

Illumos Productivity and Bus Factor

January 28, 2015

List of Figures

1	Scatterplot of commits against files	3
2	Scatterplot of commits against files, blue dots are new guard	4
3	Histogram of commit distribution by author, New Guard on bottom facet	6
4	Histogram of commit distribution by author, unsegregated	7

1 Some History

In the beginning was Unix. Unix begat BSD and System V. BSD begat SunOS. SunOS and System V begat Solaris. Probably among the worst Unix-like systems in existence at the time. Over the decades, Solaris saw immense improvements in reliability and performance. In the mid 2000s, the Solaris code base was the birthing suite for state of the art systems tech such as ZFS, Zones, and DTrace to name the three most venerable.

By the mid 2000s, Solaris was seeing fierce competition from the enterprise Linux systems — though, the competition was not in the form of technological superiority. Linux, having always been inferior in enterprise scenarios, competed on price. It cost nothing to use, and the fact that it was open source allayed users’ fears of lock-in by a big company (like, Sun). It was the software industry equivalent of **asymmetric warfare** — and it worked out splendidly. For the price, Linux was good enough.

In 2004, Sun open sourced Solaris under the name OpenSolaris. Had this happened a **decade earlier**, Linux would probably not have taken root in the enterprise. Nonetheless, Sun finally accepted that it was not enough to have better technology — Solaris had to beat Linux at its own game. The OpenSolaris project, for the first time in Solaris history, allowed a group of global volunteers to contribute to the code (before this was restricted to specific American universities). It also removed one Linux’s key advantages — giving the community total control of the code.

Unfortunately, the economic crisis hit in 2008, and by the end of the year Oracle acquired Sun. By 2009, the OpenSolaris community had grown to include a number of contributors and adopters that absolutely depended on the unique and superior technology within the system. On August, 13 2010, Oracle had **pulled the plug** on the OpenSolaris project. This was the beginning of the crisis for the OpenSolaris community, which would rechristen itself the Illumos community. It was the equivalent of **Order 66** in the Star Wars universe.

After 08-13-10, the prognosis for the code base was dim. However, the engineers that spent a large chunk of their careers working on the code, would not let it die behind the walls of Oracle. Many of the best engineers (in terms of both ability and moral integrity) left Oracle to work at or otherwise create companies that depended on cutting edge technology and the right to change and improve that technology on their own terms. In other words the Illumos project was born in a time of crisis. It was on life support, and it had to go through physical therapy.

2 The Present

That was half a decade ago. The Illumos project survived so far, however it is still a system with marginal industry and hobbyist adoption. It also isn't clear to the outside world that the project is attracting new volunteers. There are a lot of contributors that were around since the Sun days. And that is great. However, many of them are no longer as young as they used to be. Many of them have families and friends that increasingly demanding their attention more than the Illumos project. This is a basic human truth, and will probably happen to every young engineer some day. In other words, the Illumos community — in order to survive for decades to come — needs to attract young engineers, be they volunteers or employees. The community needs a New Guard that will gradually replace the Old Guard.

3 Metrics

The goal of the Illumetrics project is to measure various things — such as commit activity and mailing list activity — that can act as a proxy for the vital signs of the Illumos community. We want to answer whether or not the community has recieved New Guard contributors, how many they have recieved, how productive they are, and what the bus-factor of the New Guard is. But when can someone be considered a member of the New Guard? It seems logical that a member of the New Guard must be someone who started contributing code to the Illumos code base after the date of Order 66 (08-13-10). This is a pretty solid and unambiguous point of separation.

Another obvious question is what would be an appropriate metric for the amount of contributions one made to the project? Number of commits? Number of files modified? Number of lines modified? Or in other words: what unit of work can be considered a contribution? Even though, the Illumetrics project measures all of these things, it appears that a commit maps better to the notion of a contribution more than a file modification or a line modification. For example one contributor has modified in excess of 2000 files, while having made only 4 commits — way above the average file-changed-to-commits-made ratio. This is because this contributor's first commit was the removal of the Perl runtime from the Illumos repository. That runtime contained more than 2000 files. The number of files changed, in this case, isn't proportional to the work done. All it took to make this change was one `rm -rf perl` command.

Below is a plot of commits made by authors against the total number of files modified by the same author.

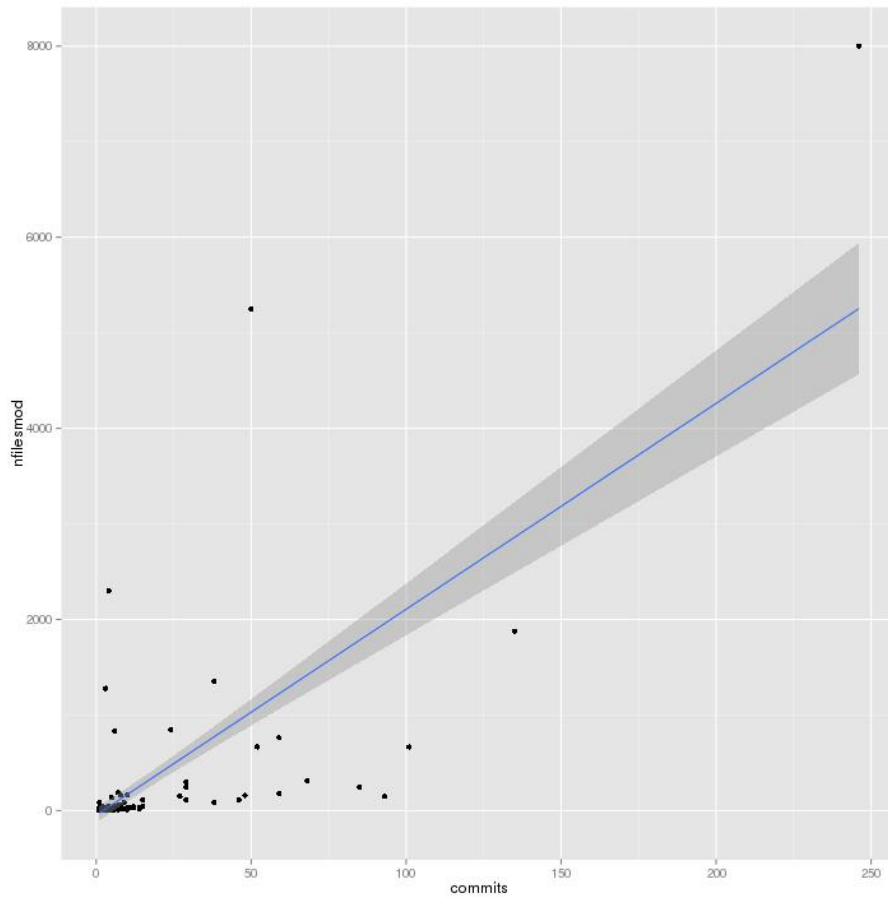


Figure 1: Scatterplot of commits against files

As you can see there is a vaguely linear correlation, but it isn't very tight, and there are a few outliers on the left side of the graph, where those with few commits have modified many files.

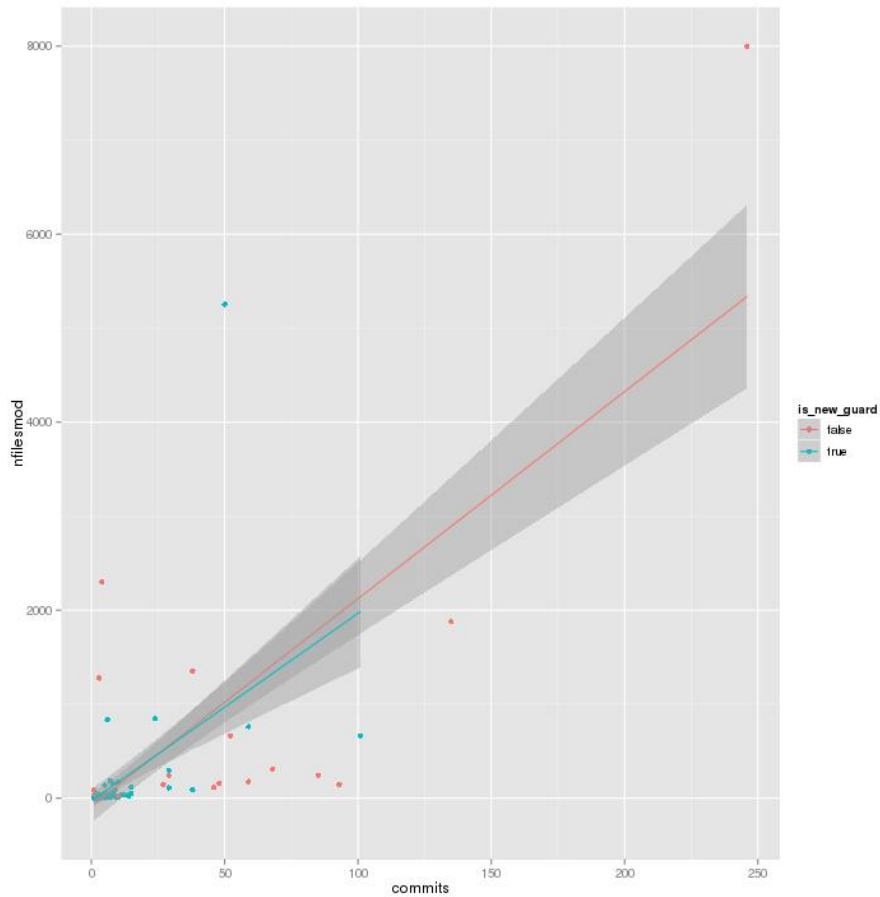


Figure 2: Scatterplot of commits against files, blue dots are new guard

The same plot, but the points are colored according to whether an author is part of the new guard or the old guard.

A good analogy to this situation would be whether one should measure scientific contributions made to a journal by number of papers written, or by number of sentences written in that journal. Number of commits made is not an ideal metric — not all commits carry the same value — but for now, it's the best one we've got. Although, I am contemplating better ways of quantifying contribution quantity and quality. I suspect issue-tracker data would be part of the answer to this. The issue tracker for `illumos-gate`, is very obviously **located**. Whereas the issue tracker for `illumos-joyent` is a bit more difficult to **find**.

4 Bus Factors

The bus factor is defined as the *number of people that have to get hit by a bus, before a software project can no longer continue*. Most people assume that this factor is equivalent to the number of contributors a project has. If only that were so. Illumos (well, the `illumos-gate` repo) has had a total of 186 contributors since Order 66. Of these 63 are members of the old guard, while 126 are members of the new guard. This is very good, since exactly $2/3$ of contributors were recruited *after* Oracle tried to kill the project. This means that there *is* interest in the project, and that it is growing. However, this does *not* give the project a bus factor of 186.

Not all contributors contribute equally. Some are much more prolific than others. In fact 59% of contributors have made 2 commits or less. In fact, if one makes more than 3 commits, one is in top $1/3$ of contributors. If one makes more than 5 commits to the project, one is in the top $1/4$ of contributors.

The commits per author follow an exponential distribution. As evidenced in the plot below.

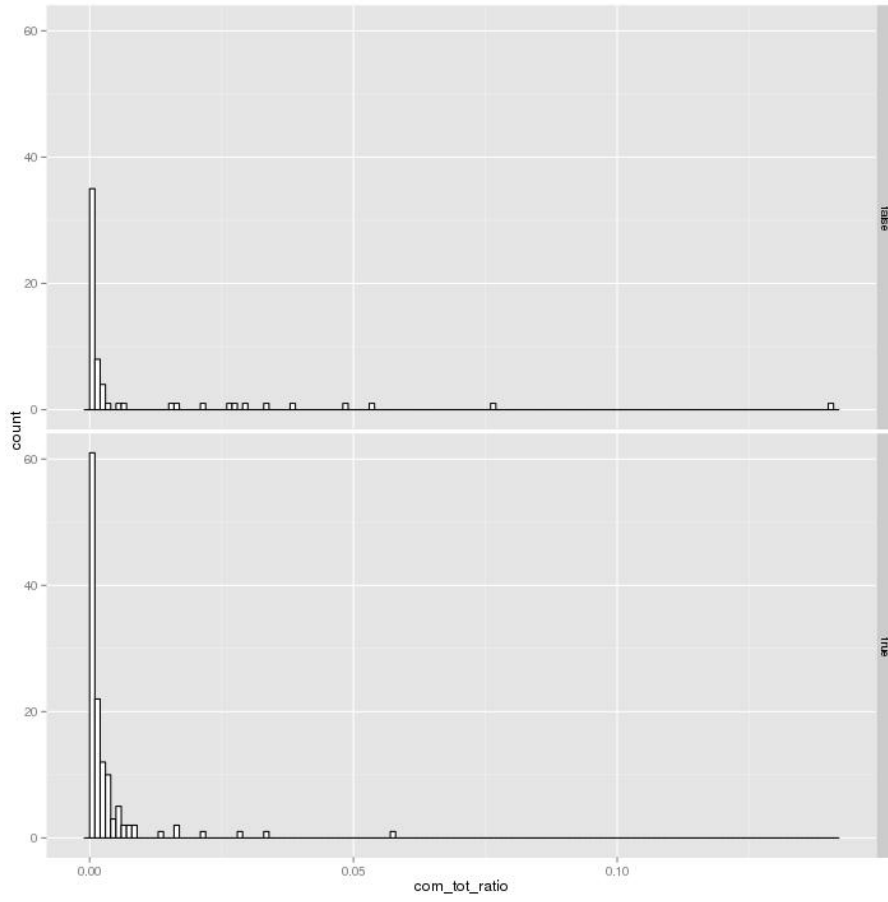


Figure 3: Histogram of commit distribution by author, New Guard on bottom facet

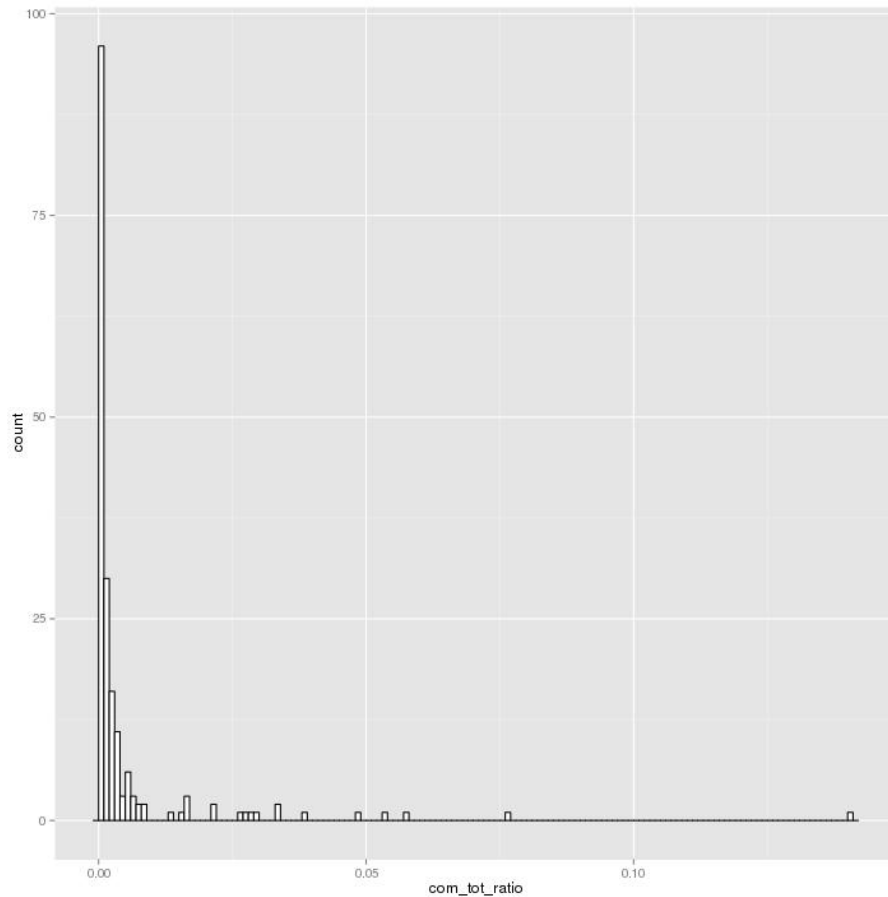


Figure 4: Histogram of commit distribution by author, unsegregated

While not precisely visible in the plots, let me inform you that the top 5 contributors — regardless of guard membership — are responsible for 37.6% of the commits to the project. The top 10 are responsible for 54% of the commits. This means that if the top 10 contributors get hit by a bus, 54% of the community’s productivity will disappear. If the top 20 get hit by a bus, 72% of that productivity disappears. If 30 get hit, 80% of the productivity evaporates. If 60 of the top contributors get hit, that’s 90% of the productivity. After 60, the percentage doesn’t change almost at all. In other words, 32% of all contributors are responsible for 90% of the work. More eerily, 20% of the contributors (37) are responsible for 82% of the commits. This coincides with **Pareto’s Economic Principle** that 20% of a country’s population holds 80% of the wealth in that country. This may suggest that **life is indeed a fractal**.

The Illumos project’s bus-factor is *at best* 60. But the loss of just the top 10 would be a major set-back for the project. Let’s hope that the top contributors aren’t in the

habit of flying on the **same air plane together**.

Of the top 60 contributors, 19 are old guard, while 41 are new guard. This, as anticipated, is a near-reflection of the total ratio of old guard to new guard: 1/3. But as we look at fewer and fewer top contributors, that ratio gets larger. For the top 40, 2/5 are old guard. The top 30, have 1/2. The top 20, have 3/5. The top 10 have 7/10 old guard. Of the top 5, only 1 is new guard.

This makes sense, as the old guard has more experience with the code base. The new guard needs time to get into the swing of things. This also means that the old guard members of the top 10, are responsible for 42% of all commits to Illumos. More globally, 3/5 of all commits since Order 66 come from the old guard even though the new guard out-number them 2 to 1. This will likely change over time, as the old guard gets older and retires, and as the new guard gradually morphs into the old guard.

5 Exemplars

Since I'm sure the question is on your mind, the only new guard member to be in the top 5 of `illumos-gate` is **Robert Mustacchi**. This guy is single handedly responsible for 5.7% of total commits, and for 14% of the commits *within the new guard*. Considering that 0.1% is the percentage of commits for the bottom 50% of contributors, this make Mustacchi 57 times more productive than each individual in the bottom half of the list. **10X programmers** may not be a myth after all. This productivity makes him the third of the top 3. Not bad for a relative new-comer. In the `illumos-joyent` repository he has contributed much more, but it will likely take time for some of those changes to trickle to the gate, through the review process. Mr. Mustacchi is certainly the exemplar for the new guard.

The most prolific contributor to `illumos-gate` is Richard Lowe, who is responsible for 14% of total commits and 24% of commits from *within the old guard*. He is a force of nature. The Illumos community owes much to his industriousness — and if I could, I would personally build a bronze statue in his image, right here in New York.

6 Conclusions

The Illumos project has avoided death despite Oracle's best efforts. It has managed to recruit a non-trivial amount of new contributors, especially when considered in terms of the proportion of old guard to new guard contributors. So the immediate future looks bright, especially when contrasted against the state of affairs immediately following the Friday 13th letter. So that's good news, as far as the developer community is concerned. The complete lack of data for the user community — so far — makes it difficult to draw any solid conclusions. Not all developers contribute equally, and so the bus-factor is fairly small. The loss of just a few developers can be damaging to the project. I'm not sure what concrete steps can be taken to increase the bus-factor.

7 Future Metrics

The commits are a good starting point, however there is much more to be measured. We have plans to scrape mailing list data from an online mailing list archive. This way the metrics won't just measure development activity but also non-development community activity. As mentioned, data in issue trackers can help us categorize commits. Also, we've only touched on the activity surrounding `illumos-gate`. We want to cover other `illumos-` branches and Illumos-powered projects (like Manta). We want to compare the "cross-pollination" between the `illumos-` branches. We would also like to get metrics for competing projects (like Linux kernel, FreeBSD kernel, etc) and see how they compare on metrics like the bus-factor. And much more. The Illumetrics team is open to ideas and suggestions in this matter.