Extending audit/uptime success ratios to a more tuneable reputation model

Storj DS&A

Success ratios are good, but limited.

The current node selection process relies on audit success and uptime success ratios to determine which nodes are suitable for use. While these measures provide a good initial guess into node reliability, they leave us without any tuneable parameters, and they do not give any indication of how widely such ratios can vary. We propose extending the current success ratios to a more tuneable reputation model by considering these ratios to be related to the expectation (mean) of the Beta distribution. Doing so allows us to introduce parameters which allow us to tune the calculated reputation score to achieve certain desirable behaviors, while allowing us to apply standard statistical tools in analysing a node's current reputation.

Why the Beta distribution?

The Beta distribution has values that fall in the interval [0,1] and its shape is controlled by two "shape" parameters α and β. The standard measures of central tendency and statistical dispersion (mean, median, variance, etc.) are all described by these two parameters, the most basic of which is the mean μ, which has the value $\mu = \alpha/(\alpha+\beta)$. If α is viewed as (related to) the number of successes and β is viewed as (related to) the number of failures of an experiment (an audit check, or uptime check), this value is descriptive of the experiment success ratio; therefore, thinking of our audit and success ratios as related to the mean of the Beta distribution is an intuitive and natural extension of our pure success ratio calculation when measuring node reliability. Additionally, considering the variance of the distribution gives us insight into how much we may expect a node's reputation to vary from its current value. Furthermore, due to the work by Jøsang and Ismail in [1], we have ways to tune the success / failure updates for α and β so that the resulting reputation score more accurately reflects attributes that we want to emphasize, giving us more confidence in our node selection process.

How do we do it?

The traits described in [1] that we find the most desireable are the forgetting factor λ, and the single value feedback v made possible by a normalization weight w.

- Informally, λ is the "memory" of the current reputation score; that is, it describes how much weight is placed on prior results when determining the current reputation score.
- The single value feedback value v allows us to update both α and β with a single choice of v, which in the context of a success/fail experiment, we define to be -1 for a failure and 1 for a success.

- The normalization weight w affects how severely new updates affect the current distribution. Larger values of w mean larger changes in the shape parameters α and β with each update, leading to a smaller variance in the probability distribution. Informally, this impacts how widely we expect the current reputation score to vary from its current value.

For our implementation, we suggest one minor modification from [1] to how v is selected. Rather than starting with either -1 for a failure and 1 for a success, we propose starting at small (but negative) values for failures, and small (but positive) values for successes, with these values tending towards -1 or 1 the longer a node has been on the network. This has the benefit of "easing" a new node's reputation toward its "true" reputation. That is, for a new node that joins the network, even if it succeeds in 100% of its initial audits and uptime checks, we don't assign a perfect reputation score to this node, simply because it has not been on the network long enough for us to trust that this is an accurate value[1].

Having described the shape parameters α and β along with λ, w, and v, we exhibit the recommended update rule for determining a node's reputation:

$$\begin{aligned}
\alpha_0 &= 1 \\
\beta_0 &= 1 \\
\alpha_n &= \lambda \cdot \alpha_{n-1} + w(1+v)/2 \\
\beta_n &= \lambda \cdot \beta_{n-1} + w(1-v)/2 \\
R(n) &= \alpha_n/(\alpha_n + \beta_n)
\end{aligned}$$

where n is the counter for the number of audit/uptime check attempts and $R(n)$ is the node's reputation after n check attempts. We start $\alpha_0 = 1$ and $\beta_0 = 1$ for two reasons: first, because 0/0 is undefined; second, this assigns new nodes a reputation score of 0.5 (or 50%), requiring them to demonstrate successful audits and uptime checks to build towards a better reputation score.

Recommendation.

We suggest that the Beta distribution be compared to success ratios when describing node reliability given how closely its mean resembles an experiment success ratio. Probability distributions are more descriptive than single value metrics, and standard measures of central tendency and statistical dispersion may be used to give better estimates than single value metrics on how nodes are expected to behave in the future. Furthermore, the Beta distribution is a commonly used conjugate prior in Bayesian inference, and using it to desribe a node's reputation allows us the freedom to consider more powerful statistical techniques when describing node reputation in the future, if such a need may arise.

References

[1] Jøsang, A., and Ismail, R. The beta reputation system. *In: Proceedings of the 15th Bled Conference on Electronic Commerce* (01 2002).

[1] For suggestions on how to scale v with time, see a Member of Your Local Storj DS&A Team (we don't want to complicate this description any more than we already have).