Introduction

Hi and welcome to module one. Now before we start, let's have a quick look at the course outline in some detail. So actually what will you learn in this training? First we'll have a look at what [inaudible] is and we will see that it is not just a file system. Then of course we'll check with a file system really is and we will talk about I note and blocks and the relationship between them. Then WAFL rights in which we will take a close look at what happens when a consistency point is written and how you can monitor that. And also what the reasons for consistency points can be. Then Snapshot Management in depth in which we will create and resource snapshots, files, and even partial files. What'd your five will be on space management and efficiency. What happens when snapshots and data collide, for example, and how can you deal with that?

And of course we'll discuss compression, compaction and deduplication. What you'll 60 is on performance and performance tools. We will first discuss common terms like throughput, latency, and others. Before we have a look at command line tools like sis, that and steadies. Then we will download and install and a box and oncommand unified manager. The last module is on working with these two tools. We will be creating our own performance dashboards in and a box and we'll also have a look at oncommand unified manager and had to create reports, manage policies, alerts and events and finally there will be some considerations in which we will discuss 10 things that you may want to take into account when managing a Netapp cluster from a performance point of view and also the modules. Obviously we'll have one or more demonstrations in which we will do what we talk about in this first lecture.

We will have a look at a WAFL is and where we can place it in the cluster environment. We'll also have a look at how a note is split up into three different levels through which the data will pass before finally ends up in a volume on disk. So we will see what happens when data enters a node before it is picked up by WAFL. After that, we will explore the data module that takes care of acknowledging the client and finally that data module, we'll write the data to the volume. Now WAFL is often referred to as a file system that NETAPP uses and [inaudible]. This is not untrue, however, WAFL is much more than just a file system. Actually it would be more appropriate to say that WAFL is a disk management tool that also has a file system on board, so the management of disks, aggregates rate groups and MV Ram is all part of the WAFL environment.

What was not one of a kind. There are other examples of disk management tools with a file system. I'd like to mention two of them is the interface and Patri. S or better of this ZFS was initiated by sun microsystems and be tree of was by Oracle for Linux environments. I would love to state that WAFL is the best, but that will be just my opinion. What I can say is that willfully is the oldest and maybe the most evolved and proven environment, but again, I am not the judge of that. Now before we can dig into WAFL, we have to place it in the architecture that makes up an alltech cluster and node in the cluster is logically divided into three parts. This is done with the use of software modules that all have their own function.

Okay. First, there's the top layer that is formed by the network module and the scuzzy module. When client data enters a node, that data can either be nas data or send date. If it's Nesse data, then it will be processed by the network much. If it sends data, then it will be received by this cause of the module. The next step is that it's copied to the cluster session manager. This is the module that checked where the volume is too, which the data should be written. It's important to realize that the session manager functions at the cluster level. This means that all session managers in a cluster, we'll have the same information, so the session manager knows whether a block of data is meant to go to a volume on the local node or to a volume that is hosted by another node in the cluster.

The configuration information that is managed by the session manager is kept in sync in the cluster via the cluster interconnect. So the data that enters the node may be sent out again to another note based on the location of the volume that the data has to be written to. If the data is sent to another node that's also done via the cluster interconnect. So all of this time WAFL is unaware of the data because it first has to be determined for which no the data's actually meant. So client data enters the node is then copied to the cluster section manager. The cluster session manager determines the location of the volume. In this case, the volume is local to the node. So the data is copied to the local data module.

Again, client data enters to node, uh, it's copied to the cluster session manager, but now the volume is hosted by different node in the cluster. So the plus the session manager determines that the data should not be stored to a volume hosted by the local node. So it sends the data via the classroom to connect to the correct note. So the data is then copied to the data module in that note. Now let's have a look at the data module. Once the data has entered the data module of the node, it's not going to go anywhere else. Then to a volume that is hosted by that particular note. Before that the data will be processed by WAFL in the data module of that node. This is about NV rams storage efficiency and the final creation of what we call a consistency during which all data of a single right will be written to disk.

Don't worry, storage efficiency as well as the creation of a consistency point and how that exactly takes effect will be dealt with in another module. And at the end of this section will also have a little demo. So when data enters the data module, it is instantaneously copied to MV Ram. If it concerns a single node cluster, the client will be acknowledged the minute it is an MV Ram. Since MV Ram is battery backed, the data will survive a node reset so it can be safely acknowledged to the client that the data is written. Even though it's not on this kit, if the node is part of a multi node cluster, then the date will be copied to the pot as MV Ram via the HIA and to connect before being acknowledged to the client. After the data has been copied to MV Ram. There are two options.

If storage efficiency is enabled, then the data will be processed for compression, deduplication and compaction. These are terms that we will deal with later, but probably or most likely you're already familiar with them. Uh, also you can have a mixture of those, right? So you could enable, um, did a blockade and not compression. Uh, you can enable or disable compaction. Um, so you can have a mixture of these different techniques depending on what you want. Finally, the data will be written to the volume by means of a consistency point. Again, we will look at efficiency and consistency points later on. So data enters the data module is copy to MV Ram and acknowledged to the client. Then it is either processed by a storage efficiency or not. And finally the data has to be written to disk. Now it's very important to realize that WAFL uses four kilobyte blocks to write to disk.

As you probably know, the volume is part of an aggregate. The aggregate is a collection of raid groups that have a parody type, apparently may be arrayed for raid DP, which is dual parity. That means to parity disks or ray tech, which is three parity droughts. So it's either single dual or triple parity. And this example, we see a rea group of only seven disks, two of which are parity drives. This is a very unlikely to set up, but it's just to demonstrate the flow of data. So this is not an actual setup. You will not, you will not run into this in real life. When the consistency points at hand, then WAFL will first create a stripe, which is called a tetris block. Then the tetris block is copied down to the raid manager that will create the parity blocks and add them to the tetras book.

And finally the strike will be written to the raid groups and a transaction to complete the consistency point after that. So after the consistency point is complete envy, Ram can be flushed and the procedure will repeat itself all the time so it's interesting to check when this habits, there are a number of reasons for that and we will have a look at that in the demo as well. But firstly I'd have a look at the bustle reasons for a consistency point to take place. So sooner or later the data will have to be copied from memory to disk before MV Ram can be flushed. First is time by default, WAFL checks every 10 seconds. Whether data can be written to disk. If your system is not very busy, that may very well be the case, so every 10 seconds or so there will be a consistency point which is based on time.

If there is data to be written, if you're a system does have a bit of load, then there are other reasons for generating a consistency point. For example, if memory buffers reach a certain threshold, this is called a high watermark and that will generate consistency point as well. The creation of a snapshot will also generate a consistency point. This is important because when the data's been acknowledged that the client, the data is still only in a memory and in envy ramp. Now creating a snapshot should reflect the client's view. So a consistency point should take care of that. Another reason is when your MV Ram is full or half full, I should say this is not bad because during the writing of the consistency point, the other half of MV Ram will be used to copy the data from memory to MV Ram. So then it can keep acknowledging the client related to half full.

And the ram is the number of entries in Inverell indexes are kept for different items like volumes. If the threshold for the maximum number of entries this past, that does not mean that Andy Rem is half full, but still the consistency point will be generated. And there may be other reasons. We'll have a look at some of those in a second. But, uh, there are multiple reasons why you can get consistency point. The bad thing is when both of the NPRM halves are filled up during the writing of a consistency point. So that means that Mv rant is completely full and from that moment on, your clients can not be acknowledged. So they will have to wait. So you will have performance loss because every ram can only be flushed after the consistency point is complete. So the minute the first consistency point is complete, the second we'll start, this is called a back to back consistency point or a back to back seat p and these are not good for performance.

Colon in the output would mean then a consistency point that started since the last interval is still not yet finished and the hash mark says that the consistency point that is running right now is still not finished and the other half of Mbm is full. So the next CP will be your back to back CB again back to back. CPS are not good. So for now just keep in mind that are back to back. Consistency point is not good around it. It's very bad. Not if it happens every now and then, but if it is continuous for a longer period of time, then it is very likely that the writing of the consistency point takes longer than is good for you. Now, to avoid back to consistency points, you could start thinking of some of the following solutions. It of course, it depends on what the real reasons are.

So you might think of changing the workload or you could grow your aggregate or maybe widen your stripe. So the larger your stripe, the more I you will get. And you could also think of moving volumes to other aggregates to take away some of the load. And if you run, how does drives, you might consider adding SSDs to create a book, which will also maybe improve your performance. Depending on the type of Iot you have. Or You could go to all flesh, which means you'd have SSDs only. Now let's do a demo on monitoring consistency point types. Uh, we will be using a two Linux vms and one cluster.