Introduction

Management of client systems in an enterprise environment typically involves an IT management console communicating with agents that require the client systems to be powered on. This situation poses a couple of options. The first is to perform management operations while the users of the client systems are present and potentially disrupt their productivity. The second is to perform the management operations when the users are not present which requires the client systems to be left on outside of normal working hours, potentially wasting energy. Management solutions have provided the means to remotely wake systems to address this, but there has not been a solution that allows the local agents to do this autonomously.

There are technologies that allow a system to be scheduled to wake locally from some sleep states, but no standard way to wake locally from off. There are also technologies that allow a system to be powered on remotely through a network connection. The Intel® Active Management Technology (Intel® AMT) PC Alarm Clock feature, available on systems with Intel® vPro™ technology, provides the means to remotely schedule client systems to wake locally from off or any sleep state without a network connection.

This paper describes how Intel® AMT PC Alarm Clock can be used to manage client systems in a more efficient and scalable manner.
Usage Model

The PC Alarm Clock feature introduced in Intel® AMT version 5.1 provides the ability to remotely schedule systems to wake up and perform tasks autonomously without need for a network connection at the time of the wake. The primary usage model is to schedule client systems with Intel® vPro™ technology to power up and perform tasks during downtime hours.

Here are the steps that would occur in this typical usage model:

1. Intel® AMT 5.1 client systems are setup and configured within a network.
2. An IT manager provides task information that it would like to be performed off hours to agents running on the client systems (e.g. security and maintenance actions).
3. The IT manager sets wake schedules for the client systems using the Intel® AMT PC Alarm Clock.
4. Users shut down their client systems when they leave for the day.
5. At scheduled time, Intel® AMT PC Alarm Clock in each system powers up the clients locally and independent of network connection status.
6. The local agents determine that the PC Alarm Clock woke the system and they run their assigned tasks.
7. After completing their tasks, the local agents power the system back down to their previous state.

This usage model demonstrates the efficiency and scalability benefits of using the Intel® AMT PC Alarm Clock to perform tasks off hours.

- Tasks can be performed off hours without interrupting users to improve productivity
- Systems can be powered off when users are not present to save energy/costs
- Task assignment is separated from alarm scheduling to provide more flexible usage for IT managers
- The wake and tasks are performed locally and autonomously, reducing the time and effort required by IT managers
- Network bandwidth requirements are typically small when performing remote power management tasks using Intel® AMT. However, this solution only uses remote calls for scheduling wakes and assigning tasks, which allows for a more scalable solution when running tasks on many systems at once.
Operation

The use of PC Alarm Clock starts with scheduling. When the PC Alarm Clock does not have a set schedule it is in the disabled state. Scheduling is performed remotely by a management console and can be set to wake once or on a recurring basis. After a schedule is set it can be changed or disabled (unset). A scheduled alarm can be missed due to the system already being powered up or ME being asleep. If a single occurrence alarm is missed, the status of the PC Alarm Clock will revert to disabled. If a recurring alarm is missed, the PC Alarm Clock will maintain its status for the next scheduled wake.

When the PC Alarm Clock wakes a system, an entry is added to the Intel® AMT event log that provides two key pieces of information. It indicates that the system was awakened by PC Alarm Clock and it provides the previous sleep state of the system. When the system wakes, the local agent can read the log and look for this event. If the event is present then the agent knows that the wake was due to the PC Alarm Clock and it can proceed with its scheduled tasks. If the event is not present, then the system was booted from another source and the agent may not want to perform the tasks. For example, if the user booted the system just prior to the alarm to get some work done it may not be a good idea to interrupt with a compute intensive management task. The agent is also expected to use this event to return the system to the previous sleep state when finished with its tasks.

The assignment of tasks for the local software agents to perform after an PC Alarm Clock wake is separate from the scheduling of the alarm. This may be done independent of Intel® AMT by direct communication from the management console to each agent while the system is powered up. Alternatively, the 3rd Party Data Store (3PDS) feature of Intel® AMT can be used to pass task information to an agent. The 3PDS provides persistent storage for use with Intel® AMT solutions. At anytime before a scheduled alarm, the management console can use Intel® AMT to connect to the client systems and place task information in the 3PDS even when the systems are shut off. In this case the local agents would be configured to read the task information from the 3PDS after an alarm wakes the system.

When configuring Intel® AMT on client systems that will be using the PC Alarm Clock feature, it is important to select the appropriate power policy for the Manageability Engine (ME). Just like the power saving sleep states of the host system, the ME can be configured to sleep under certain conditions via power packages. Since the PC Alarm Clock feature is only available only when the ME is on it is important to select a power package that will have the ME on when the PC Alarm Clock will be used.

Developing Solutions

The best place to go to get information on developing solutions with the PC Alarm Clock feature is the Intel® Active Management Technology (AMT) Software Development Kit (SDK). Support for PC Alarm Clock is available in the Intel® AMT SDK version 5.1 or later. The latest version is available for download here: http://software.intel.com/en-us/articles/download-the-latest-intel-amt-software-development-kit-sdk/
The Intel® AMT SDK provides documentation on the interfaces used in development of the PC Alarm Clock feature. The PC Alarm Clock can be developed using either the WS-management interface or the External Operations Interface (EOI) a.k.a. Soap interface. The WS-management class references provide the interface information for WS-management development. The AMT_AlarmClockService and the AMT_MessageLog class references provide the important interface information for PC Alarm Clock development. The Network Interface Guide is the comprehensive reference for all EOI. It should be noted that since the EOI is planned to be deprecated in future releases of Intel® AMT it is recommended that any new development be done using the WS-management interface.

The Intel® AMT SDK also provides sample code that demonstrates how to implement an PC Alarm Clock solution. In version 5.1 of the SDK there is a Windows C# sample that utilizes the WS-management interface and a Windows C++ sample that uses the EOI/Soap interface. There is also Event Log Reader sample code available in C++ on Windows and Linux for both interfaces.

The operations used to implement the PC Alarm Clock feature are basic “get” and “set” functions. The current status of the PC Alarm Clock can be retrieved to see if it is disabled or set and if set, when the next alarm is scheduled to occur, and any recurrence. The PC Alarm Clock can be set to perform a single wake or recurring wakes. A current setting can be modified by simply calling another set operation. And finally, a current setting can be disabled. Following is a table that summarizes the interfaces:

<table>
<thead>
<tr>
<th>Operation</th>
<th>WS-management</th>
<th>EOI/Soap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get current status</td>
<td>Get()</td>
<td>GetAlarmWakeUp()</td>
</tr>
<tr>
<td>Set single wake occurrence</td>
<td>Set() with NextAMTAlarmTime</td>
<td>SetAlarmWakeUp() indicate Set with</td>
</tr>
<tr>
<td></td>
<td></td>
<td>StartTime</td>
</tr>
<tr>
<td>Set recurring wake</td>
<td>Set() with NextAMTAlarmTime &amp; AMTAlarmClockInterval</td>
<td>SetAlarmWakeUp() indicate Set with StartTime &amp; Interval</td>
</tr>
<tr>
<td>occurrence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disable current</td>
<td>Set() with Null parameters</td>
<td>SetAlarmWakeUp() indicate Clear with</td>
</tr>
<tr>
<td>wake setting</td>
<td></td>
<td>no other parameters</td>
</tr>
</tbody>
</table>

**Conclusion**

The Intel® AMT PC Alarm Clock feature provides the means for IT managers to run compute intensive tasks off-hours to avoid disrupting user’s work. The use of the PC Alarm Clock also allows the client systems be shut off to save power and costs when not in use, as they will be booted by the alarm when needed. The PC Alarm Clock feature accomplishes this by providing the ability to remotely schedule systems to wake from any sleep state without the need of a network connection at the time of the wake.

Independent Software Vendors should utilize the Intel® AMT SDK to develop management solutions that provide the PC Alarm Clock feature. These solutions can provide cost benefits to Enterprise customers who use the PC Alarm Clock to run management tasks more efficiently.
Author Bio and Photo

Lance received his MSCS & BSEE from the University of New Mexico and has over 20 years of engineering experience starting as a civil servant working on missile systems for the U.S. Navy and spending the last 9 years at Intel. He is currently working as an Application Engineer on the Enterprise Manageability Enabling team helping ISVs adopt Intel® vPro™ technologies.

References


Special thanks to Tom Propst and Michelle Chuaprasert for providing peer review.

Related Links

Relevant Intel Links:
Intel® vPro Technology: http://www.Intel.com/vPro