend to saturate the I/O or memory capacity first and hence see much lower CPU utilization. So we'd expect a lower gap between S_{base} and S_{max} for such workloads.

P_{base} and P_{max} , R_{base} and R_{max} : Similar to the above, these denote the baseline and max-capacity consumption for peers and routers respectively.

μ_s and μ_r : these denote the average utilization at a server and router respectively.

Given the above, the total work-induced energy consumed per second by a server operating at maximum capacity M_s can be expressed as: $S_{max} - S_{base}$, and therefore the per-bit energy is:

$$\delta_s = (S_{max} - S_{base})/M_s \quad (6)$$

Similarly, the work-induced energy per bit in peers and routers can be expressed as:

$$\delta_p = (P_{max} - P_{base})/M_p \quad (7)$$

$$\delta_r = (R_{max} - R_{base})/M_r \quad (8)$$

Recall that we do attribute a fraction of the server's baseline consumption to each service request. To compute this fraction, we consider the average bits/second handled by the server as $u_s \cdot M_s$ bps and hence compute the per-bit baseline energy consumption at the server as:

$$\gamma_s = S_{base}/(\mu_s M_s) \quad (9)$$

Since our service is not held responsible for the baseline consumption at peers, we set:

$$\gamma_p = 0 \quad (10)$$

Similarly, depending on whether or not a router's baseline consumption is to be amortized across our service requests, we can compute the per-bit baseline consumption

at routers as either:

$$\gamma_r = 0, \quad \text{or} \quad \gamma_r = R_{base}/(\mu_r M_r) \quad (11)$$

We complete the model by substituting equations (6) - (11) into equations (4,5).

5 Comparison of two architectures

We now derive values of various model parameters from empirical measurements, and use them to compare the power consumption of p2p and centralized (data center) architectures.

Router parameters – δ_r and γ_r : We derive our router parameters from a recent measurement study that reports on the power consumption of a Cisco GSR router [3]. The study reports an idle power draw $R_{base} = 750W$. In a typical configuration, the routers uses 4 cards with speeds of 2.5Gbps/card for a total capacity of 10Gbps. If we conservatively assume an average router utilization of 50% this gives us a per-bit baseline consumption of $\gamma_r = 750W/5Gbps = 150 \cdot 10^{-9}$ J/bit. When routing at 2.5Gbps, the study reports an increase of at most 20W for a per-bit energy increase of $8 \cdot 10^{-9}$ J/bit.

Server and peer parameters – δ_s , δ_p , and γ_s : To measure the per-bit energy for peers and servers, we pick two machines with typical configurations for their class. As representative of peers, we use a single-core Intel Xeon 3.0GHz desktop running Linux, equipped with 1GB RAM and a gigabit Ethernet Intel PRO/1000 NIC. To represent servers, we use an 8-core machine featuring an Intel S5000PSL Server Board, two Intel Xeon X5355 processors with four cores each, 16GB RAM and 8 Gbps Intel PRO/1000 NICs.

We measure an idle power draw of 291W for the server, and 140W for the desktop; these are the baseline power consumptions S_{base} and P_{base} respectively.

To measure work-induced power consumption at peers and servers, we repeatedly downloaded files from the two machines using a number of clients and `httperf` [5] benchmarking tool. We gradually increase the client request rate and record the maximum web server capacity (in terms of processed requests).

We measured the average power draw, network throughput and CPU utilization at maximum capacity. These measurements allow us to calculate δ_s and δ_p using Equations (6) and (7). To calculate the baseline power consumption, we use Equation (9) and assume that a server is on average utilized to 50% of its maximum capacity. Note that γ_p is 0, as per Equation (10). The results are summarized in Table 5.

Note that the per-bit server baseline energy γ_s dwarfs the work-induced energy consumption. This is because the workload is communication intensive, and the CPU utilization at maximum load is relatively low. For more CPU intensive applications, the difference in the numbers is smaller.

Other parameters – c , w_p , w_r , d_{dc} , d_{p2p} : To a first approximation, a well-managed data center has an overhead c of about 2x, which implies that for every Watt of server power, an additional Watt is consumed by the

chillers, UPSs, air handlers, pumps, etc. Indications are that for some data centers this value is as much as 3x and higher [2]. We note that this factor does not include the cost of provisioning for this cooling. We use a value of $c = 2$ in our calculations.

For the communication and workload overheads, we pick conservative estimates of $w_r = w_p = 2$ [7]. Finally, we estimate network path lengths using a recent measurement study [6] that reports average Internet path length to/from a CDN (representative of a datacenter) to be around 13 hops and average path lengths between peers to be around 15 hops. We thus set $d_{dc} = 13$ and $d_{p2p} = 15$.

We now compute the per-request energy consumption for both the p2p and data-center scenarios. In each case, we consider the consumption with and without “charging” our service for the router baseline consumption.

Not charging for router baseline consumption: $\gamma_r = 0$: Using the values from the previous section, we obtain:

$$\begin{aligned} E_{dc} &= c(\delta_s + \gamma_s) \cdot B + d_s \delta_r B \\ &= (2 \cdot (5 \cdot 10^{-9} + 673 \cdot 10^{-9}) + 13 \cdot 8 \cdot 10^{-9}) \cdot B \\ &= 1.460 \cdot 10^{-6} B \text{ (J)} \end{aligned} \quad (12)$$

and:

$$\begin{aligned} E_{p2p} &= w_p \delta_p \cdot B + w_r d_p \delta_r \cdot B \\ &= (2 \cdot 16.2 \cdot 10^{-9} + 2 \cdot 15 \cdot 8 \cdot 10^{-9}) \cdot B \\ &= 0.272 \cdot 10^{-6} B \text{ (J)} \end{aligned} \quad (13)$$

We see that, even though the energy spent within the network path is larger for the p2p scenario, the baseline energy consumption of servers proves to be the dominant factor leading the data center scenario to a higher overall consumption.

Charging for router baseline consumption:

In this case, we find:

$$\begin{aligned} E_{dc} &= 3.140 \cdot 10^{-6} B \text{ (J)} \\ E_{p2p} &= 4.762 \cdot 10^{-6} B \text{ (J)} \end{aligned}$$

In this case, we see that the balance tips, with the data-center scenario proving more efficient than the p2p one. This is due to the per-bit router baseline energy which is both large and incurred at all routers along the path, making network consumption the dominant factor in the overall consumption in this case of communication-intensive workloads. Moreover, the higher p2p network consumption is exacerbated by: (a) the longer paths longer paths since $d_p > d_s$, and (b) the p2p overhead factor w_r .

6 Extrapolations

The previous section compared the energy efficiency of data-center and p2p systems for select data points. For a more general comparison, we consider the asymptotic behavior of the ratio E_{dc}/E_{p2p} . For simplicity, we consider energy consumption in the network separately from that at servers and peers and look at the above ratio for each case individually. We omit detailed derivations

	Load	Avg pwr (W)	CPU Util (%)	BW (Mbps)	δ (J/bit · 10^{-9})	γ (J/bit · 10^{-9})
Peer	Idle	140	1	0	-	-
Srv	Idle	291	0	0	-	-
Peer	Max	153	30	80	16	0
Srv	Max	336	50	864	5	673

Table 1: Measured power, CPU utilization and network throughput for a peer and a server machine.

and report only our final results.

a) Network energy: When considering only the in-network component of energy consumption, we find:

$$\frac{E_{dc}}{E_{p2p}} = \frac{1}{w_r} \frac{d_s}{d_p} \quad (14)$$

This is usually ≤ 1 , since $w_r \geq 1$ and, for random peer selection $d_s < d_p$. Thus, as expected, p2p usually fares worse, to an extent determined primarily by the efficiency of the p2p protocol.

b) End systems: For the non-network energy component, we find:

$$\frac{E_{dc}}{E_{p2p}} = \frac{c}{w_p} \left\{ 1 + \frac{1}{\mu_s(r-1)} \right\} \quad (15)$$

where, μ_s is the server utilization, and

$$r = \frac{S_{max}}{S_{base}} (= \frac{P_{max}}{P_{base}}) \quad (16)$$

With regard to r : since S_{max} and P_{max} depend on the nature of the workload, we capture this influence by considering two extreme values of r (in today's machines).

- $r = 2$, representing computation-intensive tasks with a maximum power draw as large as double the baseline power (due to high CPU utilization). For $r = 2$, we have

$$\frac{E_{dc}}{E_{p2p}} = \frac{c}{w_p} \left\{ 1 + \frac{1}{\mu_s} \right\} \quad (17)$$

For most values of c , w_p and μ_s , this ratio is likely ≥ 1 .

- $r \approx 1$, which represents computation-light and communication-heavy tasks. In this case, E_{dc}/E_{p2p} tends to infinity and thus p2p always wins. This is not surprising since at this value of r , the extra consumption at peers is essentially negligible.

In summary, from the point of view of end systems, p2p is likely to always win. This is to be expected, since the baseline energy consumption at peers comes for free. The above results lead us to speculate on the potential impact of different strategies for greater energy efficiency:

Improved data-center efficiency: To what extent can more energy-efficient data-center design narrow the gap? At best, data centers might eliminate cooling altogether ($c = 1$), and consolidate workload to run each server at full utilization ($\mu_s = 1$). The latter may be accomplished by using virtual machines (VMs) to host services, and allocating VMs to servers in a way that maximizes utilization. Amazon's EC2 is an example of such an architec-

ture. Even with these utopian improvements, the ratio in eqn. (17) equals $2/w_p$ suggesting that, even for *energy-optimal* data centers, p2p systems will be better if w_p overhead is smaller than $2x$. The culprit here is, of course, the high baseline consumption at servers which leads to our next question.

Energy-proportional machines: What would be the impact of lowering the baseline consumption at servers and having computers consume energy in proportion to their utilization[1]? In this case, our parameter $r = S_{max}/S_{base}$ tends to infinity, and the ratio contrasting end-system energy consumption (eqn. (15)) becomes $E_{dc}/E_{p2p} = c/w_p$. At this point, the comparison between data-centers and p2p systems depends only the relative penalties due to data-center cooling (c) and p2p overheads (w_p).

More efficient peers: We've seen that p2p systems are often less energy-efficient within the network. Can this be remedied? One approach would be to use smarter peer selection. Studies show that a sizable fraction of peers are closer to any given client than is the data center[6]. Intelligently selecting such closeby peers would lead to $d_s/d_p \geq 1$. With this, a p2p system can outperform data-centers even for in-network consumption provided its communication overhead $w_r < (d_s/d_p)$.

Weighing network energy and end system energy: We saw that while p2p systems are more efficient in end-system consumption, data centers fare better on network consumption. Ultimately, the decision on which is preferable depends on the relative magnitude of these two components which in turn depends greatly on the *nature* of the task; computation-intensive tasks spend most of their energy in end systems while network energy is more relevant for communication-intensive tasks.

The above discussion illustrates the importance of overall system architecture in determining the energy efficiency of network services, as also the value of *system-wide* models that capture the energy consumption of networked systems in their entirety.

References

- [1] L. Barroso and U. Hözl. The Case for Energy-Proportional Computing. *COMPUTER*, 2007.
- [2] C. Belady. In the Data Center, Power and Cooling Costs More Than The IT Equipment it Supports. *Electronic-Cooling, Vol. 13, No. 1*, 2007.
- [3] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power Awareness in Network Design and Routing. In *INFOCOM*, 2008.
- [4] X. Fan, W. Weber, and L. Barroso. Power provisioning for a warehouse-sized computer. *Proceedings of the 34th annual international conference on Computer architecture*, pages 13–23, 2007.
- [5] http://www.hpl.hp.com/research/linux/httpperf/.
- [6] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley. Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone. *IEEE INFOCOM*, 2003.
- [7] J. Li, J. Stribling, R. Morris, M. Kaashock, and T. Gil. A performance vs. cost framework for evaluating DHT design tradeoffs under churn. *IEEE INFOCOM*, 1, 2005.