Similarly, a workflow engine needs to be able to accept a new process starting at any point in time without dropping any existing processes and without making the user wait an undue amount of time to see that the workflow started successfully.

These requirements mean that simply measuring transactions per second presents an incomplete picture of workflow performance. An equally important measurement is the number of concurrent workflows that the engine can process without dropping any and without causing problems for the environment.

In light of all of this information, we directly measured the following metrics as part of our testing:

* Number of workflow starts per second (using two different methods)
* Number of tasks completed per second
* Number of concurrent workflows

Workflow functions well with its default settings, but you can improve and fine-tune performance by modifying specific configuration settings. In an environment that makes heavy use of workflow, you need to pay particular attention to these settings:

1. **Throttle**

Workflow throttle setting controls how many workflows can be processing at any one time on the entire server farm. This setting does not control how many workflows can be "In Progress" concurrently, but rather how many can be actively using the processor. When this number is exceeded, workflow instances that are started and events that wake up dehydrated workflows are queued for later processing. **The default value is 15. This setting is per farm, so the number of front-end Web servers is irrelevant. The impact of this setting is that when** a workflow starts, the number of currently active workflows is checked. If it exceeds the throttle number, the workflow is not started and instead, a timer job is created to try running the workflow later. If the number of currently active workflows is less than the throttle setting, the workflow is started. Throttle is likely the most important setting to get correct in your environment. Unfortunately, there is no magic formula to calculate how you should set this property. The value is highly dependent on your environment and the details of your workflows. If your workflows are all lightweight, you can likely set a high value. (A lightweight workflow performs tasks that do not overstress the server memory or processor and that do not perform an inordinate amount of database operations. Some examples are workflows that simply create and monitor tasks.) However, if your workflow needs to perform more heavy-duty processing—such as creating sites, iterating through collections, or performing more-intense calculations—avoid setting the throttle value too high.

1. **Batch size**

Problems with the batch size setting can be exhibited in one of two ways:

* Processing of workflows takes too long, especially when there are time-sensitive operations. This indicates that the batch size is likely set too low. This problem is unlikely because, as we discussed, computers are operating several orders of magnitude faster than their wetware counterparts.
* Workflow processing is consuming too many server resources in your farm, and the servicing of user requests or other operations are being negatively affected. This problem indicates that too many work items are being released to the timer service because the batch size is set too high.

1. **Timeout**

The timeout setting specifies the amount of time (in minutes) in which a workflow timer job must complete before it is considered to have stopped responding and is forced to stop processing. Jobs that time out are returned to the queue to be reprocessed later. The default timeout period is five minutes, which should be sufficient for most environments. However, if your workflows require more time to start, complete tasks, or modify other workflows (especially when running under load), you must increase this property value. Understand, though, that if a workflow instance encounters a problem that causes it to wait for a response (from an external system, for example) before the first commit point, you could encounter throttle issues because that waiting workflow instance is still considered part of the count of your currently running workflows that the throttle property monitors. This condition could prevent other workflow instances from processing.

Any heavily used workflow environment should do the following:

* Recycle the **OWSTimer** service periodically. A general recommendation is to recycle the timer service approximately every four hours in a **typical SharePoint farm**. If your server farm uses workflow heavily, you might need to increase this frequency. Workflows make heavy use of timer jobs, so recycling the timer service more often can provide increased performance. How often? Well, that depends on the specifics of your server farm, but certainly every three hours, two hours, or one hour is not unreasonable. Timer jobs are not lost when the service is restarted; at worst, they are delayed a few extra seconds while the service restarts.
* Be cognizant of task and workflow history list usage. Remember, these are SharePoint lists and have their own recommendations for scaling—the most important of which is to keep the number of items in a "container" (the folder or root of a list) to less than 2000. When you exceed this number, performance begins to suffer. Typically, this recommendation is related to reading from the lists, but in our testing we saw performance degradation writing to lists that had approximately 5000 items. For workflow, this fact means that you should consider creating new task and history lists for every workflow association. Doing so does not totally eliminate the problem if a workflow writes to the history list often or creates an excessive number of tasks, but it will help. You should also monitor list sizes and consider trimming lists where appropriate. For the task list, some of this trimming is taken care of by the AutoCleanUp job (discussed earlier), but it is not complete and may not be sufficient. See the next portion of this section for our recommendations on getting the best performance out of custom workflows.
* Repeatedly test and tune. Earlier in this article we discussed four settings that affect your workflow environment and its performance. Throughout this article, we have mentioned that you need to take the time to get these settings correct for your environment. Consider this a reminder of that fact. Performance tuning for any environment, not just workflow, is not something you do only one time. You need to determine optimum settings for each configuration option when the system is first brought online and then revisit these settings periodically—especially before new workflows are brought online or significant increases in existing workflows are made.

**Conclusion**

We covered a lot of ground in this article. By now, you should realize that SharePoint workflows can scale to run tens of thousands of processes concurrently when properly designed in a properly configured environment. There are specific guidelines to follow and specific setting that need to be managed to achieve this end, but none of them are difficult to implement.

The tests that we performed involved specific scenarios, but ones that are quite typical for workflow. Although your specific workflow is likely different from the built-in Approval workflow we tested, you can use our results as guidelines for the type of performance you can expect. Similarly, you can use the recommendations presented here to achieve the maximum performance in your specific environment and for your specific workflows.

One final recommendation to take away is that workflows are applications that live inside **Windows SharePoint Services**. You need to think about them in that way if you want to achieve the levels of performance necessary for enterprise environments. Poorly designed workflows or environments can significantly hamper performance and prevent you from achieving your goals. Above all else, test, test, test. It is critically important that workflows be treated like applications, and that includes performance testing in your environment.