

CLAIMS

1. A server system for scheduling jobs of a web application, comprising:
at least one memory; and
a processing system having at least one processor, the processing system being configured to perform operations comprising:
 - spawning a plurality of job schedulers by a single process during a startup of a web application, wherein each of the plurality of job schedulers is implemented as an agent having a thread pool;
 - configuring, in the memory, a job database having a plurality of job entries having different job types;
 - concurrently executing the plurality of job schedulers independently of each other, each job scheduler configured to schedule for execution of jobs in the job database that are of a particular type different from other types of jobs others of the plurality of job schedulers are configured to schedule, wherein the plurality of schedulers include an on-demand scheduler operating to schedule jobs of a first type according to an on-demand basis, a short-interval scheduler operating to schedule jobs of a second type occurring repetitively at intervals of less than a predetermined duration, and a scheduled-job scheduler operating to schedule jobs of a third type at predetermined specific times or at intervals greater than the predetermined duration; and
 - causing performance of jobs scheduled for respective execution by each of the plurality of schedulers.
2. The server system according to claim 1, wherein (1) the on-demand scheduler and the short-interval scheduler are periodically executed based upon respective timers, and cause jobs of the first type and the second type, respectively, to be performed without an additional scheduling step, and (2) jobs of the third type are performed in accordance with a scheduling algorithm.
3. The server system according to claim 1, wherein a job is determined to correspond to one of the different types of jobs according to an urgency or a frequency with which the job is to be executed.
4. The server system according to claim 1, wherein the on-demand job scheduler comprises at least a high priority job manager and a low priority job manager, and wherein the high priority job manager is configured with job processing capabilities greater than the low priority job manager.
5. The server system according to claim 4, wherein the on-demand job scheduler is configured to allocate a first number of threads to the high priority job manager and a second number of threads, less than the first number of threads, to the low priority job scheduler manager.
6. The server system according to claim 5, wherein the high priority job manager and the low priority job manager are each respectively configured to:
 - access a job entry in the job database;
 - add a job corresponding to the retrieved job entry to an in-memory queue; and
 - cause servicing of jobs from the in-memory queue.
7. The server system according to claim 6, wherein the high priority job manager is configured to determine its current processing capacity before the retrieving and adding, and to perform the retrieving and adding only if the processing capacity is determined to be above a specified threshold.
8. The server system according to claim 7, wherein the current processing capacity is determined based upon current occupancy of the in-memory queue associated with the high priority job manager.

9. The server system according to claim 1, wherein the short-interval job scheduler is invoked at intervals of a first duration, and wherein the first duration is determined based upon configured execution intervals of a plurality of job configurations.
10. The server system according to claim 9, wherein the execution intervals are specified in an XML file including the plurality of job configurations.
11. The server system according to claim 1, wherein the scheduled task scheduler is configured to cause performing of a job at a time determined according to a scheduling algorithm.
12. The server system according to claim 1, wherein each job includes an object implementing an execute method, wherein a corresponding of the plurality of schedulers invoke the execute method in order to run the job.
13. The server system according to claim 1, wherein the plurality of schedulers is spawned by a same Windows service.
14. The server system according to claim 13, wherein each of the plurality of schedulers is configured to transmit a heartbeat, and wherein the Windows service determines a health of each of the plurality of schedulers based on the heartbeats.
15. A method for scheduling jobs of a web application comprising:
 spawning, on a processing system having at least one processor, a plurality of job schedulers by a single process during a startup of a web application, wherein each of the plurality of job schedulers is implemented as an agent having a thread pool;
 configuring, in a memory communicably coupled to the processing system a job database having a plurality of job entries having different job types;
 concurrently executing, on the processing system, the plurality of job schedulers independently of each other, each job scheduler configured to schedule for execution of jobs in the database that are of a particular type different from other types of jobs others of the plurality of job schedulers are configured to schedule, wherein the plurality of schedulers include an on-demand scheduler operating to schedule jobs of a first type according to an on-demand basis, a short-interval scheduler operating to schedule jobs of a second type occurring repetitively at intervals of less than a predetermined duration, and a scheduled-job scheduler operating to schedule jobs of a third type at predetermined specific times or at intervals greater than the predetermined duration; and
 causing performance of jobs scheduled for respective execution by each of the plurality of schedulers.
16. A non-transitory computer readable storage medium having stored thereon instructions which, when executed by a processing system having at least one processor, causes the processing system to perform operations for scheduling jobs of a web application comprising:
 spawning, on the processing system, a plurality of job schedulers by a single process during a startup of a web application, wherein each of the plurality of job schedulers is implemented as an agent having a thread pool;
 configuring, in a memory communicably coupled to the processing system, a job database having a plurality of job entries having different job types;
 concurrently executing the plurality of job schedulers independently of each other, each job scheduler configured to schedule for execution of jobs in the job database that are of a particular type different from other types of jobs others of the plurality of job schedulers are configured to schedule,

wherein the plurality of schedulers include an on-demand scheduler operating to schedule jobs of a first type according to an on-demand basis, a short-interval scheduler operating to schedule jobs of a second type occurring repetitively at intervals of less than a predetermined duration, and a scheduled-job scheduler operating to schedule jobs of a third type at predetermined specific times or at intervals greater than the predetermined duration; and

causing performance of jobs scheduled for respective execution by each of the plurality of schedulers.