Issues in Linux Mirroring: Or, BitTorrent Considered Harmful

John Hawley
3Leaf Systems
warthog19@eaglescrag.net

Abstract

The Linux community has risen to the challenge of sharing distributions by developing an ad-hoc worldwide collaborative mirroring infrastructure capable of withstanding some of the heaviest network traffic imaginable. It is already capable of moving tens of terabytes a day and it is continuing to grow and expand to meet the needs of a demanding user base. However this ad-hoc infrastructure is not without its faults. Distribution maintainers have complicated, non-intuitive websites to direct users to downloads. Things are made worse by a lack of communication amongst the major releases throughout the year, and a user base that is not always correct in its requests and demands that it puts upon the system.

As the infrastructure grows, the administrators seek out new ways to help manage the stress on everyone involved. BitTorrent has been heralded as one such technology; however, its claims of being better, faster, and more manageable seem to fall short. BitTorrent itself seems to have an upper limit to its capacity that does not match the existing infrastructure. In fact, it has significant downsides to the maintainers, the mirrors, and the users, making it unsuitable as a large-scale primary distribution mechanism.

1 Distributions

Distribution maintainers are at the very core of the Linux mirroring infrastructure. They are creators of the data that the mirrors will be providing to users, and wield a significant amount of control over the experience that both mirrors and users have when downloading. Mirror maintainers and end-users have different issues that distribution maintainers must do their best to respect and work with.

Users, however, are by far the biggest challenge facing both distribution maintainers and their mirrors. Users are an un-relenting mob, capable of bringing some of the largest, and fastest, machines to a crawl. The effect users can have on mirrors during a major release is not dissimilar to a distributed denial of service attack, with the added affect that each user is fighting to use as much bandwidth as can be obtained.

1.1 Keeping it simple

Linux users already face many problems with Linux—getting it shouldn't be one of them. Currently, users trying to download a distribution are asked a multitude of questions, many of which can be unclear and not well understood by the user. This complicates the download process, making it difficult for users to make good choices for their needs. Options should be kept to a minimum by default, as the more options exposed to a user, the more potential for confusion. Where possible, choices should be guessed at for the user, such as choosing a download-mirror based on the geographic location of the user's IP address. A clear and simple mechanism to override the default should be available, so that users can correct or alter the assumptions if necessary.

For example, when users wish to download a distribution, they should be directed by the distribution's website to a download page. The page should default to downloading the latest version, indicate the mirror currently selected, and display "download" icons for each processor architecture. The mirror should be shown in a drop-down list so it can be changed easily. If multiple formats exist (CDs and DVDs, for example), clear icons listing the processor architecture and the format should be present. In the case of a single-file download (like DVD ISOs), upon clicking the icon, the download should just commence without further user action. In the case of multiple file downloads (like CD ISOs), users should then be directed directly to the directory on the mirror server that has the CD ISO images in it so that they may select and download each file on their own.

Archives should be linked to on the page, and a similar strategy to the current version should be used. A full listing of mirrors and their contents should be linked on a separate page, should a user wish to manually browse.

This particular strategy encompasses a number of simplifications to users, and gives distributions much more flexible control over the distribution process. For starters, having a centrally controlled download page gives the distribution a common and simple way to direct users to resources—in this case, their ISO images. It also gives the distribution the ability to attempt to spread the load amongst mirrors, by having a mirror declare its download speed, the country it is located in and what countries it serves a distribution to; the distribution site may intelligently choose, using something like geographic IP lookup, where a user is attempting to download from and provide a mirror that serves that country with sufficient capacity. It also gives the users a very clear and obvious path to get the data, by use of clear icons defining their available choices and providing a simple means of getting the data.

1.2 Mirrors are your friends, treat them with care

There is a growing trend to place more and more requirements on mirrors. They need to mirror more data, requiring more disk space. They may be asked to verify this data, both by internal scripts and by allowing external crawlers to browse their filesystems. And, of course, they need to be able to handle an ever-growing user base. Each new requirement slowly adds a straw to their backs. At some point, even the most powerful mirrors must ask the question—is this too much?

Thankfully, there are ways that this load can be managed so that mirrors don't become overburdened—for instance, limiting the amount of data that needs to be mirrored, having a controlled schedule for crawlers, and spreading out the dates when distributions are released.

1.2.1 Diet Time: Mirrors choice in Legacy data

With new releases coming out regularly, space on a mirror is becoming a greater concern. An average release, ranging from 5GB on the low end to 20GB on the high end, is quite a bit of data that not only has to be stored, but served. Archiving older releases is essential, and distributions already doing this should be applauded. However, distribution maintainers should give mirrors the choice to help by mirroring those archives. This can either be done as an additional target to sync from, or by making the archives available in some other mirror-friendly format. This will not only alleviate loads on slower archive machines, but it also provides legacy users with guaranteed and stable means of downloading packages and ISO images into the future.

1.2.2 Blowing disk cache: Filesystem traversal pain

Distributions have a need to know when a mirror has been updated and to verify that it is up to date and should be included as a valid mirror. The easiest way to do this is to either externally crawl the mirror or to have the site admins add a local process that runs to crawl the repositories and report the results. Both methods have advantages and disadvantages; however, distributions and mirror administrators should be very aware of what these processes do to the servers, as each method causes a linear traversal of the filesystem. This traversal can and does push active data out of the disk cache, causing more data to be sought from disk instead of from the memory cache. This results in severe performance penalties for busy and active mirrors. These kinds of checks should be done sparingly at best as to prevent thrashing of the mirror's disks. A recommendation would be that these kinds of checks be performed at most twice per day, per distribution. This should give distribution maintainers reasonable verification of a mirror's status without causing undue additional stress on the mirrors.

1.2.3 Talk to your neighbor: Scheduling Releases

Lastly, when it comes to distributions, there is one thing that would help mirror administrators immensely—communication amongst the distribution maintainers themselves.

It is becoming common for distributions to follow a set release schedule. While this is a boon to mirror administrators, as they can now easily plan downtimes, upgrades, etc., there is a problem in its current state. A number of these release schedules have become very close together, to the point where in 2007 three major distributions had releases all within the same week of each other. This causes what is best described as chaos

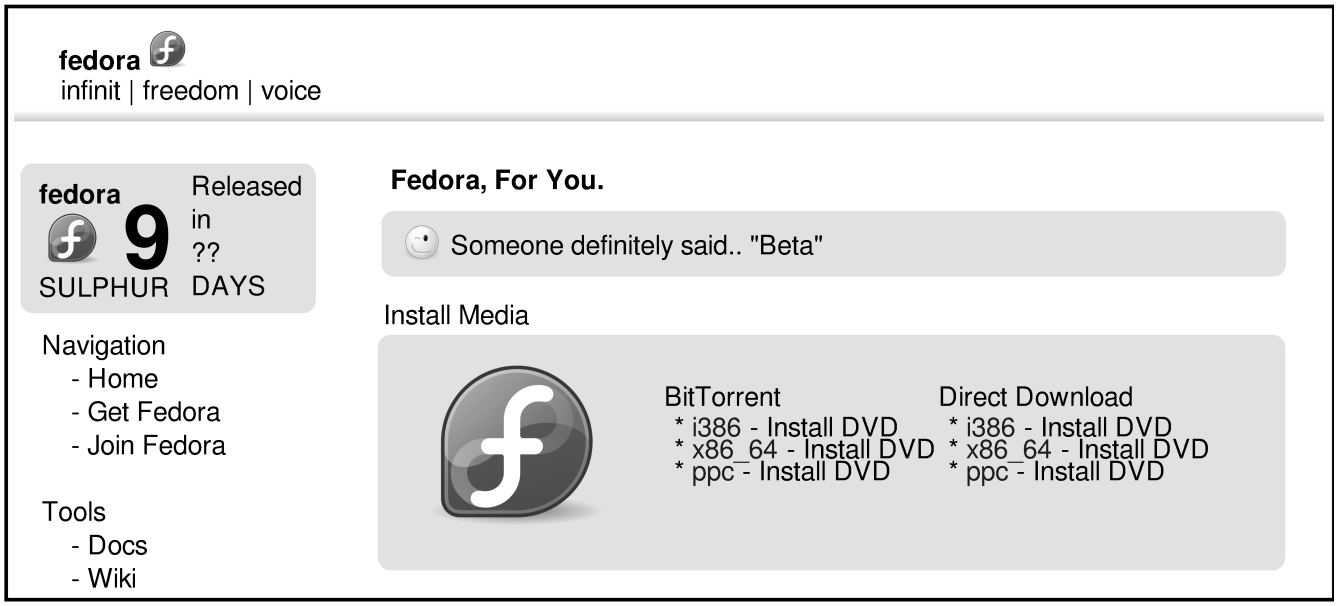

Figure 1: Current Fedora download page

on the mirrors. Where a single distribution could be considered a mad rush, having three distributions release simultaneously is akin to a swarm of locusts.

Handling a release means keeping the working dataset (CD and DVD ISOs, packages, etc.) in memory, having fast enough disk to fetch what's not in memory, and having sufficient network bandwidth. The main problem in this scenario is that while many mirrors are capable of the strain of a single release, tripling the working set's size will greatly exceed the memory available on most systems. Using the Fedora Project as an example, and assuming that only the ISO images are served, this is a baseline of about 18GB of data—a dataset that many mirrors are easily able to hold entirely in RAM. But triple that to 55GB or so of data, and even the largest mirrors must now constantly read everything from disk in order to serve data. This is compounded by the corresponding increase in download requests; more people downloading data means less bandwidth for everyone, thus downloads take longer and load is substantially increased on the mirror servers. For this reason, better communication amongst the distribution maintainers is essential to mitigate these overlaps in releases, and provide the best possible experience for everyone.

2 A better understanding for the user

User:
noun

1. a person who makes use of a thing; someone who uses or employs something
2. a person who uses something or someone selfishly or unethically [syn: exploiter]

... [1]

Users are the reason the mirrors exist in the first place: they are the client and the customer, and as a whole are a very demanding and diverse group. Each individual brings a very different set of expectations, needs, and goals when he or she goes to download the data that is being served. However, there are some things users should be aware of, and keep in mind, to gain a better understanding of what is going on under the surface. This knowledge will help them make better choices in their downloading, from mirror selection, to package selection. This will have set the expectations they bring to the entire downloading process.

[1] http://dictionary.reference.com/browse/user

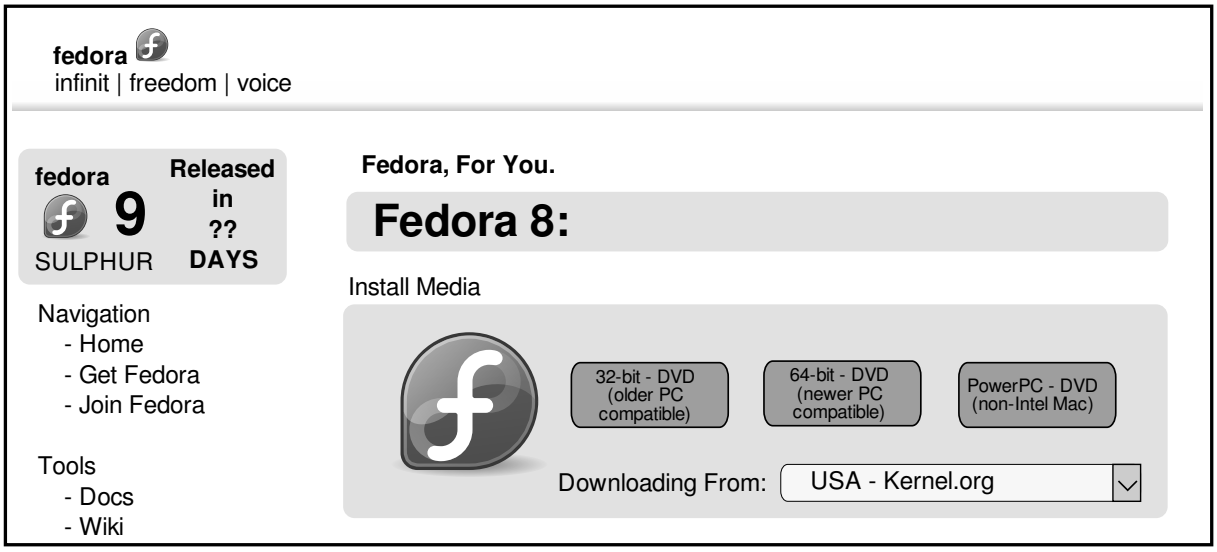

Figure 2: Proposed Fedora download page

2.1 The user isn't always right

Seeing the users as the clients or the customers is quite natural; they are seeking to download data from the distributions. However, going against accepted wisdom when it comes to customers, users are not always right. It is not that their opinion is explicitly invalid, it is that the vast majority of users are ignorant of the complexities in the entire mirroring process, and as a result lack sufficient insight and details to make the correct choices or significant suggestions. This can be alleviated primarily by the users recognizing their own limitations, and that their primary goal is to download data.

2.1.1 Users see things from their perspective

Consider a group of people sitting around a table with an irregular object in the middle of it. Ask each person in this group to describe what they are looking at, and you will get a slightly different response from each of them as they attempt to describe what, to their perspective, is a completely different object. Users are in a similar situation, as each user is staring at the entire mirror process from a different perspective; some go directly to a mirror, some to a distribution, and some follow links on third-party sites to get at the data they are seeking. They are also seeing just one side, the user-facing side, of a much greater system that is working behind the scenes. Users should be aware that what they are seeing is the culmination of a huge amount of work, theory, and practice, so that they may click a link of some sort and download their data.

Users may not be aware of the ramifications in suggesting or demanding changes in the entire mirroring structure. These consequences may be non-obvious and have significant impact, of which a case in point: many users consider BitTorrent to be something that can and does help alleviate the loads of mirrors; however, this is not strictly the case, and in fact may be detrimental the the mirrors and distributions. Push from the user community at large can have consequences that distribution maintainers and mirrors must carefully balance. To ensure the best user experience for everyone, it is sometimes necessary to discard even popular suggestions.

2.1.2 Diversity of Thought

There are by far more users than there are distribution and mirror administrators. Therefore the push from the user community for enhancements and changes can be quite strong and varied. This does not mean, unfortunately, that the outcry for change in the mirroring infrastructure is necessarily valid or useful. Distribution and mirror administrators should be cautious and careful in implementing the demands of their user base, as many changes are ultimately detrimental. This is not to say that user suggestions are all invalid, but rather to note

that they may have consequences that adversely affect the entire mirroring process. It is also possible that a suggestion that is brought forth is attempting to solve a problem that no one is actually facing. Users should be careful of getting swept up in hype or marketing about specific solutions or technologies; they may be great for certain applications, but they are not always guaranteed to give a better user experience, or may provide benefit to a small minority, but be detrimental to the majority.

It should be kept in mind that every user may be capable of putting forth suggestions, but with there being more users than administrators to experiment, analyze, and verify things, there will always be technologies, ideas, or problems that cannot be addressed or investigated. Distribution and mirror administrators' ranks are filled with incredibly smart and dedicated people who are working on a multitude of problems that many of these individuals relish working on; however, they are a small bunch of people and their time is not unlimited. If there is a problem, it should be voiced, but it should not be taken for granted that it will be solved immediately just because the issue was raised, or that any solution chosen to solve the problem will necessarily match anything suggested.

2.1.3 Beware Arm-chair Administrators

Many users do have valid points, concerns, and issues. There are, however, some that do not have a willingness to accept that they may be wrong. There are countless individuals, the world over, who are working within the systems they have at their disposal in universities, corporations, non-profits, and their own personal equipment to provide mirrors. Each mirror administrator knows the limitations of what they can and cannot do with what they have, and the distributions have a nigh-impossible job of herding these volunteers into a coordinated force capable of providing, with amazing efficiency, the huge amount of data that is downloaded 24 hours a day, 7 days a week, 365 days a year. These people are not perfect, but users who proclaim that they themselves "can do it better," that the administrators are incompetent, or who lambaste the complicated and often hard decisions either that these administrators are forced to make, are usually wrong. If these individuals that claim they can "do it better" are convinced they are, they should make constructive criticism and suggestions to the distribution and mirror maintainers. Everyone is open to criticism, constructive suggestions, and help. Users who assume their solution, when put forth, is the absolute correct solution should look back on the last two sections. Each user when putting forth a suggestion or comment should consider that he isn't seeing the entire picture, and it may be that the opinion causes more harm than good.

3 BitTorrent

Since its inception in 2001, BitTorrent has been proclaimed as the means of eliminating the mirror infrastructure, and that it will providing a faster, better means of content distribution. It is based not on downloading from a central repository, or repositories (the typical mirror infrastructure), but on a central point that coordinates the masses of users who wish to download the data and harnesses their collective bandwidth to alleviate the load from the mirror servers and increase total available bandwidth. This is accomplished through every user participating in both downloading and uploading content to the cloud. Distributions and mirrors have recently been exploring or adopting BitTorrent as an alternative means to download their content. This in part due to a perceived user demand, and to explore the possibilities of this technology as a means to more effectively use the resources at the disposal of the distributions and mirrors. However, the motivation should be questioned, beginning with, what problem is BitTorrent really solving? If relatively few nodes perform the vast majority of the uploading, how is this any different or better than providing the same files via more traditional mechanisms like HTTP and ftp? Is BitTorrent straightforward enough for the average user to understand the complex implications of using it, as opposed to traditional download mechanisms? With the rising resentment against BitTorrent from Internet service providers, is this going to adversely affect BitTorrent as a download mechanism? These are but a few of the questions that must be asked about BitTorrent as the answers to these questions affect every layer in the mirroring infrastructure: distributions, mirrors, and users alike.

3.1 What it's good for / Where it's useful

BitTorrent's original intent was to provide a simple mechanism to alleviate the problem of downloading large amounts of data when there is no established mirroring infrastructure in place, or the mirroring infrastructure is incapable of handling the demand put upon

it. In 2001, this was a serious concern, as it was quite possible for large and popular datasets to cause the meltdown of both servers and network infrastructure. In some cases, this caused noticeable slowdowns and bottlenecks on the entire Internet. BitTorrent's intention was to come to the rescue by distributing the combined load to every user who was participating. By taking advantage of the aggregate resources available, users were than able to download faster, and in downloading, helped make downloads for others faster by also uploading the content that a user has.

As BitTorrent has matured and become more accepted, it has been found to be exceedingly useful for moving large datasets of any type, be it multimedia, software, or anything, really. BitTorrent performs best in scenarios where their is more than a single server and client in the cloud. This has become particularly popular where there is not, or cannot be, large and established mirroring infrastructures. This is seen in small open-source projects with large datasets and small followings, but more commonly in illegal downloading. While BitTorrent has been popular for these smaller, more targeted, distribution channels, there are a few commercial exceptions[2] that are providing torrents.

3.2 Where BitTorrent falls flat on its face

While BitTorrent has the ability to create a respectable distribution mechanism where none exists, by its very nature it has an Achilles heel when large number of users are in the cloud. The tracker, or the controlling unit of the cloud, must pass messages to each of the clients being used. This puts a load on the tracker, and sets a finite limit to how fast it can respond to and process the data in the cloud. As the cloud increases in size, it does not keep the same level of efficiency or productivity when pitted against a mirror structure or a very large user base, such as the one used to distribute Linux. There are facets of BitTorrent that make it particularly painful to an established mirroring structure, especially if the mirrors themselves participate in the BitTorrent cloud.

[2]Warner Brothers, Paramount, and BitTorrent Inc.'s own entertainment network.

3.2.1 What the numbers show

With BitTorrent's rise in popularity, `kernel.org` has been running experiments exploring its use as more distributions attempt to push it as a download mechanism. During these tests, data has been recorded and analyzed for many distributions. This paper discusses the Fedora 7 and 8 releases, as they are most consistent and established of these numbers. `Kernel.org` on both of these occasions joined and participated in the BitTorrent cloud from machines that were dedicated to this purpose. These machines were not providing the same data over traditional download mechanisms like HTTP and ftp. The numbers reflecting the Fedora 7 release used a stock configuration of rTorrent, which would be the normal and expected setup for a typical user. The only exception to this was that the two machines running in this experiment each had three instances of rTorrent running simultaneously. The numbers reflecting the Fedora 8 release, however, add two additional machines and the original two machines maintained three instances of rTorrent, while the two added machines each ran five instances. For the Fedora 8, release rTorrent's configuration was also modified to allow for the maximum possible peers, simultaneous uploads, upload and download rates. Figures 3 and 4 show the amount of data moved by the cloud as a whole versus the data moved by `Kernel.org` acting as a part of the cloud.

It should be noted, in the case of Fedora 8, that `Kernel.org`'s Pub 1 and Pub 2 servers were explicitly throttled. This was done to prevent bandwidth issues to the machines serving HTTP, ftp, and rsync, which reside on the same network.

The numbers for BitTorrent reveal quite a bit, not the least of which is that a small change in configuration can cause a dramatic change in the behavior of the BitTorrent client. The data also brings into question BitTorrent's ability to keep up with the mirroring needs of a major distribution such as the Fedora Project. In the Fedora 8 release, it can be shown that it is quite possible (in fact, quite probable), that a very few number of nodes are performing the vast majority of the work in the BitTorrent cloud. This is likely due to people leaving the cloud once they have completed their downloads: there is no continuing advantage for the user to continue uploading into the BitTorrent cloud after acquiring the full download. This leaves the cloud increasingly dependent on the few seeders who have a full copy of the

Fedora 7

	size	downloaded	data transferred by BitTorrent cloud	percent of total	transferred by Kernel.org[a]
Fedora-7-KDE-Live-i686	686MiB	4,900	3,200,000	18.85%	603,269.4
Fedora-7-KDE-Live-x86_64	831MiB	1,615	1,280,000	46.06%	589,629.2
Fedora-7-Live-i686	699MiB	8,044	5,360,000	12.95%	693,861.8
Fedora-7-Live-x86_64	779MiB	3,084	2,290,000	15.03%	344,127.6
Fedora-7-i386	2.79GiB	33,909	92,590,000	3.35%	3,097,718.9
Fedora-7-ppc	3.49GiB	957	3,260,000	28.03%	913,751.0
Fedora-7-x86_64	3.3GiB	10,448	33,730,000	6.45%	2,175,682.0
		Totals:	141,710,000	5.94%	8,418,039.9

	Pub 1[b]	Pub 2[c]	Total
Fedora-7-KDE-Live-i686	301,655.9	301,613.5	603,269.4
Fedora-7-KDE-Live-x86_64	145,080.6	444,548.6	589,629.2
Fedora-7-Live-i686	346,413.4	347,448.4	693,861.8
Fedora-7-Live-x86_64	175,252.0	168,875.6	344,127.6
Fedora-7-i386	1,503,176.6	1,594,542.3	3,097,718.9
Fedora-7-ppc	460,143.1	453,607.9	913,751.0
Fedora-7-x86_64	1,111,975.6	1,063,706.4	2,175,682.0
Totals:	4,043,697.2	4,374,342.7	8,418,039.9

[a] this is the total amount of data transferred through BitTorrent by Kernel.org's Pub1 and Pub2 servers
[b] Machine was un-throttled, and has 1gbps of upstream bandwidth
[c] Machine was un-throttled, and has 1gbps of upstream bandwidth

Figure 3: Fedora 7 BitTorrent downloads of the cloud as a whole and of `kernel.org`

data and who are dedicated enough to stay in the cloud despite having a complete download, or mirrors such as `kernel.org` acting explicitly as a seeder.

BitTorrent's performance also falters when you directly compare it to more traditional download mechanisms such as HTTP, FTP, or rsync. For our testing purposes BitTorrent was given 15% more usable bandwidth, and four machines while the traditional download mechanisms used only two machines. Despite these advantages, BitTorrent did not outshine the traditional download methods. For the Live CD images BitTorrent only moved more data in four of the five torrents. In the more popular DVD install images BitTorrent was unable to keep up lagging by 72% and 204.12% for the x86_64 and i386 downloads respectively, and beating out the PPC downloads by a small margin. Looking beyond pure number of bytes moved, BitTorrent moved 33,111 images as a whole. This pales in comparison to the mirroring infrastructure which has a hundred or so mirrors in it, and with a single mirror, `Kernel.org`, moved 21,901 images. These numbers, however, reiterate BitTorrent's primary purpose: a distribution mechanism for downloads that do not have more structured mirroring and distribution mechanisms.

3.2.2 Immensely manual process for admins

The classic distribution mechanisms (HTTP, ftp, and rsync) are very simple for both distribution and mirror administrators. Simply put the files in a downloadable location, and the mirrors download the data to their servers. When the time is correct, the distribution and the mirrors perform what is commonly known as a "bit flip" (or a changing of the file permissions) to allow normal users to acquire the data. This is quite simple for both parties; in fact, if a mirror admin wished, after initial setup was done in such a way as to download from the distribution on a regular basis, the distribution is the only entity that needs to manually change the permissions on the data and those changes will propagate to

Fedora 8

	size	downloaded	data transferred by BitTorrent cloud	percent of total	transferred by Kernel.org[a]
Fedora-8-Live-KDE-i686	698MiB	6,710	4,460,000	28.24%	1,259,591.4
Fedora-8-Live-KDE-x86_64	805MiB	1,663	1,270,000	29.55%	375,280.9
Fedora-8-Live-i686	697MiB	10,642	7,070,000	22.08%	1,561,068.2
Fedora-8-Live-ppc	698MiB	641	437,550	36.18%	158,286.5
Fedora-8-Live-x86_64	766MiB	2,649	1,930,000	25.2%	486,375.4
Fedora-8-dvd-i386	3.28GiB	33,111	106,380,000	22.81%	24,261,040.5
Fedora-8-dvd-ppc	3.96GiB	1,071	4,140,000	36.48%	1,510,322.9
Fedora-8-dvd-x86_64	3.71GiB	12,017	43,550,000	28.86%	12,569,610.7
		Totals:	169,237,550	24.92%	42,181,576.5

	Pub 1[b]	Pub 2[c]	Pub 3[d]	Pub 4[e]	Total
Fedora-8-Live-KDE-i686	232,696.2	257,395.4	221,166.9	548,333.9	1,259,591.4
Fedora-8-Live-KDE-x86_64	79,563.7	78,471.2	68,880.2	148,365.8	375,280.9
Fedora-8-Live-i686	286,141.2	322,965.5	242,441.7	709,519.8	1,561,068.2
Fedora-8-Live-ppc	35,926.2	36,520.0	29,412.0	56,428.3	158,286.5
Fedora-8-Live-x86_64	97,050.5	109,541.5	82,232.9	197,550.5	486,375.4
Fedora-8-dvd-i386	4,956,911.9	5,492,479.7	3,586,870.5	10,224,778.4	24,261,040.5
Fedora-8-dvd-ppc	381,919.8	300,517.5	299,703.8	528,181.8	1,510,322.9
Fedora-8-dvd-x86_64	2,479,605.9	2,760,286.6	1,751,454.1	5,578,264.1	12,569,610.7
Totals:	8,548,815.4	9,087,677.4	6,282,162.1	17,991,422.6	42,181,576.5

[a] this is the total amount of data transferred through BitTorrent by Kernel.org's Pub1 and Pub2 servers
[b] Bandwidth throttled to a max of 240.8mbps for the machine
[c] Bandwidth throttled to a max of 240.8mbps for the machine
[d] Machine has a maximum of 100mbps of bandwidth due to upstream provider
[e] Machine was un-throttled, and has 1gbps of upstream bandwidth

Figure 4: Fedora 8 BitTorrent downloads of the cloud as a whole and of `kernel.org`

the mirrors automatically. This is very straightforward, simple, easy to verify, and robust for all parties involved in the mirroring process. BitTorrent is not, at least in its current implementation, quite as simple for the distribution or mirror administrators to set up.

The process for BitTorrent is more cumbersome from the distribution administrator's point of view. The administrator must put together package sets and create the torrents, which takes some additional effort. Typically these torrents are unavailable until the point at which "bit flip," or release, occurs; so there is no way for the mirrors themselves to join the cloud early. There is an added difficulty that per distribution, per release, the torrent files are either inconsistent in where they are, or not present at all. It is also made more difficult if the files defining the torrent are not present, or they are not in a location where the torrent file is expecting the ISO images to be. This makes it immensely time-consuming for the mirror admins to participate, should they choose, in the BitTorrent cloud, as they must hand-craft the entire structure, or face re-downloading the data once the torrents are available. There is a means of setting up a more automatic searching of the file system to find and automatically join torrents; however, this would cause additional load on the system, as it will have to walk the entire file space regularly in search of those torrents, and again the structure in many cases is not set up to have the files pre-configured in the correct structure.

3.2.3 Loading of a machine

BitTorrent is designed to manage the cloud and all of the portions of the images that are available. While this works well when a small number of machines are asking a host for data, it does not scale to thousands. This causes the hosting machine to get, effectively, random requests for sections of the data, meaning that it can not sequentially read the file out, and take advantage of

Fedora 7

	Mirrors1			Mirrors2			Totals	BitTorrent Ratio (From Above)
	HTTP	Ftp	Rsync	HTTP	Ftp	Rsync		
Fedora-7-Live-x86_64.iso	144	24		146	31		345	441.75
Fedora-7-Live-i686.iso	1,062	184		1,031	142		2,419	992.64
Fedora-7-Live-KDE-x86_64.iso	120	21		104	24		269	709.54
Fedora-7-Live-KDE-i686.iso	664	129		633	94		1,520	879.40
Fedora-7-ppc-DVD.iso	145	20		107	17		289	441.75
Fedora-7-x86_64-DVD.iso	2,337	281	17	2,145	226	47	5,053	320.46
Fedora-7-i386-DVD.iso	10,579	1,193	55	9,146	865	63	21,901	365.43

Fedora 8

	Mirrors1			Mirrors2			Totals	BitTorrent Ratio (From Above)
	HTTP	Ftp	Rsync	HTTP	Ftp	Rsync		
Fedora-8-Live-ppc.iso	42	5	18	37	2	14	118	226.40
Fedora-8-Live-x86_64.iso	178	31	34	162	16	30	451	634.36
Fedora-8-Live-i686.iso	1,339	111	50	1,167	80	38	2,785	2,238.90
Fedora-8-Live-KDE-x86_64.iso	67	29	33	67	17	30	243	456.19
Fedora-8-Live-KDE-i686.iso	588	82	50	547	71	37	1,375	1,803.80
Fedora-8-ppc-DVD.iso	171	27	16	141	14	13	382	398.71
Fedora-8-x86_64-DVD.iso	2,703	364	38	2,322	238	47	5,712	3,307.36
Fedora-8-i386-DVD.iso	10,716	951	55	9,416	712	51	21,901	7,201.36

read-ahead when sending data to clients. This random access across the disk, which will only get more frequent with the number of clients in the cloud, will very quickly begin to adversely impact the system, causing a rise in load and added stress to the disk. On a system that may already be serving traditional download methods, this constant seeking on the disk can cause loads to rise dramatically, impacting performance for both BitTorrent and the traditional download methods. This makes its use on a normal mirror machine questionable due to the adverse impact it would have on the system.

3.3 Increasing problem to users

There are several technical and logistical reasons that BitTorrent is unsuitable for mirror usage, but there is also a number of hurdles and complexities to a user, both external and internal to their control, that can adversely affect a user's experience in using BitTorrent. BitTorrent, to make it usable, needs to have a routable port for other clients to connect to and request data from. This, however, poses an issue for users, as many of them are behind a NATed firewall that they may or may not control. On top of that, many users are unaware of this particular issue and don't know that a port needs to be forwarded to their computer. This causes confusion and a lack of understanding about why it "doesn't just work." Things like rsync and HTTP do this without firewall changes. Users may also not be able to control the network they are on, for example a corporate network, where a user is unable to alter the firewall to make use of BitTorrent, thus making the experience painful and unusable.

BitTorrent has also come under fire from Internet Service Providers (ISPs) who feel that BitTorrent is primarily being used for illegitimate purposes. This has lead many large, and small, ISPs[3] to begin performing various things to either slow down BitTorrent traffic or to outright block it. This can be problematic for users, and may be undetectable by them, causing frustration at the inability to find what the cause is.

[3]Comcast is probably the most famous currently; however, Azureus, a Java-based BitTorrent client, keeps a list of known problem ISPs at `http://www.azureuswiki.com/index.php/Bad_ISPs`

3.4 BitTorrent—too late to the party

While evaluating the feature list and promises of BitTorrent, it seems like it could be the silver bullet that solves many problems for distributions and mirror administrators. However, during the course of real testing and looking at BitTorrent from the perspective of the distributions, the mirror administrators, and the users, there are a number of rather serious concerns and issues that come up that should give all three groups concern. Requests are coming from users to provide BitTorrent, and maintainers are seeking ways of making their distribution process faster and better. Users are seeking a magic bullet, and they have been lead to believe that BitTorrent is it. They want a faster and easier means of downloading the data they want. The reality of the matter, however, is that their calls and howls for BitTorrent to be provided are not made with a full understanding of the impact it has on the system. Much of the infrastructure to provide the user a better experience is in place today. There are hundreds of mirrors around the world ready to mirror the data and distributions have created the infrastructure to manage and pre-distribute the data to the mirrors before release. All that is left to be done is for the distributions to provide a simple and straightforward user interface to interact with so that users can simply download the data they are seeking.

Distributions should endeavor to make and *keep* things simple and straightforward for the end user. Users should be given few (but clear) options, and choices should be limited to the bare few needed. Distributions should endeavor to provide a only the most popular formats as a default, leaving less-common formats to be generated by the end users themselves, with a mechanism provided by the Distribution. The intent is not to take all options away from the users, but rather to make things as straightforward and simple for the majority, giving the minority tools to meet their more specific needs. Along with this "simpler is better" approach, basic coordination amongst the distributions is critical, mainly to prevent overlapping release schedules, but to provide better discussion and feedback on what mirroring practices are working and which aren't. There are issues in the mirroring infrastructure currently, but these are solvable problems. With a better understanding of the issues and problems faced by everyone, solutions and practices can be put in place to better served.

Proceedings of the Linux Symposium

Volume One

July 23rd–26th, 2008
Ottawa, Ontario
Canada

Conference Organizers

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*
C. Craig Ross, *Linux Symposium*

Review Committee

Andrew J. Hutton, *Steamballoon, Inc., Linux Symposium, Thin Lines Mountaineering*
Dirk Hohndel, *Intel*
Gerrit Huizenga, *IBM*
Dave Jones, *Red Hat, Inc.*
Matthew Wilson, *rPath*
C. Craig Ross, *Linux Symposium*

Proceedings Formatting Team

John W. Lockhart, *Red Hat, Inc.*
Gurhan Ozen, *Red Hat, Inc.*
Eugene Teo, *Red Hat, Inc.*
Kyle McMartin, *Red Hat, Inc.*
Jake Edge, *LWN.net*
Robyn Bergeron
Dave Boutcher, *IBM*
Mats Wichmann, *Intel*