

CLAIMS

1. A computer system for dispatching a multithreaded guest virtual machine (VM), the computer system comprising:

a configuration comprising a machine that includes a single physical core enabled to operate in a single thread (ST) mode and a multithreading (MT) mode, wherein the single physical core includes multiple physical threads, and

wherein the machine is configured to perform a method comprising:

issuing a start-virtual-execution (start-VE) instruction to dispatch a guest entity comprising multiple logical threads on the single physical core, wherein the guest entity includes all or a part of a guest VM, and wherein the issuing of the start-VE instruction is performed by a host running on one of the multiple physical threads on the single physical core in the ST mode; and

executing the start-VE instruction by the machine, wherein the executing of the start-VE instruction by the machine comprises:

mapping each of the multiple logical threads to a corresponding one of the multiple physical threads;

initializing each of the multiple mapped physical threads with a state of a corresponding logical thread; and

starting execution of the guest entity on the single physical core in the MT mode.

2. The computer system of claim 1, wherein only multiple valid logical threads are mapped to a corresponding physical thread.

3. The computer system of claim 1, wherein the initializing includes:

loading, for each of the multiple logical threads, a portion of the state of the corresponding logical thread into the corresponding physical thread; and

completing, by each mapped physical thread, the loading of the rest of the state of the corresponding logical thread, wherein the rest of the state of the corresponding logical thread is determined based on the portion of the state of the corresponding logical thread.

4. The computer system of claim 3, wherein the portion of the state of the corresponding logical thread includes an address of a state description of the rest of the state of the corresponding logical thread.
5. The computer system of claim 3, wherein one of the multiple physical threads performs the loading, for each of the multiple logical threads, a portion of the state of the corresponding logical thread.
6. The computer system of claim 1, wherein the executing further comprises verifying that the multiple mapped physical threads are not currently running any program code or internal code prior to the initializing.
7. The computer system of claim 1, wherein the starting execution is performed for each of the multiple physical threads based on the initializing completing for the physical thread.
8. The computer system of claim 1, wherein the starting execution is performed for each of the multiple physical threads based on the initializing completing for all the multiple physical threads.
9. The computer system of claim 1, wherein the executing further comprises:
determining that software is running on one of the multiple mapped physical threads; and
based on the determining, completing the start-VE instruction and notifying the host that the start-VE instruction failed.
10. The computer system of claim 1, wherein the executing further comprises:
determining that internal firmware is running on one of the multiple mapped physical threads; and
based on the determining, performing at least one of:
nullifying the start-VE instruction and returning control to the host; and

waiting for the mapped physical thread to become available.

11. A computer implemented method for dispatching a multithreaded guest virtual machine (VM) in a configuration comprising a machine that includes a single physical core enabled to operate in a single thread (ST) mode and a multithreading (MT) mode, wherein the single physical core includes multiple physical threads, and wherein the machine is configured to perform a method comprising:

issuing a start-virtual-execution (start-VE) instruction to dispatch a guest entity comprising multiple logical threads on the single physical core, wherein the guest entity includes all or a part of a guest VM, and wherein the issuing of the start-VE instruction is performed by a host running on one of the multiple physical threads on the single physical core in the ST mode; and

executing the start-VE instruction by the machine, wherein the executing of the start-VE instruction by the machine comprises:

mapping each of the multiple logical threads to a corresponding one of the multiple physical threads;

initializing each of the multiple mapped physical threads with a state of a corresponding logical thread; and

starting execution of the guest entity on the single physical core in the MT mode.

12. The computer implemented method of claim 11, wherein only multiple valid logical threads are mapped to a corresponding physical thread.

13. The computer implemented method of claim 11, wherein the initializing includes:

loading, for each of the multiple logical threads, a portion of the state of the corresponding logical thread into the corresponding physical thread; and

completing, by each mapped physical thread, the loading of the rest of the state of the corresponding logical thread, wherein the rest of the state of the corresponding logical thread is determined based on the portion of the state of the corresponding logical thread.

14. The computer implemented method of claim 11, wherein one of the multiple physical threads performs the loading, for each of the multiple logical threads, a portion of the state of the corresponding logical thread.

15. The computer implemented method of claim 11, wherein the executing further comprises verifying that the multiple mapped physical threads are not currently running any program code or internal code prior to the initializing.

16. The computer implemented method of claim 11, wherein the starting execution is performed for each of the multiple physical threads based on the initializing completing for the physical thread.

17. A computer program product for dispatching a multithreaded guest virtual machine (VM) in a configuration comprising a machine that includes a single physical core enabled to operate in a single thread (ST) mode and a multithreading (MT) mode, wherein the single physical core includes multiple physical threads, and wherein the computer program product comprises:

a computer readable storage medium having program instructions embodied therewith, wherein the computer readable storage medium is not a signal, and wherein the program instructions are readable by a processing circuit to cause the processing circuit to perform a method comprising:

issuing a start-virtual-execution (start-VE) instruction to dispatch a guest entity comprising multiple logical threads on the single physical core, wherein the guest entity includes all or a part of a guest VM, and wherein the issuing of the start-VE instruction is performed by a host running on one of the multiple physical threads on the single physical core in the ST mode; and

executing the start-VE instruction by the machine, wherein the executing of the start-VE instruction by the machine comprises:

mapping each of the multiple logical threads to a corresponding one of the multiple physical threads;

initializing each of the multiple mapped physical threads with a state of a corresponding logical thread; and

starting execution of the guest entity on the single physical core in the MT mode.

18. The computer program product of claim 17, wherein only multiple valid logical threads are mapped to a corresponding physical thread.

19. The computer program product of claim 17, wherein the initializing includes:

loading, for each of the multiple logical threads, a portion of the state of the corresponding logical thread into the corresponding physical thread; and

completing, by each mapped physical thread, the loading of the rest of the state of the corresponding logical thread, wherein the rest of the state of the corresponding logical thread is determined based on the portion of the state of the corresponding logical thread.

20. The computer program product of claim 17, wherein the starting execution is performed for each of the multiple physical threads based on the initializing completing for the physical thread.