Application Jet Lag

Consolidating Global Data Services

Jack Di Giacomo
TANDsoft, Inc.
Beaconsfield, Quebec, Canada

Pity the jet-lagged traveler. He departs from Chicago, flies to New York for an early business dinner, takes the red-eye to Paris for two days of presentations, then heads home just in time to tuck his kids into bed. In only four days, he has crisscrossed seven time zones on very few hours of sleep. He never changed his wristwatch settings, so he is always counting the number of hours difference between where he is and where he lives. It's no wonder his internal clock is out-of-sync; he has totally lost track of time.

Now pity that same traveler when he attempts to explain to his boss why the receipt he submitted for a 9 a.m. breakfast meeting with French colleagues was time-stamped 3 a.m. "Honest, it was only breakfast!"

As increasing numbers of businesses consolidate globally distributed services into a single processing environment, the challenge arises as to how to handle system time calls for applications expecting to run under their own local clocks and in their own time zones. Our traveler may have eaten his meal at 9 a.m. in Paris; but the transaction that printed his receipt was operating out of New York on a system whose internal clock was set to New York time, six hours out-of-sync with the French capital.

Similar to people, applications not prepared to cross time zones get jet-lagged, too.

Time Zones – A Brief History

The concept of standardizing time was adopted in the late 19th century through the efforts of Sir Sandford Fleming, chief engineer of the Canadian Pacific Railway and the surveyor of many of Canada's major train routes. Railway travel across great distances in Canada and the United States had rendered obsolete the old practice of local timekeeping as a daily community function based on when the sun reached its zenith (Noon). In essence, every town was its own time zone; and maintaining railroad schedules was very confusing. To simplify arrival/departure timetables, Fleming proposed the worldwide adoption of a standard, or mean, time with hourly variations based on a system of time zones.

In 1884, delegates from 27 nations met in Washington, D.C., for the International Meridian Conference. It was there that the current internationally accepted system of *standard time zones* was established. In the United States, observing the use of time zones became mandatory in 1918.

The present system consists of 24 meridians of longitude (lines running from the North Pole to the South Pole), separated by 15°. Each time zone represents one hour; thus, 24 time zones equal a full day, or a complete rotation of the Earth. The meridians represent the theoretical centers of their respective zones. In practice, however, time zones are sometimes altered in shape for the convenience of local populations.

Jack Di Giacomo has more than 20 years of experience in the design, development, and support of NonStop software solutions. In 1993, he founded TANDsoft Inc., a company focused on delivering quality NonStop solutions for time virtualization, application modernization, and business continuity. Di Giacomo also worked as an HP NonStop instructor and as a software architect for ETI. Contact him at jack.digiacomo@tandsoft.com.

Greenwich Mean Time

Throughout each zone, time is standardized and differs from its neighboring zones by plus or minus one hour. They are measured by reference to a specific point, namely the Prime Meridian of Longitude (0º), that runs through Greenwich, England. Time there is known as Greenwich Mean Time, or GMT(0). A time zone one hour ahead of Greenwich is GMT+1; a time zone one hour earlier than Greenwich is GMT-1. New York City's time zone is GMT-5; Paris is GMT+1; Sydney is GMT+11; Honolulu is GMT-10. GMT works in conjunction with Coordinated Universal Time (UTC), which is essentially the same but is expressed in terms of a 24-hour clock. Local time Midnight is 00:00; Noon is 12:00; 6:30 p.m. is 18:30.

Used in conjunction with Greenwich Mean Time, Noon in Toronto (GMT-5) can be calculated as 12:00 + 5 (time zones from GMT) = 17:00 UTC. Noon in Tokyo (GMT+9) is 12:00 – 9 = 03:00 UTC. It can get very confusing.

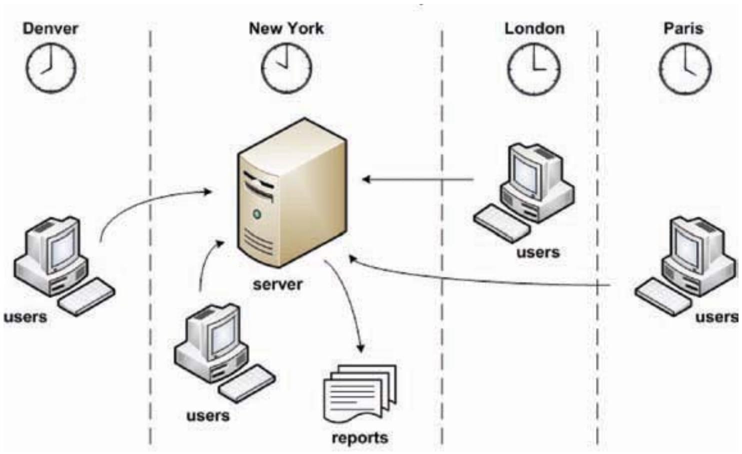
Offset Time Zones

While most places in the world are in time zones that differ in increments of one hour, some places in the world observe what are known as *offset* time zones. They do not observe normal time-zone parameters and instead measure time by an offset of standard time.

China, for instance, spans five time zones but recognizes only one for the entire country (GMT+8). India's offset is GMT+5.5, five and one-half hours ahead of Greenwich Mean Time. An unusual offset is Nepal. The country is only 15 minutes behind neighboring Bangladesh, which recognizes a standard time zone (GMT+6). Although there are 24 standard time zones, the addition of offset times zones brings the total worldwide to 40.

▶

What Time Should the Reports Reflect?



Daylight Saving Time

Daylight Saving Time (DST) is a common system by which time is advanced one hour from standard, typically to extend daylight hours during conventional waking times (usually in the summer). Its use was formally adopted in the United States in 1918. DST does not always occur on the same dates. In 2009, Daylight Saving Time begins in the United States and Canada on March 8, in Europe and the United Kingdom on March 29, in Mexico on April 5, and in Egypt on April 24. For state-specific reasons, neither Hawaii nor Arizona observes DST.

Time-Zone Simulation: A Consolidation Challenge

Consolidating an IT environment onto fewer, more powerful, and highly reliable servers or into a virtual data center has many obvious benefits – significant savings, streamlined architecture, reduced system administration, enhanced service levels, optimized speed and reliability, reduced network complexity—and the list continues.

Not always so apparent are the challenges: the need for increasingly robust server platforms, complex failover requirements, greater bandwidth requirements, more devastating consequences of system failure, and more. Overall, the advantages outweigh the disadvantages. However, companies often become so focused on the benefits that they fail to adequately develop a consolidation strategy that prepares them for issues they must resolve.

One such issue is that of time-sensitive applications. Today's consolidation efforts permit the hosting of multiple applications with different date/time requirements on the same platform. As a result, problems occur with having to provide each application with its own clock and calendar for development, testing, production, disaster recovery, and quality assurance activities. Remember, there are 40 time zones worldwide, not even counting the added complication of Daylight Saving Time.

Take, for instance, an automated teller machine (ATM) network that services customers worldwide. The bank's global operation rests on data centers located within each of four time zones, and every data center manages the ATMs within its own time zone. A major consolidation effort reduces the four centers to just one, situated on the East Coast of the United States. Customers still expect their ATM receipts to reflect their local times, yet the bank must reconcile all ATM transactions with the system clock of its centralized server.

How does an application sitting on a server located within Eastern Standard Time (EST) accurately timestamp local transactions generated outside EST? In a consolidated environment, what must be done to accommodate applications that need to run in “user” time, not “system” time without interrupting normal system operation?

Pick A Solution

Option 1 – Reverse the consolidation effort. Restore servers to each time zone affected.

Concern: Are you crazy? Aside from the massive costs and the loss of all the benefits that consolidation brings, how do you explain your decision to upper management?

Option 2 – Allow applications to be GMT/UTC-dependent. Don't convert to local times.

Concern: Disgruntled users. People think and act in their local times, not UTC. They expect computer-generated bills, statements, reports, receipts, e-mail, and airline reservations to have a local timestamp, not a UTC timestamp.

Option 3 – Create a custom, time-sensitive solution for virtualized time services, provided you have access to the source code.

Concern: A potentially huge and expensive programming effort may be required for modifications. Besides, why risk damaging an application that otherwise is working well?

Option 4 – Use an off-the-shelf product, provided one exists for your IT environment.

Concern: There aren't that many products out there.

▶

However, most that exist install easily, are cost-effective, are user-friendly, and require no application modifications. They often work by establishing virtual clocks that intercept system time calls and return the local times in their specified time zones.

Time-Simulation Products Supporting HP Platforms

Assuming you pick Option 4, below are representative products that provide time-zone simulation for HP Integrity servers.

1. **OPTA2000™** (TANDsoft, Inc.)
   www.tandsoft.com
   A clock and time-zone simulator for HP NonStop servers – S-series, Itanium, and Blades. OPTA2000 eliminates the need to change system clocks in order to test time-sensitive environments and supports multiple time zones on a single server.
2. **Time Machine®** (Solution-Soft)
   www.solution-soft.com
   Provides a unique and versatile solution for user-based time morphing for HP 3000 MPE, Windows, and HP-UX servers. No application modifications are necessary, and there is no need to alter the system clock. Time Machine can simultaneously run up to 20,000 individually defined virtual clocks.
3. **HourGlass™** (Allegro Consultants, Inc.)
   www.allegro.com
   A date/time simulation tool that permits cross-time zone consolidation of applications that aren't time-zone aware. HourGlass is available for HP e3000 (MPE/iX) and HP 9000 and HP Integrity Servers (HP-UX).

Consolidated Data Centers Using Time-Zone Simulation

- A major U.S. East Coast bank uses time-zone simulation to run multiple NonStop NetBatch environments in its central data center. Each NetBatch environment runs in its own simulated time zone and is conscious of the GMT offset and standard time/Daylight Saving Time transition schedules for its time zone. The NetBatch jobs are responsible for preparing market-closing reports that must represent the local time with which the reports are associated. Though NetBatch jobs are run in their respective remote time zones, a system operator can monitor the NetBatch job schedules according to that operator's local clock.
- A major global computer manufacturer has consolidated all of its disaster-recovery systems into one U.S.-based data center. The manufacturer's backup systems are flexible and can be assigned dynamically to take over the processing of one or more failed systems anywhere in the world. By using time zone-simulation, the backup of a failed application can be given an operating environment that has the same time zone as the failed application. In this way, only a few backup systems are needed to protect many systems deployed worldwide.
- A global health care services provider offers Web-based clinical applications to thousands of health professionals worldwide. It is critical that all medical reports and other documentation are properly stamped with the local time. The company originally considered placing individual servers in each of the time zones where customers were located, but the huge costs associated with that endeavor prompted the development of a large server farm centrally built within one time zone. The use of time-zone simulation permits users to benefit from virtual clocks that correspond to their own time zones regardless of the centrally located server's physical clock and time-zone settings.

Summary

Don't pity the "jet-lagged" traveler. He makes sure he's in-sync with whatever time zone he is in, and he's always aware of the hour.

Better yet, his breakfast receipt knows what time it is, too. It's stamped 9 a.m., not 3 a.m.; and, it gets reimbursed without question. The New York-based server now uses time-zone simulation to reflect local time in Paris. The application is no longer jet-lagged.