zisoftdocs Documentation

Release 3.0.0.1

zisoft

Mar 19, 2020

CONTENTS:

CHAPTER

ONE

ZISOFT AWARENESS SIZING

1.1 ZiSoft Awareness HW Sizing

ZiSoft Awareness is a flexible web-based application that is designed to accommodate both small-scale implementation in a single department and large-scale application in organizations with thousands of users. Large-scale execution, however, requires careful planning and an adequate setup to accommodate the processing, bandwidth and storage needs of the system.

*** General Environment Sizing Notes ***

ZiSoft Awareness is a data intensive platform. As such, the majority of the workload is usually dependent on the database environment. A properly tuned database environment is an absolute requirement for a production environment in ZiSoft Awareness. Many of the sizings detailed will be focused on data storage.

ZiSoft Awareness is used in various enterprise environments with different use cases. The Best Practices described below is not a one size fit all solution. Please take your use cases into account and use your best judgement to determine the best architecture solution.

ZiSoft Application Server Sizing and Deployment

The key to any server sizing project is to determine and understand the guidelines and types of workloads involved when running on a large-scale. Read the following guidelines below to help accurately evaluate your system requirements:

Process Capability and Memory

- Estimate the number of cases to be created per day. The number of processes created in a day has an effect on the performance of Zisoft Awareness, which is why its important to have an estimate before sizing the server.
- Consider the total number of nominal users that are expected to be using Zisoft Awareness (250,000 users)
- Use a web application server tuned for medium CPU and memory capacity.
- Depending on the level of redundancy security the customer requires, the production web application server cluster should at LEAST support one server being unavailable. This means the other remaining servers should have surplus CPU/RAM capacity to handle the extra load if this occurs. For example, if there are two production servers, ideally 50% of one server's capacity should handle 100% of the volume at peak, at a minimum.
- Consider one CPU core (or hyper-thread) to every 4 gigabytes of memory.
  - For example, for a web application server that has 64GB of memory, a 16 thread/core processing environment is suggested.
  - Assume a maximum allowed size per server to be 128GB of memory, with a minimum of 32 thread processing.
  - Assume a minimum base frequency of 2.4GHz
- Consider dedicating 64MB for one active user at a time.

  - This value might change depending on memory requirements for certain processes, etc.
  - This value is based on the memory_limit in PHP's configuration file and should also account for the overhead web server memory requirements.
  - For the total amount of users, ensure that there is a buffer that allows for 30% spikes at any time.

Network Speed

- These servers should have gigabit network interfaces and should be local to the shared resources of the load balancer and dependencies (file server, databases and caching). Ensure that your router and Internet bandwidth supports your expected incoming and outgoing traffic.

*** Standard Calculating Hardware Requirements***

The sections below provide guidance on how to calculate hardware requirements, taking various considerations into account.

*** Total Memory Required Calculation***

(pu * mmpu * 1.3) / 2048 = tm

Where:

- **pu:** Peak number of Active users.
- **mmpu:** Maximum amount of memory in megabytes per user.
- **1.3:** Allows for 30% growth/spikes at peak.
- **tm:** Total memory required for this environment.

Total Minimum Servers Required Calculation

(tm * rl) / 2 * mps = sc > 2 (sc rounded up) or rl = ms

Where:

- **tm:** Total memory required.
- **rl:** The redundancy level specified (a minimum of 2). There is always a minimum level of redundancy required.
- **mps:** Maximum amount of memory per server the customer can acquire (a maximum of 128GB is recommended).
- **sc:** Total server count. Round up sc to ensure an adequate number of servers.
- **mw:** Minimum number of servers required.

Memory Per Server Calculation

(tm * rl) / (2 * ms) = (mmps rounded up to whole number divisible by 8) = mps

Where:

- **tm:** Total memory required.
- **rl:** Redundancy level required by customer.
- **ms:** Minimum number of servers required.
- **mmps:** Minimum amount of memory required per server.
- **mps:** The value of mmps rounded up to a whole number divisible by 8.

*** CPU Core/Thread Per Server Minimum***

(mps / 4) = cps

- **mps:** Memory per server.

- **cps:** Number of cores/threads per server.

*** Data Transfer***

(users * adt / cput) = dt

- **Users :** Number of Active users
- **Adt** = (high transfer session data * 0.33) + (Avg transfer session data * 0.33) + (low transfer session data * 0.33) / 3 = (100 MB * 0.33) + (40 MB * 0.33)+ (1 MB * 0.33) / 3 = 15.51 MB
- **cput:** cpu session tme out = 30 s

Estimation HW Sizing

Estimate HW Sizing when Active users = 5%

Active users = Concurrent users who access the application at CPUT (server CPU Session time out 30s)

Active users = Total number of users * 5 / 100 = N

Estimate 5% has an estimated n simultaneous users at their peak time. They've stated they can only support servers with a max memory of 128 GB. They request that they have a redundancy level of 3 (30 % of their fleet could be down at any time). Memory per user will be 64MB).

1.2 ZiSoft Awareness Load Testing

Load testing is performed to determine ZiSoft Awareness behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation. When the load placed on the system is raised beyond normal usage patterns to test the system's response at unusually high or peak loads, it is known as stress testing. The load is usually so great that error conditions are the expected result, but there is no clear boundary when an activity ceases to be a load test and becomes a stress test.

Installation

You can install Jmeter from (https://jmeter.apache.org/download_jmeter.cgi). We used the binary version of Jmeter 5.0, you will also need Java 8 or 9 installed on your pc as a prerequisite to launching Jmeter.

Launch Jmeter

After you install the binary version, go to bin folder, then run jmeter.bat as administrator to avoid any error. Now we want to check the server performance for 250 concurrent users who will go to the Awareness login page, authenticate with the correct credentials, watch a lesson, and finally submit quiz. please follow the below steps to execute the mentioned scenario

Right click on "Test Plan" –> Thread (Users) –> Thread Group

Thread Group Form Content:

Thread Group is a set of threads/users executing the same scenario. Number of Thread (Users) are the number of users that will execute the scenario. Ramp up period (in Sec.) is the time wanted to get the scenario executed by ALL users. Loop Count represent how many times do you want the scenario executed.

So in our case, we will make Number of Thread = 250 users, and the Ramp up period = 120 seconds "Two Minutes", and Loop Count = 1.

Browser Proxy

Now we want to adjust the browser proxy so the Jmeter can listen to it. In case of Chrome web browser…

a.Go to(chrome://settings/?search=proxy) b.Open Proxy Settings c.In the connections tab, open LAN settings d.Enable the proxy server, Address : localhost, Port : 8080

Recorder

Back to Jmeter, Right click on "Test Plan" –> Non Test Elements –> HTTPS Test Recorder

This the tool that we used to record the desired scenario, there are alot of tools such as Blazemeter browser plugin and some built in plugins. please feel free to use any of these tools.

HTTPS Test Recorder Form Content:

Port : This is the port number in which the Jmeter application will listen to, through the browser to record the scenario. HTTPS Domain : the domain that the Jmeter will listen to. Target Controller : Where you want to put your recorded browser components "The GET & POST requests"

So in our case, we will Port Number = 8080, and the HTTPS Domain = localhost or you can leave it empty, and the Target Controller = "Test Plan –> Thread Group" which we just created.

Now you can press START button, and execute the required scenario through Chrome, then press STOP when you finish.

Clean up Thread Group

You may have noticed now that there are alot of requests under the Thread Group object, which are all the GET & POST requests that your Jmeter recorded since you pressed START till you STOPPED the recording. You need to delete some of these requests which are not neccessary to trace its server responses such as the GET *.png files or *.CSS files. In our scenario, we deleted all and kept the main 4 requests, you can check them in the JMX file in the stress folder.

Listeners

A listener is a component that shows the results of the samples. The results can be shown in a tree, tables, graphs or simply written to a log file. It listens to the performance of the server.

For our scenario, we need to add 3 listeners ("View Result Tree" - "Aggregate Report" - "Graph Results")

Right click on Test Plan –> Listeners –> View Result Tree Right click on Thread Group –> Listeners –> Aggregate Report Right click on Thread Group –> Listeners –> Graph Results

The View Results Tree listener displays samples that the JMeter samplers generate, and the assertion results that are related to these samples (USERS). This listener displays the samples in the order they are generated by the JMeter script, and provides parameters and data for each of them. For instance, for each sample, the HTTP sampler produces the View Results Tree listener provides the request parameters, response parameters and the response data. This is displayed under the corresponding tabs: sampler result, request, and response data. The Sampler result tab contains the response code, headers, cookies and information about time, latency, response size in bytes - separately for the headers, the body and the error count.

The Aggregate Report listener shows the aggregated and statistical data for each sample of the script.The number of times it was executed in the script, minimum, maximum, average response times, percentages, response time, throughput, the number of samples per time unit, Kbytes per second and error percentage. These KPIs are useful for tracking your test performance as well as your system's health and for monitoring trends.

Some of KPIs meaning

- Average: It is the average time taken by all the samples to execute specific label.
- Min: The shortest time taken by a sample for specific label.
- Max: The longest time taken by a sample for specific label
- Std. Dev.: This shows the set of exceptional cases which were deviating from the average value of sample response time. The lesser this value more consistent the data. Standard deviation should be less than or equal to half of the average time for a label.

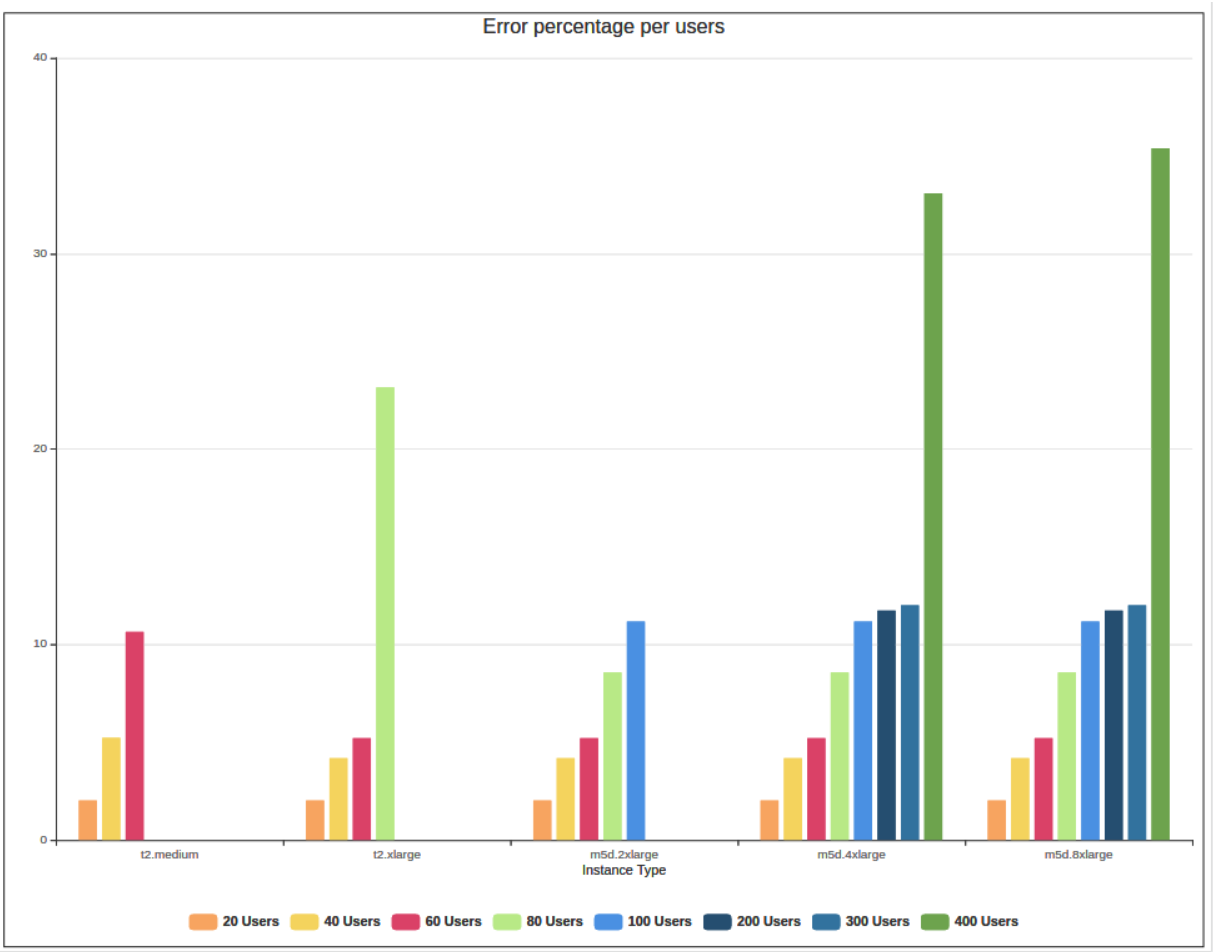
- Error%: Percentage of Failed requests per Label.
- Throughput: Throughput is the number of request that are processed per time unit(seconds, minutes, hours) by the server. This time is calculated from the start of first sample to the end of the last sample. Larger throughput is better.
- Median: It is the time in the middle of a set of samples result. It indicates that 50% of the samples took no more than this time i.e the remainder took at least as long. 90% Line: 90% of the samples took no more than this time. The remaining samples took at least as long as this. (90th percentile)

Graph Result Listener is as same as Aggregate report but it just draws the results through graphing.

CSV Dataset

Now, we want to create 250 users, So the Jmeter can use them to execute the scenario, run "docker exec -it php artisan zisoft:demo 250 1 100", that command will create 250 users in a 100 days duration campaign, the password with each user is "Password123@" Create a CSV file and put USER1, USER2 …..USER249, USER250 vetically in one column. back to Jmeter, right click on Thread Group –> Config Elements –> CSV Data set Config. In this page you should browse the CSV file location and determine the separator between each column in that CSV file and set "username" as a variable that will represent the usernames that you have already put in the first column in the CSV. Go to the LOGIN app. request component, set the username value = ${username}

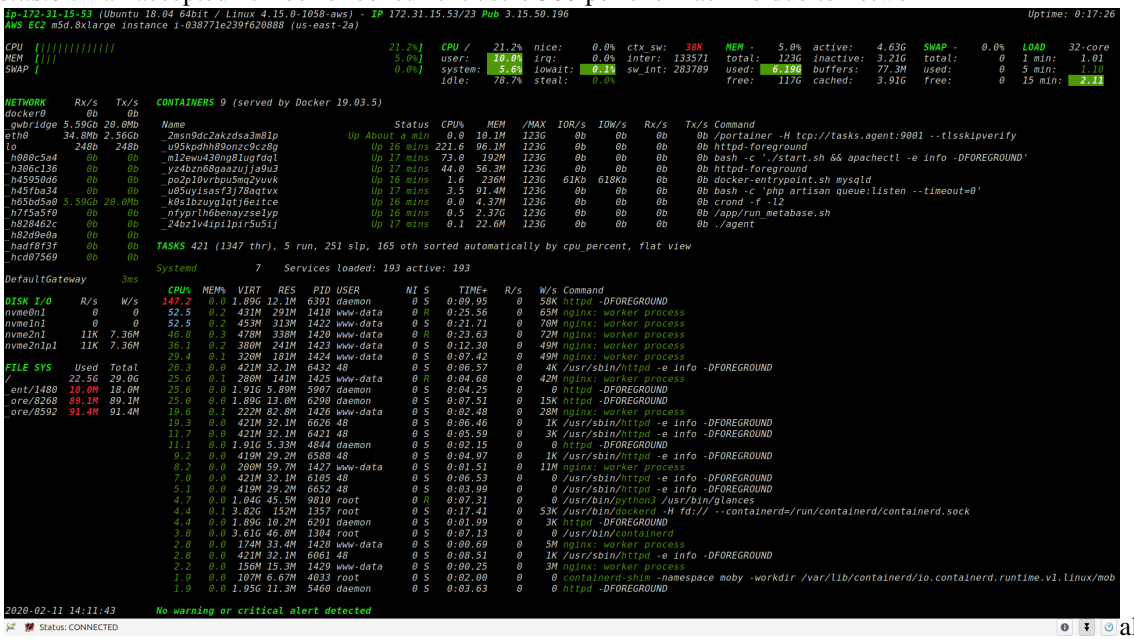
Error Percentage Per users Diagram



alt text

Error Percentage Table Per Instance

| Instance Type | vCPU | Memory | Network I/O | 20 Users | 40 Users | 60 Users| 80 Users | 100 Users|200 Users|300 Users| 400 Users| |——|—-|—-|——|——|—-|——|—-|—-|——|——|——| | t2.medium | 2| 4 | 1 G | 2.02% | 5.22%| 10.64%| - - - | - - -|- - - |- - - | - - - | | t2.xlarge | 4 | 16 | 1 G | 2.02% | 4.19%| 5.21%|23.15% | - - -|- - - |- - - | - - - | | m5d.2xlarge | 8 | 32| 10 G | 2.02% | 4.19%| 5.21%|8.57% | 11.17%|- - - |- - - | - - - | | m5d.4xlarge | 16 | 64| 10 G | 2.02% | 4.19%| 5.21%|8.57% | 11.17%|11.73% |12.01% | 33.38% | | m5d.8xlarge | 32 | 128| 10 G | 2.02% | 4.19%| 5.21%|8.57% | 11.17%|11.73% |12.01% | 35.38%|

Load testing conclusionMax accepted number of concurrent users 300 per one machine due to network traffic congestion 

alt text

Cost Estimation per N users

| Users | vCPU | Memory | Network I/O | No . of Machine | Estimated Cost| |——|—-|—-|——|——|—-| | 20 | 2 | 4| 1 G | 1| 60 USD| | 40 | 2 | 4| 1 G | 1| 60 USD | | 60 | 4 | 16 | 1 G | 1| 175 USD| | 80 | 8 | 32| 10 G | 1| 400 USD| | 100 | 8 | 32| 10 G | 1|400 USD | | 200 | 16 | 64| 10G | 1| 800 USD| | 300 | 16 | 64| 10 G | 1| 800 USD| | 400 | 16 | 64| 10 G | 2| 1600 USD| | 600 | 16 | 64| 10 G | 2| 1600 USD| | 800 | 16 | 64| 10 G | 3|2400 USD | | 1000 | 16 | 64| 10 G | 4|3200 USD |

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search