Pricing granularity

With a worth based approach then I have a strong incentive to: -

- reduce the operational cost of the project because the cheaper it is then the more profit I make.
- provide reliability because if the system went down, I wasn't making any money.
- ensure the system maximises the value metric. In the case of LFP then this metric was "generating leads".

But I also had questions on where to invest. In the case of LFP, it was doing very well (this was prior to the budget shenanigans) and so I decided to look at investing an additional $100K. But where do I best put the money? Improving the site reliability? Reducing the operational cost of the application through better code? Maximising the number of users through marketing? Improving conversion of users to leads? Which choice brings me the better return? This is a particularly tough question to answer if you can't effectively determine operational cost of an application beyond hand waving or if other data is also guessed at.

One of the huge benefits of Zimki (our platform as a service play in 2006) was not only its serverless nature and how you could simply write code through an online IDE but also its pricing granularity was down to the function. This was no accident as I had a genuine need to know this. Any application is nothing more than a high level function