APF/RTD ON INTEL ARCHITECTURE FOR 300MM

Stephen P Smith & Robert Madson
Intel Corporation
Components Automation Systems

Johnny Wu
Intel Corporation
Logic Technology Development

ABSTRACT

This paper describes Intel's experience using RTD in production on a Pentium III® platform and the effective Cost of Ownership of the IA/NT solution. It describes the real issues with this configuration and our response to them.

In early 2000, Intel began production use of an early alpha release of RTD running on an Intel Architecture (IA) platform under Microsoft NT in its 300mm Technology Development (TD) factory. This paper will describe:

(1) the architecture and configuration of that system

(2) its performance gains & price/performance advantages over a similar Sun/Solaris setup

(3) the availability and reliability numbers we achieved

(4) our experiences with the 300mm usage,

(5) and finally our rational for moving back to Sun/Solaris until Auto Simulations (ASI) finally releases the fully supported production version of RTD on Win2000.

INTRODUCTION

Intel has made a significant commit to utilize the Real-Time-Dispatch (RTD) system in its 200mm and 300mm factory automation suite [1,2]. RTD's ability to cache factory state information in a form that can be efficiently accessed in factory dispatch rules allows manufacturing sites to rapidly implement complex and sophisticated WIP management policies. This is a key requirement for full 300mm automation, where lots will be moved without manual intervention due to ergonomic constraints.

Currently, ASI only provides and supports RTD on Sun platforms running the Solaris operating system. As early as 1998, Intel began a program with ASI to evaluate the impact of moving the production side of RTD (secondary writer, monitor, repository, etc.) to an Intel Architecture (IA) platform running Microsoft Windows NT. Intel received from ASI a "port" of the new LMR version of the software from Solaris to NT. This occurred at the same time as ASI's rewrite of the MMR architecture into LMR and aided them in removing Solaris-only constructs. It was also during this period that Intel was evaluating RTD for rollout into 200mm automation. The port and its comparison data helped Intel to set the proper configuration (memory, number of CPUs, fail over configuration, etc.) for our production facilities. At that time, Intel and ASI came to the mutual conclusion that the time wasn't correct for IA/NT (not enough resources at AutoSimulations could be devoted to a production release and not enough resources at Intel would be available to handle NT issues), though the basic performance data on IA/NT met or beat the Sun/Solaris numbers.

In early 1999, Intel targeted starting up 300mm automation on IA/NT. Late in 1999, we took delivery from ASI of an NT version of V4.2.3. After regression testing (and bug fixes) we put the software in production in our 300mm Technology Development factory (D1c).

Some key questions arise when moving a mission-critical component from one hardware/operating system combination to another. These include:

- Does the core functionality work the same?
- What is the performance of the software application under the new hardware and operating system?
- Does the new OS have the stability to run mission critical 7x24 operations?
- Does the software supplier have the correct knowledge of the new hardware and software environment to accomplish the port and provide support?

In the rest of the paper, we will address each of these questions.

RTD ON WINDOWS NT©

While having the full RTD implemented on NT was desirable, we agreed with AutoSimulations that it was key to have the server-based components available first (so additional cheaper hardware could be added to meet volume/performance needs). Since moving GUIs from X-Windows to MS-Windows is a time consuming chore, our insertion consisted of only the server components of RTD (2ndary writer, Monitor, Repository Server, and Repository, etc.).

CONFIGURATION OF THE SYSTEM

For our current Solaris RTD implementations, Intel worked with ASI to define the recommended architecture of two independent platforms, one to support dispatching/reporting, and the other as a warm spare in case of failure with the first box. The IA/NT implementation followed the same architecture layout. The MES or primary system remained the same, a single primary writer, feeding two secondary platforms.

The secondary or IA/NT system consisted of dual Pentium III® 550Mhz with 1GB RAM running Microsoft Windows NT 4.0 Service Pack 4. Each system had a secondary writer, monitor and two dispatchers running as the RTD processes. Each secondary process was installed as a WinNT© service. Having the processes setup as services had both upsides and pitfalls. On the upside, in the event of a system failure the processes were configured to automatically restart. Also by having the processes as services allowed system users to log off, and have the processes continue running. However, if multiple similar processes were configured to run, only one service was responsible for each. This caused had a serious pitfall; such as if one dispatcher were to crash, the other would have to be stopped as to restart both processes through the service.

Testing from 200mm implementations on Solaris showed that the platform could only support 2 dispatchers, with the interception and dispatch volumes at the time. However testing on the IA/NT system revealed that up to 4 dispatchers could be supported without significant impact to dispatch or interception performance.

REGRESSION TESTING & ISSUES

Before the IA/NT version of RTD was implemented in production, a series of regression and volume tests were run to insure a hardened product was being installed. The issues that were found were categorized into 2 categories, critical and CIP (Continuous Improvement.)

Initial critical issues for NT included lack of “handler” support in the NT Monitor (for Intel’s use of handlers, see [1]). ASI quickly fixed these.

As for CIP issues, many were found. Most arose in the initial regression/volume testing, and were deemed not crucial to the livelihood of the product. One of these which hurt us later was a bug which caused log files not consistently being written, As with any application, any errors or clues to problems with the system are logged in some file. The lack of log files made debugging very difficult and yielded use of external application logs to work through issues.

Another CIP issue (one part-and-parcel to the overall porting strategy) was the lack of a rule editor, formatter and reporter. Without these processes, the RTD system would become essentially useless. Intel defined a manual procedure to use the rule editor on a small Solaris workstation. Dispatch policies were written on the Solaris system, and then copied over to the IA/NT system. This process was cumbersome, but enabled Intel to gain the use of the IA/NT system and reap the cost benefits of using IA/NT architecture over Solaris.

However the formatter and reporter were not ported to NT. This caused some pain, as many dispatch rules relied on input files that were generated through reports and then input to the rules as flat files. To overcome this issue, we incorporated this report information into the rule, which would be run at each rule execution. Overall some time was added to the overall dispatch request, but with the system speed, and small size of the TD factory, the impact was negligible

PRICE/PERFORMANCE

Cost of Ownership is always an important factor when choosing any system whether SW or HW. For RTD both components are extremely important to the cost of the overall system. RTD is an extremely I/O intensive system. Intercepted data it received from the MES, and written to disk several times, while simultaneously dispatch requests are being serviced.

For all of this to occur without issue a stable and high performance system is required. Solaris systems are renown through out the industry as high power and stable systems. However the cost of Solaris systems is extremely high, making the cost of ownership of the system lofty.

In order to increase the utilization or reduce the CoO of the system, a low cost, stable, and high performance system was needed. A series of volume tests were generated whereas to tax the overall system on this new IA/NT architecture. As was described earlier, RTD is extremely I/O intensive, so the limits were pushed to determine what the system was capable of.

The bottlenecks of the current system are the monitor and dispatcher. Volume tests were setup to test both of these components. Using the Virtual Primary Writer, we simulated the number of intercepted transactions being input into the system. Also using an Intel developed script we varied the number of dispatch requests over the test period.

Test results showed that the monitor on IA/NT was capable of processing up to 100tags/sec, while the dispatchers were able to handle up to 2 dispatches/second. This is a giant leap forward from the current Plan of Record HW. Using a Sun3500 with 2 336Mhz Sun CPU's and 1GB RAM yields results with the monitor around 60 tags/sec and the dispatcher at right around 1 dispatch every 2 seconds. Each test was conducted with 2 dispatchers running per node.

One of the biggest improvements was seen with the performance of the batch lookup dispatcher cache. In several cases initial dispatch policy executions would take up to 10seconds for the dispatcher to cache all necessary information. However with the IA/NT system this number dropped to about 2 seconds for the initial run, or for an 80% improvement.

RELIABILITY/UPTIME

Uptime is a critical metric when sustaining a mission critical system. Intel defines uptime as the amount of time that the system is running by the total amount of time available. This metric does not include issues that do not cause the overall system to fail. However if an issue impacts the system for more than 3hrs then it will be considered a downtime, and will affect the uptime indicator. An example of this would be if the PW were to fail for some period, dispatching can continue without issue.

Indicators show that over the last 12 months uptimes are above 99.98% for Intel's 200mm RTD implementations.

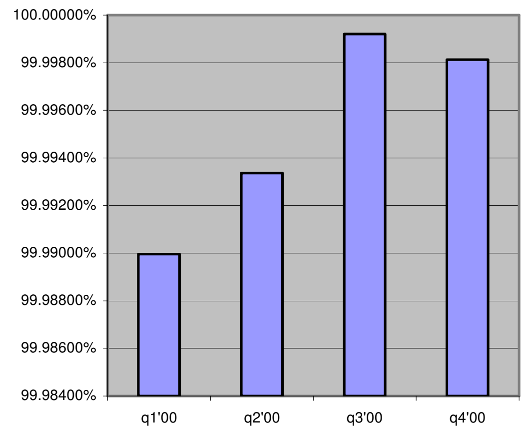

Figure 1: APF Availability/Quarter

Sun/Solaris is renowned for building mission critical HW/SW systems, Microsoft NT has also been defined as capable of supporting mission critical software. Moving to RTD on IA/NT, concern was not so much with the HW as with the OS. Microsoft NT is much more stable than any non-mission critical OS.

In moving RTD to IA/NT systems, the largest concern was with the maintainability of the Microsoft product suite. RTD sustained production manufacturing for over 1year without any downtime. Dispatch rates were at 2 dispatches per minute, while the interception rate was at 1 per second. For the 1.3 yrs that IA/NT system sustained production, uptime was 100%, bettering our 200mm Sun/Solaris availability. The caveat to this result is if there is any major volume impact on uptime, since D1c had about one-third the volume has our HVM factories.

Once the IA/NT system was removed from supporting production, a high power Sun/Solaris system was implemented. This system had four 366Mhz CPU's and 2GB RAM. Dispatch times on the IA/NT system were about 20% less than existed on the Sun/Solaris implementation. As seen in Figure 2, dispatch times increased by 20% with the switch back to Sun/Solaris with the same policies as that which were run on the NT system.

Over this period, some issues were seen and will be discussed in following section.

FACTORY ISSUES

Low number of issues and quick turnaround time to solve them is critical to the implementation and execution of mission critical products.

From the numbers discussed in the sections above, the IA/NT version of RTD outperformed the Solaris version for dispatch performance. However in 300mm fabrication facilities FOUP tracking is extremely important to continuous production and WIP movement. FOUP's have a specific state model. After processing, each FOUP must be cleaned and then recycled for the next available lot. RTD was setup to track the empty FOUP's and to provide the dispatch lists showing the available FOUP's. This dispatch policy was executed on average 300 times per day and the average return size contained 300 entries. As the number of FOUP's increased in the factory and the number of times the policy was run, the runtime of this algorithm increased. At the peak it, runtime exceeded 20sec. Once ASI investigated this issue they found that this was caused by a default buffer size set by Microsoft Windows NT©. Once the root cause of this issue was found, and a patch installed runtimes decreased to the average runtime of 4sec per request. However, this issue took over 8 weeks for ASI to investigate, but only 4 days to solve and release a patch.

The other critical issue was found to be with missing lots when using the DispatchLots folder. On several occasions, an operation based dispatch using the Default rule (and thus the DispatchLots folder) would result in missing lots. These lots were seen using an equivalent rule that used the WIPLOT and a filter. As a result, several rules specific to the operation number was created to fix the situation. This was another case where ASI bandwidth was an issue. The issue took 7 weeks to investigate and 3 days to solve and deliver a patch.

All in all the overall number of critical issues logged against the implementation was minimal, as compared to the CIP items. However when a critical issue arose, it was very difficult to get the necessary resources to investigate and fix the issue.

FINAL DECISION

Late in 2000, Intel made the decision to move back to a Sun/Solaris platform in our D1c factory and stick thru this decision during initial ramp up of our 300mm High Volume factory.

Key reasons for the decisions were:

- As described above, ASI was not able to guarantee a quick turnaround time on NT-only issues until their full production release was available,

- ASI's announced slip to their roadmap in the delivery of the full NT/Win2k supported release past the Q2'2001 freeze date for the 300mm automation baseline at Intel.

Intel recently inserted V4.3 into D1c production on dual Sun Solaris boxes. The cost of the boxes was 10X the cost of the boxes which had been running D1c for 10 months. In additional, a clear increase in dispatch rate was seen by the factory after this change.

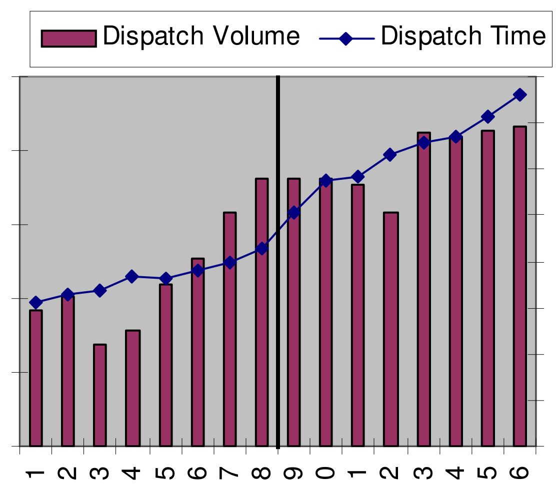

Figure 2: Dispatch Rate Summary

While it was disappointing to have to remove our NT boxes from D1c, we gained excellent experience with production use of the product without the risk of impacting a revenue-producing factory. All in all, Intel spent about 1 man-year of effort to test, validate, and coordinate the work RTD on IA/NT.

RTD is a key component of Intel's factory automation architecture. To keep price/ performance within reasonable bounds (not withstanding strategic business drivers!) Intel desires to have RTD running on IA under Windows. Our experience with RTD in production on IA/NT shows that this is going to be a excellent product. ASI's current roadmap for RTD on NT is APF v5.0, expected late in '01. We assume that by the APF Version 5.0 production release, AutoSimulations will have the necessary bandwidth to be able to support this mission-critical product.

REFERENCES

1. Johnny Wu and Robert Madson, APF Dispatch Rule Performance Improvements Using Custom Monitor Handlers, in *Proceedings of the AutoSimulations Symposium*, 2000.

2. Johnny Wu and Robert Madson, Using RTD in 300mm, this Symposium.

BIOGRAPHY

ROBERT MADSON, Software Engineer, joined Intel in 1997. Robert provides primary, technical and project expertise for APF proliferations at Intel. His current responsibilities also include central support and ownership of APF at all of Intel's fabrication facilities and development of Intel's next generation scheduling systems. He previously implemented AutoPlan based planning tools for Intel's Assembly/Test sites. Robert holds a B.S. degree in Industrial Engineering from University of Arizona.

JOHNNY WU, Senior Automation Engineer and has been working at Intel on Factory Scheduling since 1997. Currently, Johnny is primarily responsible for 300mm Execution Controls and scheduling systems development, as well as working on next generation factory control systems. Johnny holds a M.S. degree in Control Systems Engineering from the Dept. of Mechanical Engineering at the University of California, Berkeley.

STEPHEN P. SMITH, PHD, was the Group Leader for Factory Scheduling in Intel's corporate factory automation group. He is now working with strategic ISVs to better their performance on IA32 and IA-64 platforms. He holds a Ph.D., Masters and B.S. in Computer Science, all from Michigan State University. Prior to joining Intel in 1991, Dr. Smith was a member of the research technical staff at a remarkable little place called the Northrop Research and Technology Center (NRTC) in Palos Verdes, California, where he advanced esoteric fields such as machine vision and learning while programming Lisp Machines.