Grid Computing in the Investment World

ADAPTING TO THE NEW CLOUD LANDSCAPE

Uros Pavlovic (Technical Writer): **Most financial services are relying on HPC (high performance computing) infrastructure grids for various aspects such as valuing portfolios, calculating risk, and so on. Are there other common scenarios where grid computing provides, shall we say, an instant advantage?**

Marko Djukic (CEO): Sure, there are common underlying common factors, which help make a case for deploying something across a grid computing platform. One of them is, obviously, to scale anything that has flexible workloads; running small workloads, or running a huge workload that scales out across a massive resource, or not running at all, which is also kind of important the intermittency, it might be bat operations, which come into play on a once-a-night, or once-a-week basis, and rest of the time they don't actually run, and you can distribute this across the grid when needed. This leads us into the following factor, which is parallelism: the benefits in this scenario is that operations are run in parallels, hence you're chunking up the compute so that it can be distributed across multiple nodes and done in parallel streams. That may not be possible in some workloads, and that is a key common factor because grid computing is all about getting as many nodes as possible to work in use and get to the operations as quickly as possible. What all of this then applies to is kind of irrelevant – so, it doesn't matter if it's trading, portfolio analysis, finance, backend accounting, and we've even seen it in use cases such as sales, or at the end of the day, anything which improves the business. What's crucial is tackling a problem in a matter problem in a matter of hours or minutes, which would otherwise take days to solve. Alternatively, if it's hours, it can be reduced to minutes and seconds.

Again, if the result is unlocking massive potential for the business, then it makes sense to employ grid computing, because it brings answers faster. There's also another element, which is cost. In most cases it does turn out to be much cheaper than running dedicated resources or monolithic resources or trying to scale up vertically with the resources, as opposed to horizontally scale across the clusters.

UP: Sure, that make sense. So, what you're saying is that this saves a lot of time and improves performance, right?

> **MD:** Well, it can do, but it's not necessarily always about saving time. In most cases it is. But as we touched on, the cost-effectiveness of it can come into play. One of the use cases we've had a number of times and also quite recently, was taking a large monolithic compute structure for portfolio analysis and breaking it up into microservices that can actually run across a grid computing engine behind the scenes and that cut the cost for the entire compute to the 10th of what it was before.

UP: So, I guess you could say there's a big challenge to all this, especially when it comes to grid computing at scale. As we've mentioned in our earlier chat, when companies decide to go with cloud adoption, the biggest thing for them is knowing what to pick – which cloud provider. From your experience, our team of experts can supply a solution and the best scenario based on our earlier, extensive experiences. Can you go into more detail as to how Hentsu "picks things apart" to get the best possible recommendations for clients?

> **MD:** Sure. Well, there is no single answer. Each approach is quite unique. What we're looking at is often where the client is coming from in terms of the current setup, where they are trying to get to, what kind of data they dealing with, what kind of businesses value are they trying to bring at the end of the day. The greatest challenge is migrating some of the existing workloads if they are already on-premise, and pulling that apart, looking at the legacy setup, and trying to figure out how that maps out to the cloud. The challenge gets greater if the mandate is to keep it hybrid. It's doable, but it's a balancing act of what are you trying to use. Is it the burst-out capability into the cloud? At which point, your on-premise is the master; there is always that challenge of where does the data reside? What's the golden copy? Even before getting into what kind of functionality can we use between both the on-premise and the cloud. Fundamentally, they are different "beasts." You can get into some common layers, especially around Docker, but they are two different platforms.

UP: Hm, the cloud providers then jump in to save the day when legacy code is involved?

> **MD:** Well, there is a flip side and other approaches to bridge it the other way around. This is where the cloud providers offer 'outposts' (as they are defined by Amazon), which are kind of on-premise stacks of the cloud environments. Mind you, these use cases are not that common. What we've seen more often is cases where we are using serverless approaches. It's a challenge when there's legacy code involved; how do you refactor that? How do you get that into a state, which is a kind of serverless

> "
> There is a fine line to walk between too much locking down and too much freedom as well.

> environment? Usually they are quite different beasts. And what may have worked well on-premise, with stand-alone instances, servers and so on, is a complete rewrite for doing serverless. But again, that comes down to unique approaches to each client, looking at what environment they have, how we tie into the data where it says what compute they're going to go after and what code they're trying to run.

UP: Right, it's trying to optimize and improve things and in each individual case it's a challenge, but it's always achievable.

MD: Yes, there's always repeatable patterns that we apply, but it's always tailored to what we have as the underlying challenge.

UP: Let's move a bit towards how things are in IT. As of late, it's been noted that many IT organizations are using different analytic tools on separate clusters, and that this can cause a so-called "cluster sprawl." Cloud computing is something of a turn-key solution in that regard, thanks to various cloud-based PaaS offerings.

MD: Interesting question, 'cluster sprawl' in some ways it did help along with the ease of spinning up resources and being able to access different cloud providers, different cloud platforms, different compute platforms and features and functionality. We've seen it with a number of clients – the ease of spinning up resources and being able to delegate a lot of workloads to end-users that are trying to work on the data, even non-technical users, in some ways, handing all this power to them they can pretty much go wild. These turn-key solutions they're great because they have the democratization of the technology and empowerment of end-users, but like the regular resources sprawl and cloud sprawl these clusters can themselves get multiplied exponentially. We try to mitigate this through templating, reusable patterns, and tight business guardrails around what users can do – this helps mitigate the cluster sprawl.

The other key approach to mitigating this is tight configuration management, we extensively use Terraform to keep on top of our infrastructure. At the end of the day, you do need to balance it out with trying to empower the users in getting the right business value as quickly as possible. There is a fine line to walk between too much locking down and too much freedom as well.

UP: Yeah, from what I'm getting, it is a challenge to have it all set up, but once it's set up everything runs smoothly.

MD: Sure, but you're always innovating. It's the beauty of the cloud platforms, that they are constantly evolving. What was used to great effect a year ago, is suddenly replaced with something new, fancy, and shiny. You want to get that into the user's hands quickly. So, it's finding those methods to get them out to the users through such patterns and repeatable templates that we have and using the configuration management tooling as well.

UP: In regards, to compute and networking (network bandwidth and so on), let's talk a bit about on-demand elastic compute power.

MD: Sure. The interesting trend during last few years has been the serverless approach, and that's underpinned a lot by the Docker framework. We've seen this used both for the ad-hoc workloads; you need something which just answers an APi que-

> "
> You are always innovating, and that's beauty of the cloud platforms - they are constantly evolving. What was used to great effect a year ago, is suddenly replaced with something new and shiny and you do want to get it into the user's hands very quickly.

ry once a day; it can be spun up as an ephemeral resource that just responds to a query and then disappears after that. But it also lends itself well to the massive scale outs. We're talking about that elasticity the cloud is famous for, the utility computing that's often talked about; the fact that we can take if from nothing to tens of thousands of nodes in a matter of seconds. But it obviously cannot work without certain considerations about networking, data access, security, what are the intermediate layers to the other workloads that work with this elastic compute. Because you can easily solve one problem where you scale out to 10,000 nodes, but you're piped back to the office to get it to the users who can't handle it. So, there is a kind of holistic approach to trying to marry up what is the result that we're trying to get into these hands. And obviously there's the growing of the Function as a Service approach where you're not even thinking about the serverless approach, you're not thinking even the Docker containers, you're just looking at the code and getting it out, doing the compute on that. That's something that's both growing as the cloud provider function as a service, but also as this kind of hybrid approach of running your own frameworks across the Docker and the underlying workload you leave behind it.

UP: If I'm not mistaken, one of the reasons why financial services are shifting grid computing to the cloud is because of automation and being able to handle single projects with massive resources and computing power, all working towards the same goal. We can talk about that for a bit, and other benefits of cloud-enabled grids.

> **MD:** Sure. We got to start with the basic premise of what business value we're trying to bring to a business. There are elements of the scaling, throughput, and provisioning resources, and once again coming back to the configuration management – that becomes very clean, very tidy as well. But also, it does get into some of the security; this element of it is quite key, especially with some of the financial data analysis and IP that's being put through these grid clusters. The other aspect that's been coming up recently in some engagements, you mentioned enterprise-wide visibility, but it's broader than that, it's access from anywhere, so we can have the resources in London working with the same data and resources in Singapore with very little drawbacks in terms of latency and access to data. Just being able to use this global platform and the scale of it. That allows more workloads being used by distributed teams as well, and not necessarily trying to have a single location and single machines or servers – it's using that cloud compute wherever it is and obviously you come into those challenges around data and the access to it and where it resides, but we've seen it solve quite easily that we can enable users to work across time zones, across regions and use the scale-out across multiple cloud locations as well as cloud platforms. That's the flexibility of having this cloud power under your fingertips.

UP: Let's use this opportunity to talk about Azure Batch and how it works and its biggest advantages in your experiences at Hentsu.

> **MD:** We've seen Microsoft picking up the pace on rolling out some their grid computing and AI and machine learning and business intelligent tools quite rapidly. We've rolled it out in a number of different scenarios both for clients and for us internally. We've built out reusable patterns around data ingestion and analysis, specific to the asset management workloads. Internally we use it for different businesses cases; one of them is the financial services that we provide around accounting, billing and so on - all of that is based off of Azure Batch. Because at the end of the day, you talk about workloads that bring value to the busines, and not necessarily exactly what they are. In terms of client scenarios these are around data ingestion, portfolio analysis. What we've seen quite encouragingly is the rapid uptake of these Azure Batch solutions purely around the ease and cost-effectiveness of it. What we've been able to do is stand-up these environments, and hand them over very rapidly to users with minimal ramp up. That's what's been successful. A lot of the functionality in this platform as a service, easy-to-consume type of packaging.