Distributed Caching Provider

User manual

Version 1.1

Table des matières

[I. Introduction](#_Toc308164934)

[II. Installation](#_Toc308164935)

[III. Configuration](#_Toc308164936)

[A. Configure the client to access your cached cloud](#_Toc308164937)

[B. Cloud settings](#_Toc308164938)

[C. Caching strategies](#_Toc308164939)

[D. Measure and optimize](#_Toc308164940)

[IV. Support](#_Toc308164941)

# Introduction

**Aricie - Distributed Caching Provider (DCP)** drives and leverages a caching cloud from a DotNetNuke installation. It allows multiple DNN instances to share cached information throughout the cloud, allowing your server farm to benefit from the cache mechanism globally.

It bridges the gap between DNN’s caching API - essentially made for local caches synchronization - and distributed caching technologies - such as Microsoft AppFabric/Velocity or MemCached.

Moreover, Aricie - DCP will organize the flow of cached objects between the nodes of an application farm. Its specific configuration parameters enable to design a caching strategy at multiple levels throughout the application.

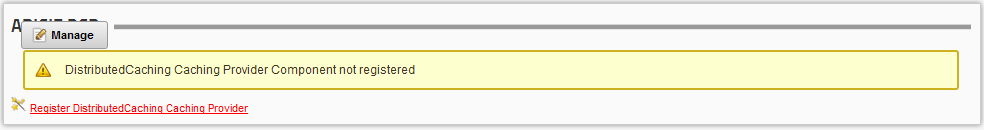
An auto-learning routine makes it easy to initialize the module. This system will log how objects are cached in order to help you understand and tweak the way your application accesses its cache. To speed up the performances of your farm in record time, you can either go GLOBAL – with global optimizations – or LOCAL – with fine-tuned adjustments.

# Installation

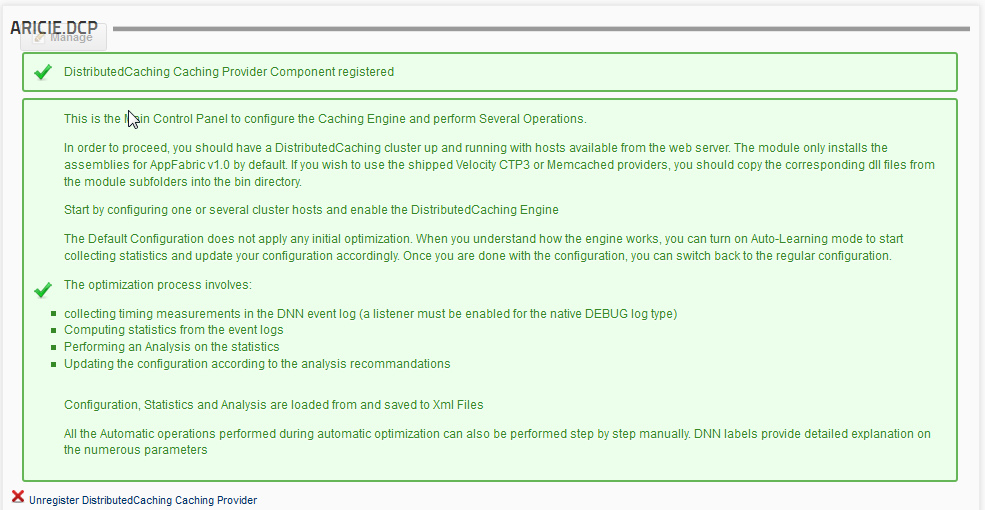
Installing the Aricie – DCP module is handled through the classic DotNetNuke module installation interface. However there are additional steps that you may have to handle by yourself:

* **Working with Velocity or Memcached**? By default only the AppFabric provider is deployed when you install the module. To add the provider you’re interested in, just copy the content of the correct folder from <your website path >\DesktopModules\Aricie.DistributedCachingProvider\external\<provider> to <your website path>\bin

After installing the module, just add it to any page. The module will present its administration interface on this page. First thing you have to do is to register the module to activate it. Just click on the link to install Aricie - DCP:

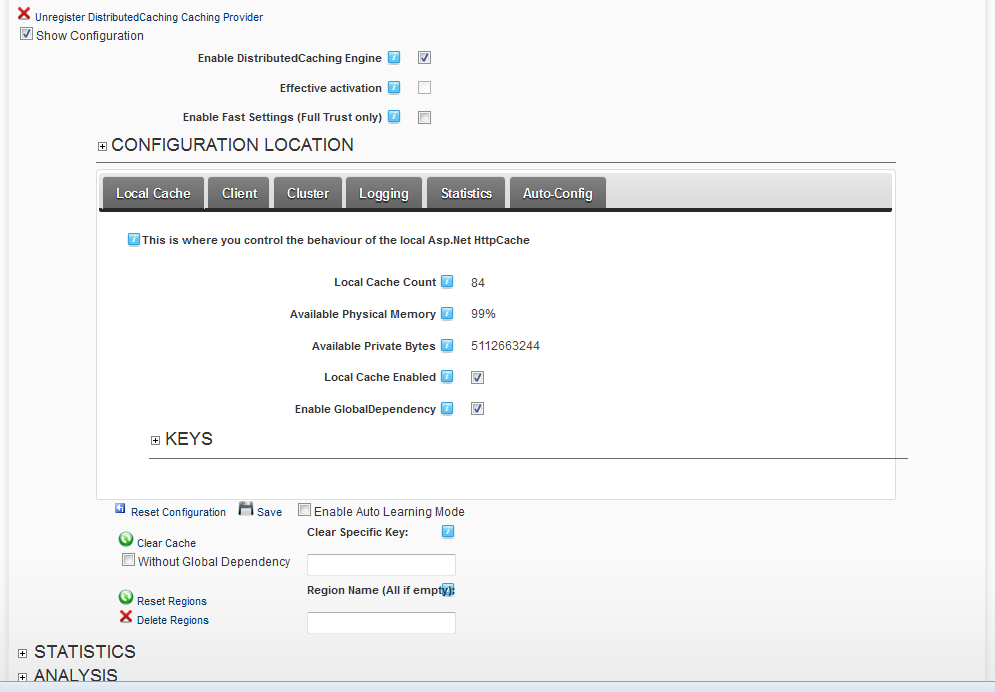


To unregister the module, just click the link that allows you to unregister.



For now let’s continue and configure the module.

# Configuration

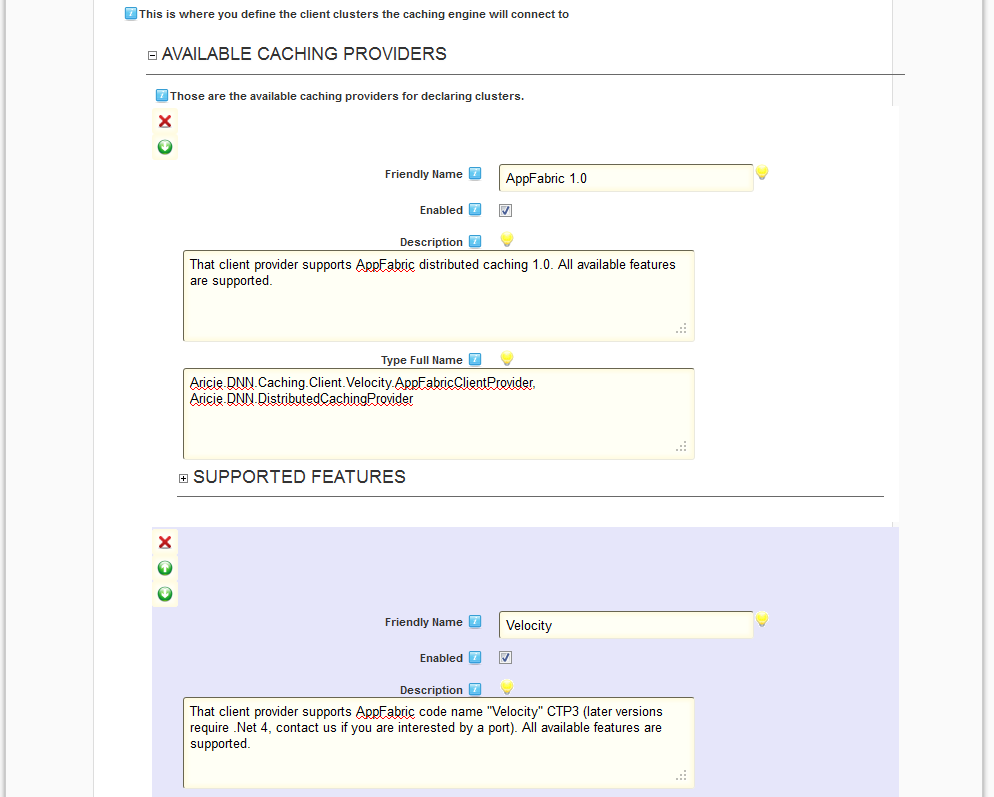


## Configure the client to access your cached cloud

The first step of your installation should be to configure the distributed system that you’ll use for your cloud cache.

Aricie – DCP natively offers drivers for the following cloud cache solutions:

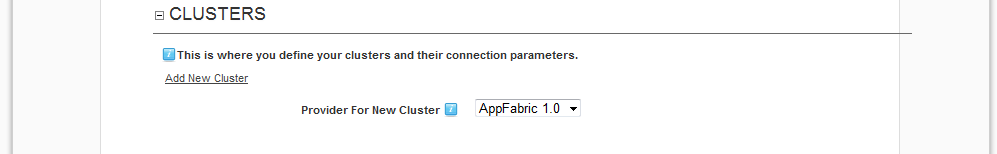
* AppFabric
* Velocity
* Memcache



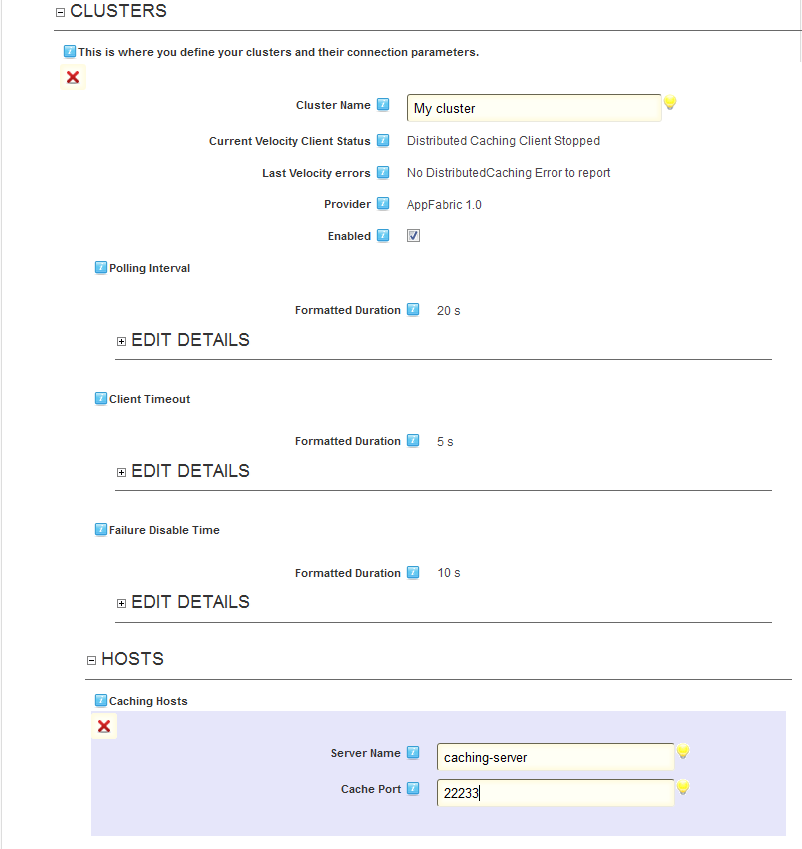
Even though these systems can differ in their feature sets, the Aricie – DCP drivers can emulate the missing features to level their functionalities and offer a unified configuration interface. Let’s add an AppFabric cluster to our configuration in order to connect to this cloud cache.

We’ll assume that you’ve successfully installed and configured an AppFabric instance.

Let’s add a new cluster to our client configuration.



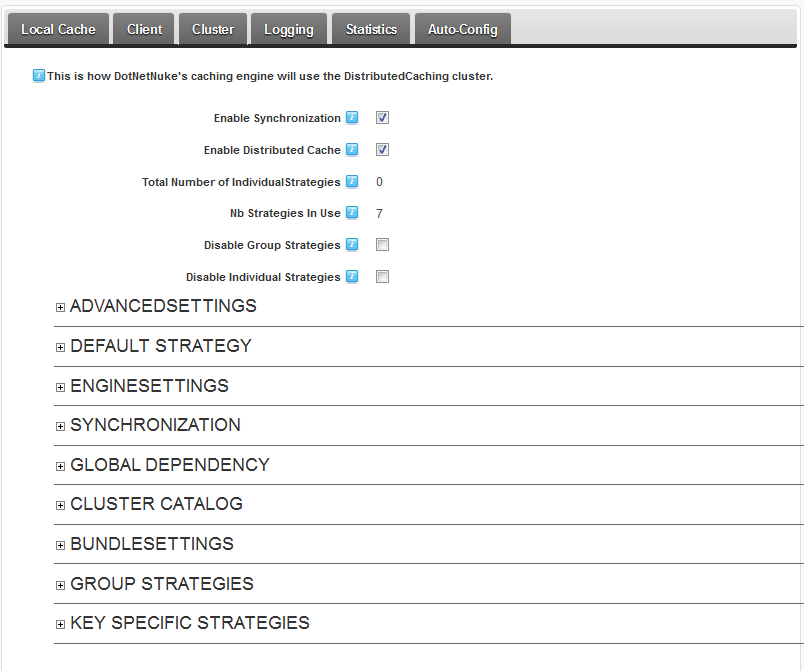
After adding the new provider, you’ll have to fill in some information regarding the configuration of the client for this cluster.



After adding the information for your appfabric cluster, your cache is in the cloud you’ve configured. Navigate your DotNetNuke instance for a bit and check for entries in the cache. (In our example we would for example check the cache statistics with the powershell commandline for AppFabric, but you feel free to use the tool you fancy)

## Cloud settings

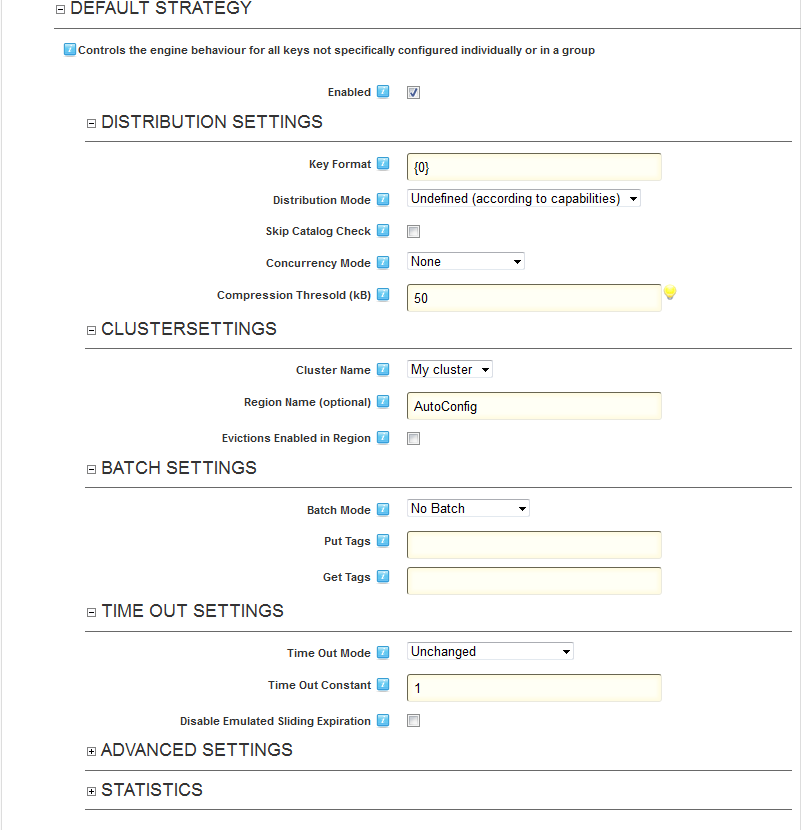
The Cloud Settings configure how DNN interact with the caching clouds.



## Caching strategies

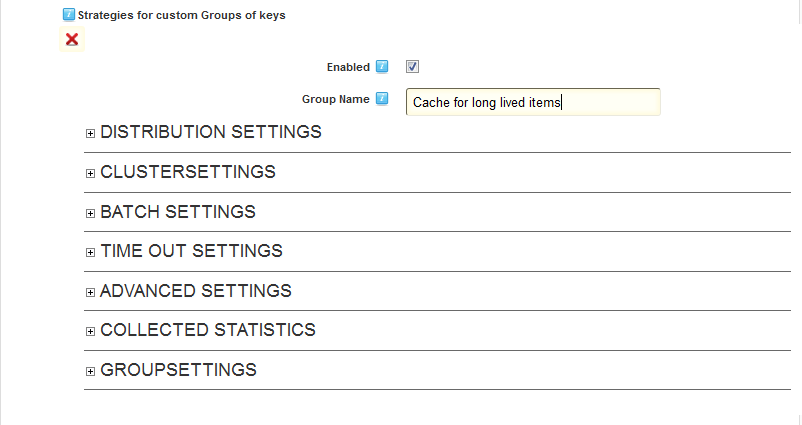
DNN cached objects are processed according to what Aricie – DCP calls “caching strategies”, which provide the configuration for the way the caching engine and the target cloud interact.

A default strategy is associated to unknown objects. This default strategy is conservative in order to preserve the system robustness.



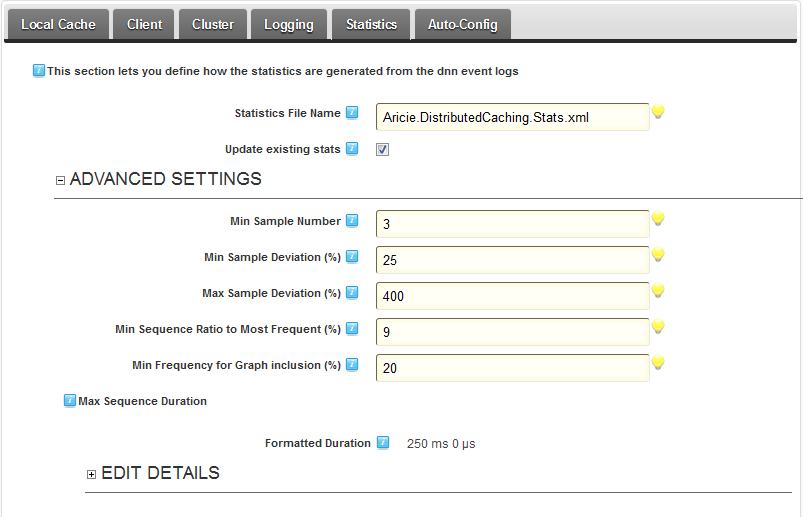
You can also specify various strategies in order to fine-tune the caching behavior of your application. It’s possible for example to configure certain keys in order to keep them from being cached, or cached for a certain duration.

## Measure and optimize

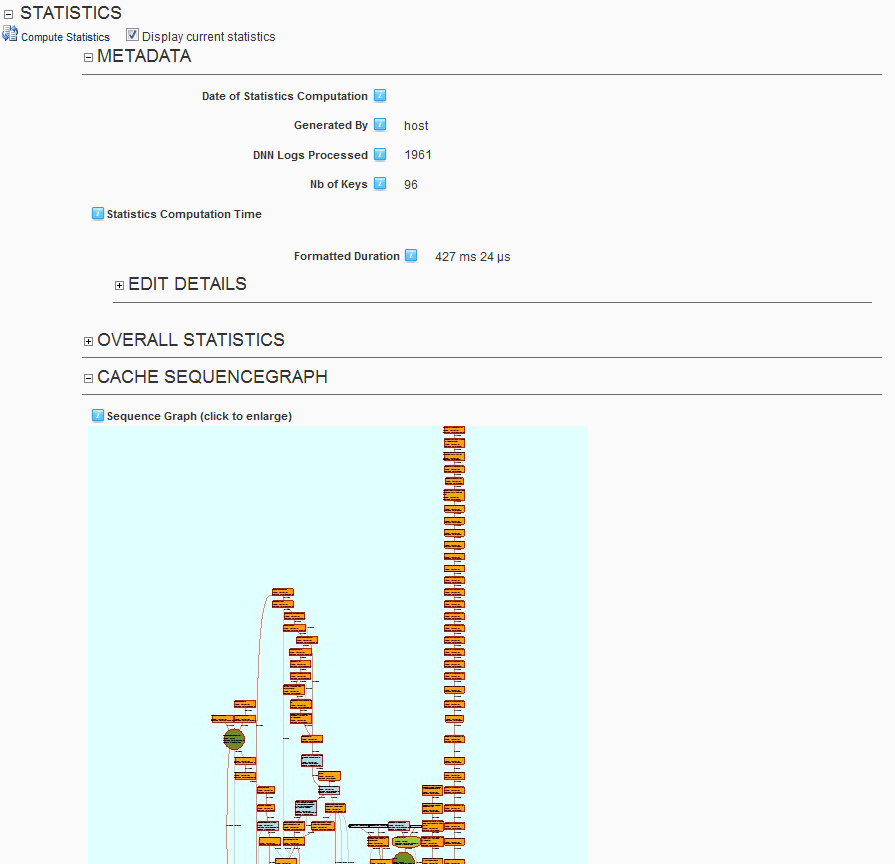
Of course, the bulk of the configuration is a complicated task, one you can’t really take on without good information about the way your application uses the cache provider of DotNetNuke. This information is not easily accessible without knowing the ins and outs of every part of the program you’re using. Without this knowledge you cannot evaluate and configure the various strategies optimizing the overall data flow. 

That’s where Aricie – DCP Auto Learning tools comes in. The Aricie – DCP comes with a set of tools designed to help you build the optimal strategies based on the usage of your DotNetNuke instance.

The first tool gathers statistics about the real usage of your website in order to build a view of the existing cache topology.

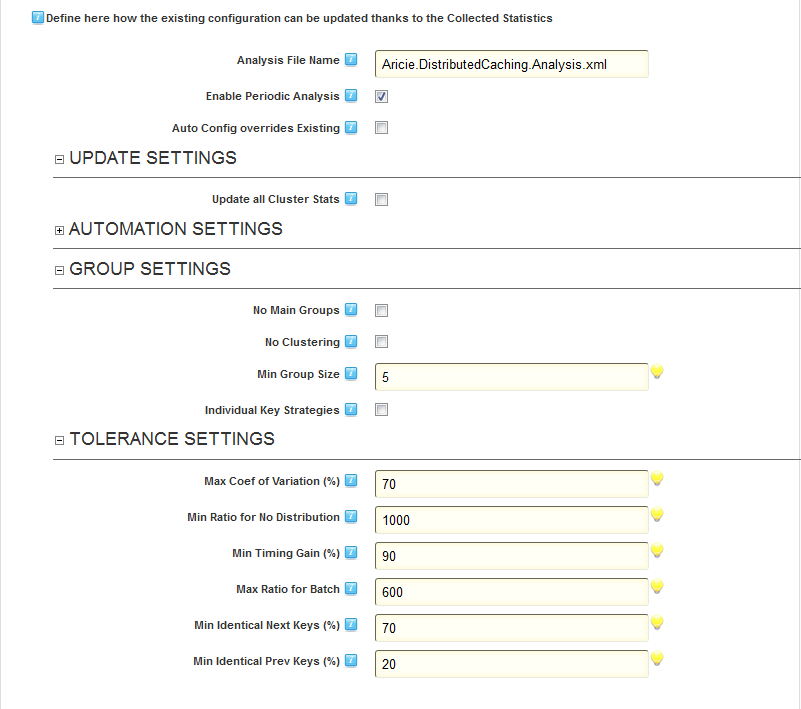


After some measures, the accurate statistics are computed and reported. They provide a broad understanding of how the farm applications behave and leverage the cache.



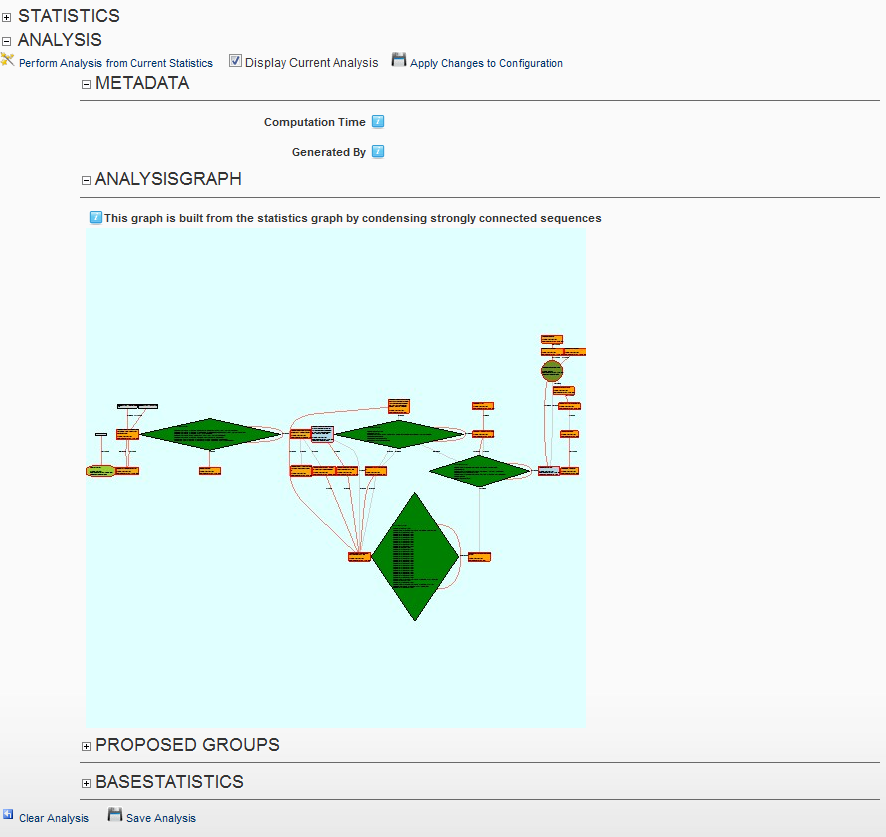
A complete training engine can be made after those statistics by the second tool.

With the automation activated, all possible strategies are systematically investigated by the second tool and the corresponding statistics analyzed.



The resulting analysis uses segmentation and graph condensation to propose an optimal set of strategies covering all observed objects.sSuch strategies can be fine-tuned manually before they apply to the overall Settings.

Several complete rounds of analysis can be necessary before the available set of strategies is considered truly optimal. It is important that during the auto-configuration phase, your site usage reflect as much as possible what you’ll expect during the production phase.



# Support

We hope you enjoy Distributed Caching Provider and the control it gives you over your caching cloud.

For any question and other suggestions, please contact us on <http://www.aricie.com/en/resources/support.aspx>