Preface

Lin Chao
Editor
Intel Technology Journal

Technology is often a key business differentiator. This is especially true for e-Businesses. Simply put, e-Business is defined as Internet-based commerce. Both start-ups and established companies alike face the challenge of doing business on the Internet, at Internet speeds.

Intel is one such company adapting to doing business at Internet speed. We are striving to make our own internal processes "e-Business-like." For example, Intel's Web-based order management system allows our customers to place orders, track deliveries, post inquiries, and obtain product and pricing updates. We also have an Internet hosting business called Intel® Online Services, Inc., whose mission is to provide second-generation web hosting to companies worldwide.

Internet data centers are the physical environments where all the e-Business magic occurs. The five papers in this issue of the *Intel Technology Journal* focus on the planning, implementation, and deployment of our e-Business technology data centers, including those data centers managed by Intel's information technology group and Intel Online Services, Inc. The first paper gives an overview of Intel's own internal e-Business data centers. It outlines the current and future direction of the technologies that are being applied to fulfill Intel's future e-Business growth. The second paper describes IP addressing issues for Internet data centers; in particular, it raises the concern about the dwindling amount of IP address space. The authors describe how IP address space concerns impact the design, implementation, and operation of Internet data centers.

The third paper discusses the process used to certify the reliability of systems and service offerings in each new data center that comes online for Intel Online Services, Inc. To meet the rapid growth in the number of e-Business applications that are being deployed and the rate at which they are being released, the fourth paper looks at release management and application landing. And, finally, the fifth paper looks at asset management and capacity planning. Asset management can mean anything from determining where a system is located to what applications are running on that system. Effective asset management and capacity planning are vital to the success of any Internet data center.

Fueling the Engines of the Internet Business Machines

By Mike Aymar
Corporate Vice President; and President, Intel Online Services, Inc.
Intel Corp.

By Doug Busch
Vice President, Information Technology, and Director, Information Technology
Intel Corp.

Market research firm Gartner Group estimates that Business-to-Business e-Commerce will be a $7.29 trillion market segment by 2004. To remain competitive, companies must move online. Responding to an internal and external focus on e-Commerce, Intel is addressing this competitiveness head-on by becoming the building block supplier to the worldwide Internet economy.

Since 1993, Intel has been building strong global networking and Information Technology (IT) expertise. Intel's internal IT organization, with nearly 4000 employees, manages a very large worldwide infrastructure that supports Intel facilities in 93 locations across 32 countries. A sense of the scale of Intel's IT infrastructure is provided by a few statistics: there are over 4300 infrastructure and app servers; 2300 network devices (routers, WAN links, LAN connections, hubs, and switches); 325 Mbps worldwide network capacity on more than 300 circuits; 3.5 million mail messages per day; and 24X7 e-Business and enterprise applications operations with better than 99.97% uptime. This worldwide, highly reliable environment is a crucible for the development of IT expertise and techniques. This infrastructure allows Intel to do the majority of our business — almost $24 billion of $31.8 billion in sales — over the Net.

Building on Intel's own internal experience and expertise, Intel® Online Services was formed in early 1999 to deliver global e-Commerce and application hosting for other companies. By having Intel deploy and manage their e-Business applications and infrastructures, these companies can remain focused on their core competencies.

Intel Online Services' core infrastructure is a growing network of data centers that provides application-hosting services to worldwide customers. Currently, seven data centers are located in the US, Europe, Australia, India, Japan, and Korea with more planned for 2001. First- and second-generation hosting are offered. "First-generation" Internet hosting is generally defined as "co-location" meaning that the customer brings everything but the basic facilities and connectivity. "Second-generation" hosting is also available. Features include multiple open-standard applications that enable the fast development of complete e-Business solutions. These capabilities translate into secure, ultra-reliable Internet applications, as well as world-class network and server infrastructures.

Quokka, CommerceRoute, Inc. and a variety of other companies are already taking advantage of the e-Business solutions offered by Intel Online Services. Intel Online Services has also forged relationships with top solutions developers and network providers, including Appgenesys, Proxicom, and PricewaterhouseCoopers

Over the next few years, Intel will continue to grow our internal and external e-Commerce support. Internal e-Business systems will be an important vehicle for developing Intel expertise and leading the industry in the development of e-Business capability. Intel Online Services will continue to open data centers globally and provide more end-to-end application services in support of Intel's external customers. Intel will continue to focus on being the preeminent building block supplier to the worldwide Internet economy.

IP Addressing Space Design Issues for Internet Data Centers

Lynne Marchi, Intel Online Services, Intel Corp.
Sridhar Mahankali, Intel Online Services, Intel Corp.
Jeff Sedayao, Intel Online Services, Intel Corp.

Index words: Internet protocol, address space, data center

ABSTRACT

The implementation of Public Internet Protocol (IP) address space is a key factor in the size and growth of Internet data centers. IP addressing space decisions affect how many servers can be hosted at a data center, and they influence the kind of network connectivity technology that will be used and even how web sites are implemented. This paper describes IP addressing issues for Internet data centers. First, we provide an overview of Internet addressing and routing: we discuss IP networks, autonomous systems, and high-level Internet network routing. Key Internet constraints are described, particularly the finite amount of IP address space and autonomous systems and the current addressing and routing policies that result from those constraints. We then go over key IP address design decisions. The Internet data center builder needs to decide what address space to use, the size of that address space, the autonomous system number to use, and the address allocation policy to use with customers. These choices are constrained by the difficulty of obtaining space, the required speed of implementation, Internet Service Provider (ISP) routing policies, ISP connectivity decisions, and security requirements. Next, we describe how these design choices affect technology choice and implementation with the data center, by using virtual web site design and Network Address Translation (NAT) as examples. We then provide examples of how address space constraints affect the design of Intel® Online Services (IOS) data center address spaces and other technology choices. The last section discusses some trends and future technologies that may alleviate IP address constraints.

INTRODUCTION

Issues with Internet Protocol (IP) address space are critical, yet often overlooked, factors in building and maintaining Internet-accessible data centers and web server farms, such as those hosted by companies like Intel Online Services. A shortage of address space can limit the growth and expansion of data centers. Moreover, dealing with the scarcity of IP address space and with situations where the data center and customers have to communicate while sharing the same private address space drives technology decisions and complicates the debugging of server and network problems. This paper describes how IP address space concerns can impact the design, implementation, and operation of Internet data centers. First, we present an overview of how IP addressing and routing works on the Internet. We then discuss key address space design considerations. The next section describes the effects of addressing choices on data center design and implementation, and it is followed by a section in which we show examples of address space design decisions at Intel Online Services. We end with a discussion of future technology development trends regarding IP address space.

A BRIEF INTRODUCTION TO INTERNET ADDRESSING AND ROUTING

Internet protocol addressing and routing must first be understood before any discussion of IP address space design issues will be useful. This section goes over the original IP address scheme, its limitations, and the current methods used to deal with the finite number of IP addresses. This information is crucial to an understanding of the choices and constraints for IP addresses in a data center.

Class	Range of Network Numbers	Default Network Mask	Network vs. Host Portion	Number of Hosts
A	1.0.0.0 to 126.0.0.0	255.0.0.0	network.host.host.host	16,777,214
B	128.1.0.0 to 191.254.0.0	255.255.0.0	network.network.host.host	65,534
C	192.0.1.0 to 223.255.254.0	255.255.255.0	network.network.network.host	254

Table 1: Original IP version 4 address class

Table 1 shows the different classes of IP addresses. Note that two other classes of address space, class D and E, were not included in the above table. Class D addresses start at 224.0.0.0 and are used for multicast. Class E addresses start at 240.0.0.0 and are used for experimental purposes.

Original IP 4 Addressing Scheme

In order for two hosts to communicate over the Internet, there needs to be a way to uniquely identify hosts. In 1981, the Internet Engineering Task Force (IETF) created the Internet Protocol, IP version 4 (IPv4) [1], which defines the current method of uniquely identifying hosts. IPv4 addressing uses a 32-bit binary address. The IETF also incorporated support for decimal representation of addresses to make the addresses human-readable. In decimal form, an IP address consists of 4 octets (sets of 8 bits), separated by dots. Each octet can be a number ranging from 0 to 255. Examples of valid (decimal) IP addresses are 10.245.171.1 and 172.16.50.224.

IP addresses are partitioned into a network portion followed by a host portion. Hosts belong to a network, and that network is defined by the network portion of the IP address. The original design called for classes of address space that divided the IP space into large, medium, and small networks that could be assigned to organizations (businesses, universities, government agencies, etc). Included in the design was the notion of a network mask that defines what part of an IP address is the network portion (as opposed to the host portion of the address). In binary, the network portion of the address is a series of ones that is then followed by a series of zeroes representing the host portion of the address. In decimal, the network portion of the mask is equal to 255 for each octet.

Autonomous Systems

Another requirement for Internet communication is that each host needs to know how to reach all of the other hosts. To facilitate this, organizations advertise the path to their network to other networks. Devices called routers learn about networks in this way and forward packets appropriately. Routers exchange network and routing information through what is called a routing protocol. A *routing table*, which is a list of networks and the next hop (often another router) to forward packets for those networks, is stored in the router's memory. The router will select the best path (next-hop router) to put into its table when multiple paths exist to the same network.

In the early days of the Internet, all connected routers shared their routing tables. As use of the Internet started to grow, more routers and networks were added, and the amount of overhead required to store the routing table and manage changes to the routing table also increased. In addition, as more companies began manufacturing different routers that ran their own implementation of the routing software, compatibility issues between different vendors arose. For these and a number of other reasons, it was decided to break the Internet into smaller routing domains, called Autonomous Systems (AS).

An autonomous system (AS) [2] is a set of routers and networks that are managed by one or more administrative entities (e.g., company, university, Internet Service Provider, etc.). Each AS is assigned a unique number so that communication between different autonomous systems can occur. Routers inside the AS run an Interior Gateway Protocol (IGP) such as RIP [3] and OSPF [4]. To communicate externally, one or more border routers are chosen. Border routers use an Exterior Gateway Protocol (EGP) to exchange routing information with routers in different autonomous systems. Today, the *Border Gateway Protocol Version 4* [5] (BGP4) is generally used for this purpose.

Each AS has a number associated with it. BGP4 uses 16 bits for AS numbers, so that AS numbers range from 0 to 64535. The upper 1024 are reserved as private AS numbers, usable only within an AS and not directly reachable from the Internet. This leaves AS numbers 1 to 64511 as valid, Internet-usable AS numbers.

Issues With IP Addressing

Since IPv4 was finalized, use of the Internet has grown exponentially, causing major addressing issues. In the early days of the Internet, organizations were able to obtain large blocks of IP space without proof that it was needed or even going to be used, and as a result, IP address space was being rapidly depleted. Another side effect of address space allocation policies was that the routing tables for Internet routers were once again becoming huge [6]. Remember, routers store a list of networks and next hop information in memory. When routing tables are large, they take up more memory and more CPU processing time is required to search them.

Finally, the class of address space as defined in Table 1 did not always meet, and sometimes exceeded, the needs of the organization receiving it. For example, a small business that expected to grow to no larger than 300 hosts would require two Class C networks (508 addresses). This wasted 208 addresses (two 256 host networks minus four addresses that are network overhead and minus the 300 hosts)!

Address Allocation Authority

To slow the depletion of IP space, the Internet Assigned Numbers Authority (IANA) [7] was established to oversee allocation of the remaining IPv4 addresses. IANA further delegated this authority to the regional registries:

- American Registry for Internet Numbers (ARIN)
- Asia-Pacific Network Information Center (APNIC)
- Réseau IP Européens (RIPE NCC)

Today, it is much harder to obtain IP address space as the requesting body must provide a detailed plan that shows that the requested space is justified and how it will be used.

Subnetting Changes

Several new methods of addressing were also created so that usage of IP space was more efficient. The first of these methods is called *Variable-Length Subnet Masking* (VLSM) [8]. Subnetting had long been a way to better utilize address space [9]. Subnets divide a single network into smaller pieces. This is done by taking bits from the host portion of the address to use in the creation of a "sub" network. For example, take the class B network 147.208.0.0. The default network mask is 255.255.0.0, and the last two octets contain the host portion of the address. To use this address space more efficiently, we could take all eight bits of the third octet for the subnet.

One drawback of subnetting is that once the subnet mask has been chosen, the number of hosts on each subnet is fixed. This makes it hard for network administrators to assign IP space based on the actual number of hosts needed. For example, assume that a company has been assigned 147.208.0.0 and has decided to subnet this by using eight bits from the host portion of the address. Assume that the address allocation policy is to assign one subnet per department in an organization. This means that 254 addresses are assigned to each department. Now, if one department only has 20 servers, then 234 addresses are wasted.

Using variable-length subnet masks (VLSM) improves on subnet masking. VLSM is similar to traditional fixed-length subnet masking in that it also allows a network to be subdivided into smaller pieces. The major difference between the two is that VLSM allows different subnets to have subnet masks of different lengths. For the example above, a department with 20 servers can be allocated a subnet mask of 27 bits. This allows the subnet to have up to 30 usable hosts on it.

Class	Private Address Space
A	10.0.0.0 to 10.255.255.255
B	172.16.0.0 to 172.31.255.255
C	192.168.0.0 to 192.168.255.255

Table 2: Private address space ranges

Private IP Space

In 1996, IANA set aside three blocks of the global IP space to be used by organizations solely for the purpose of internal communication [10]. This address space, called private IP space, meant that a company could assign private addresses to hosts inside the company that did not require direct access to the Internet. Any organization could use private space without fear of colliding with another organization's address space. This allowed companies to conserve on the public IP space they had already acquired by assigning it to only those hosts that needed to communicate directly with the Internet. Table 2 shows which networks can be used for private addressing.

Classless Internet Domain Routing

So far, the discussion on IP address allocation has used the model shown in Table 1. This model is often referred to as a "classful" model because it relies on using the definitions of class A, B, and C networks. *Classless Inter-Domain Routing* (CIDR) [11, 12] eliminates classful addressing in the same way that VLSM eliminated fixed-length subnet masks. CIDR uses a prefix to indicate the number of bits used for the network portion of the address, while the remaining bits are used for the host address. For example, 147.208.61.8/20 is a CIDR address in which the first 20 bits contain the network portion of the address, leaving 12 bits for the host portion. The network mask for a /20 prefix is 255.255.240.0 and is equivalent to 16 traditional class C networks!

Another advantage of CIDR is it allows routes to be aggregated. This means many networks can be summarized into a single route. For example, 147.208.0.0/19, 147.208.32.0/19, 147.208.64.0/19, and 147.208.192.0/19 can be summarized as 147.208.0.0/17. Once CIDR was implemented, the growth in the size of Internet routing tables was significantly reduced.

ADDRESS SPACE DESIGN ISSUES

When implementing an Internet data center, there are a number of key decisions that need to be made about IP address space. In this section, we discuss four key design points: what address space to use, the size of address space to advertise for a data center, what autonomous system number to use, and the IP address allocation policy. For each of these design issues, we talk about the different choices available and the tradeoffs involved with each choice.

What IP Address Space to Use

The first critical choice that data center implementers have to make is what IP address spaces to use. There are several choices here:

1. Private IP address space
2. Currently owned and used space
3. Space from the data center's ISPs
4. New space obtained directly from Internet registries
5. Customer space

These choices are not mutually exclusive within a data center, and we will go over the tradeoffs involved with each choice.

Private IP address space has the primary advantage of being plentiful and immediately available. It has the primary disadvantage of not being immediately usable on the Internet without some form of Network Address Translation (to be discussed later) or some kind of proxying technique. This disadvantage is at times not a problem. For applications and services that do not require direct access to the Internet, this is not a concern. Also, for hosts such as database or application servers that do not directly talk to other systems on the Internet, this has some security advantages, as these systems are not vulnerable to direct attack from the Internet. (Do not think that they are invulnerable because of this, however).

If data center implementers already have their own IP address space, another possibility is to use that space. This can be advantageous, as an organization may have plenty of address space to utilize immediately. The prime disadvantages can be with routing. For example, some ISPs do not accept network prefixes longer than a /16 for parts of traditional class B networks. So if you wanted to use part of a /19 part of a traditional class B for a data center, that data center would not be accessible from all ISPs.

Another option is to use space from an ISP. ISPs have address space that they will provide to customers. While this option has the advantage that there is address space to use immediately, this option has a number of powerful disadvantages. The first disadvantage is that using an ISP's address space will typically limit you to using one ISP. That address space is bound to that ISP, and you typically will not be able to have traffic routed through another ISP. Another disadvantage is that should you choose to discontinue your service with that ISP, you would have to give back all of the space you received, forcing you to renumber all of the hosts directly accessible from the Internet.

Another option is to obtain new address space directly from the Internet registries that distribute space. This option has a number of advantages. A data center using space obtained from the Internet registries can change ISPs without having to worry about renumbering hosts. The registries usually allocate addresses in /19 blocks, making that address space immediately routable. The disadvantage of getting space from registries is that the process takes time, on the order of weeks, and longer if you need to first join the registry. The process also involves rigorous justification of address allocation and why currently owned addresses will not suffice. In addition, once space is allocated, it cannot be used all at once. To use more of an address allocation, another justification process is required, often requiring verification that previously assigned addresses have been used.

A final choice is using the data center's customers' address space. When possible, this is good, but it is often not possible, as Internet data center customers usually expect that you will use your own address space. Even if a customer is willing to do this, it may take some time to make changes to the data center's ISP's address filtering to make it possible to use that space. Also, customer address space is also vulnerable to the same problems as address space you already own. The customer may use a piece of existing address space, such as part of a class B, that some ISPs may refuse to accept as a route.

Size of Address Space to Advertise from the Data Center

A decision closely related to what address space to use is what size of address space to use and how to advertise it on the Internet. Clearly, you can only advertise the space that you have: that puts an upper limit on the address space advertised for the data center, and thus a lower limit on the prefix length of the network advertised. Many ISPs will not accept route advertisements for networks with prefix lengths longer than /19, which puts an effective upper limit on the prefix length of what you advertise and a lower limit on the size of the address space.

There are a number of factors that affect the size of the address space advertised. First, it depends on how many hosts within the data center need public addresses. You want to advertise enough address space to cover the hosts that need public addresses, both immediately and in the near future. To make routing more manageable and to help reduce the growth in the size of the Internet routing table, it is better to advertise fewer routes. Instead of advertising each network in a data center, if you advertise a single aggregation of those routes, there are fewer routes to manage. As mentioned above, some ISPs will not accept parts of a traditional class B network. One way to deal with such ISPs is to connect data centers to ISPs who will accept parts of a class B. Each data center can advertise a prefix that is short enough to be accepted, while at least one data center can advertise the whole /16 class B network. This way, those ISPs who only accept a whole class B will see a route for the whole class B and send it to the data center's ISP. Once it is in the data center's ISP, the ISP will route traffic to the most appropriate data center because each data center is also advertising a route for its section of the class B.

Another factor in deciding what size address space to advertise is the backbone infrastructure connecting data centers. If data centers are connected by a backbone network that has enough capacity to route significant amounts of public Internet traffic coming into one data center that is bound for another (the worst case being that the backbone will handle all of a data center's traffic), then it is feasible to advertise a single route that aggregates all of the data center's networks into one. If the backbone connecting the networks doesn't have the capacity, advertising a single aggregated route can result in performance bottlenecks when end users accessing one data center access that data center through another.

Autonomous Systems Number

The issues and choices regarding Autonomous Systems (AS) numbers are very similar to those regarding IP address space. A data center's address space can be advertised from the following:

- Private AS
- Currently owned and used AS
- The data center's ISP's AS
- A new AS obtained directly from Internet registries
- Customer AS

Advertising a route from a private AS number is fast and immediately available, but those private AS numbers are only usable within an organization's public AS. Like private IP addresses, private AS numbers are not usable over the Internet. Advertising from an existing AS number that is already owned by a data center's administrators is also easy and quick. One consideration to keep in mind when using an existing AS is the routing policy implemented by ISPs and other organizations. Routing policies are often implemented by AS numbers, and each of the data center networks advertised from that AS will be affected by such a policy. This can become a great disadvantage when data centers are spread across geographies. Internet conditions can vary greatly: the routing policy made on one continent may be (and usually is) totally inappropriate on another.

If a data center's routes are advertised from an ISP's main AS number, the data center is locked into using only that ISP and cannot have connections into other ISPs (although it should be mentioned that some ISPs provide a special AS for multihomed customers). Getting a completely new AS number from an Internet registry has the advantage that a data center can change ISPs with much less work, usually just changing entries in routing registries. Multihoming to multiple ISPs is now possible, and end-user access to the data center can be improved by routing traffic based on using the full Internet routing table. The down side to getting a new AS is that it takes time: you usually have to join an Internet registry and apply for an AS number. Also, AS numbers are limited in quantity, as mentioned above, with only 64511 AS numbers available for use directly on the Internet.

Finally, a data center can advertise address space/route from a customer's AS number. This has the advantage of allowing that address space to be served by multiple ISPs. It has a number of constraints and disadvantages. Only the address space/routes that the customer owns can be advertised as using that AS. As with using a previously owned AS, this option has the disadvantage of being affected by any other policy that organizations and ISPs may implement based on the customer's AS numbers. The data center effectively becomes an ISP, and the data center's ISPs often must change AS and network filtering policies to allow that route to be advertised. In addition, routing registry information concerning that network will also have to be changed.

IP Address Allocation Policy

Given that IP addresses are limited in quantity and their use has serious constraints, the allocation policy for IP addresses to data center customers is a serious concern. There are a number of constraints and tradeoffs. The first choice that needs to be made is whether to allocate a separate address space to each customer. From a customer, security, administrative standpoint, it is better to give each customer separate address spaces. Firewall policies are easier to implement on a subnet basis, and any special traffic policies, such as giving certain customers a different path or giving them priority over others, are much easier to implement if customers are on different subnets. Customers may be competitors, and the thought of a competitor on the same subnet may be unpalatable to a data center customer. The cost of separate subnets per customer is loss of usable address space. Each subnet has a subnet number, and typically that is not used as a host name to avoid confusion. Also, each subnet has a broadcast address that cannot be used for hosts. As a result, there are two addresses consumed as overhead for each subnet. The more subnets, the more addresses that are lost from subnetting overhead. Figure 1 shows the fraction of a subnet that is lost to varying degrees of subnet overhead if that space is divided into subnets with prefixes of the specified length. Half of all addresses in a /30 are consumed by subnet overhead even with the lowest possible subnet overhead.

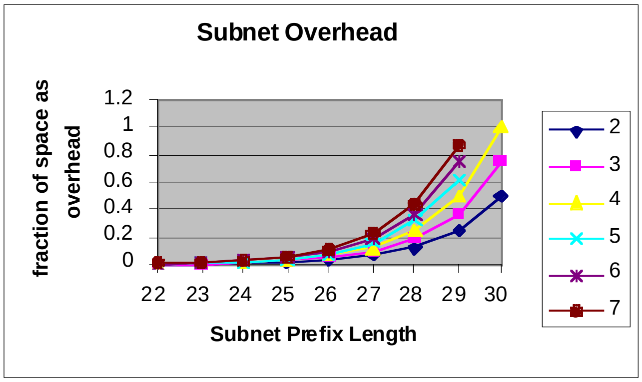


Figure 1: Fraction of overhead per subnet, depending on subnet length and overhead per subnet

The amount of overhead consumed per customer subnet affects usable address space availability. There are router redundancy techniques such as *Hot Standby Router Protocol* (HSRP) [13] that allow more than one router to handle traffic for a virtual interface. The overhead cost of HSRP is one virtual address and one address per router. Thus for a /29 segment with two routers using HSRP on the subnet interface, 62.5% of the available address space is consumed with overhead, leaving only three usable IP addresses. Figure 1 also shows how usable address space disappears with the increase in per subnet overhead.

From a customer and administration standpoint, it is also advantageous to have as much address space as possible. This makes it easy for a data center customer to expand operations by adding servers. Moving servers to a completely different, larger subnet in order to expand forces a customer to reconfigure all the hosts, typically involving significant downtime. As mentioned above, firewall and other access policies are often configured by subnet, and having as large a subnet as possible dedicated to a customer allows additions and changes to be made to servers without having to change those firewall and access policies.

Of course, since the supply of IP addresses is limited, customers cannot have all the space that might be convenient for them. Data center administrators must consider what happens when address space becomes nearly exhausted. In that case, they need to consider meeting Internet registry requirements to obtain new space. Typical registry requirements for new public IP space are as follows:

- 25% of the new space must be utilized immediately.
- 50% of the space must be used within one year.
- To get more space, the address space must be 80% utilized.

If these rules are not followed, the data center will be hard pressed to get more space if necessary. Again, the cost of giving smaller allocations of address space is that there will be more subnets and more addresses consumed with subnetting overhead.

The final consideration that a data center needs to evaluate is economic. An IP address has economic value, and if a customer is willing to pay for space that is unused, the value needs to be weighed against another customer using that space and also generating income.

EFFECTS OF ADDRESS SPACE CHOICES ON DATA CENTER IMPLEMENTATION

The scarcity of IP addresses drives many implementation and technology choices. In this section, we discuss some of these choices, particularly Network Address Translation, the complications of implementing Network Address Translation, and web server implementations.

Network Address Translation

A very useful service that an Internet data center can offer is connectivity between a customer's internal network and their servers hosted at the data center. One problem that such a service can encounter is conflict between private address spaces. The resolution of private IP address conflicts can affect address space design choices made at the customer end as well as those made for data center internal networks. Another major design factor is the preference to hide customers' internal networks from being seen in the data center network.

A popular technology used for resolving IP address conflicts is *Network Address Translation* (NAT) [14]. NAT helps translate IP addresses to a non-conflicting IP space and can be used to resolve IP conflicts that occur between a customer network and data center internal networks, as well as those that occur between two different customers' networks. There are two different modes of NAT:

- many-to-one or many-to-few NAT
- static one-to-one NAT

Many-to-one or many-to-few NAT entails hiding a set of networks or IP addresses behind a single IP address or a small pool of IP addresses respectively. A key characteristic of this form of NAT is that in addition to the IP being translated to a non-conflicting IP space, the port numbers are also translated to dynamically assigned port numbers to enable differentiation among the set of networks or IP addresses being hidden.

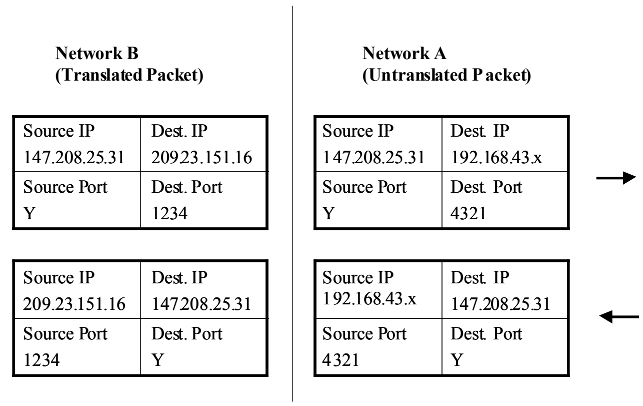


Figure 2: Many-to-one NAT packet flow

Figure 2 shows a sample flow of how a packet changes as it traverses the many-to-one NAT boundary between networks A and B. Any traffic traversing from network A (designated by 192.168.43.0/24) to network B has its source IP address translated to a single IP address (209.23.151.16). To differentiate the multiple hosts on network A trying to communicate with hosts on network B, the source port 4321 is also translated to a dynamically assigned port 1234 as the NAT boundary is crossed. The device performing NAT maintains a dynamic table of these IP addresses and port translations to ensure appropriate communication between network A and network B.

A consequence of using many-to-one or many-to-few NAT is that network communication cannot be initiated bidirectionally. Consider the example in Figure 2. Network communication can be initiated from hosts on network A to those on network B. Hosts on network B cannot initiate connections to hosts on network A because network A is hidden behind a single IP address 209.23.151.16, and hosts on network B cannot uniquely address a host on network A for communication. This could potentially be considered a security feature as well.

Static one-to-one NAT entails translating an IP address uniquely to another IP address. In addition, static NAT does not involve any port translation. Further, there is no restriction regarding which direction network communication can be initiated in because the device performing NAT can uniquely translate back to a specific host on the network with the conflicting IP space.

Based on network communication requirements, many-to-one or static NAT or both can be used. Later on, we discuss how Intel Online Services uses NAT for IP conflict resolution under various remote access scenarios. Extensive use of NAT also drives the need for choosing a device that allows enabling of flexible NAT configurations to fit the diverse requirements of multiple data center customers.

NAT may need to be used more than once as the network traffic traverses between the customer networks and the data center networks and back. An example of such a situation is when one customer's internal IP space conflicts with another customer's internal IP space. In such a situation, NAT would need to be performed at one of the customer ends to resolve IP conflicts with the other customer's IP space. NAT would need to be used a second time at the data center end to hide the customer network from data center internal networks.

To further complicate matters, there could be situations where a customer needs to run applications such as DCOM [16] that do not work across NAT. In such situations, alternate solutions like readdressing customer end systems to a public IP space and allowing that IP space to be visible within the data center may need to be explored.

Debugging Complications from NAT

While NAT can be a very useful tool for resolving IP address conflicts and enabling end-to-end connectivity to customer-end networks, the use of NAT can lead to complex troubleshooting scenarios [17]. In situations where NAT is performed more than once as a packet transits from its source to its destination, the IP address of an end system will change as many times as a NAT boundary is traversed. Debugging access issues in such cases requires intimate knowledge of the end-to-end network path and also a clear comprehension of which IP address is associated with an end system on a given section of that network path.

A packet sniffer is critical when troubleshooting potential connectivity issues across a device performing NAT. Visibility inside the packets captured on both sides of the NAT boundary helps establish whether the IP address translation is occurring as desired. Looking at the payload in a packet can also help identify if an application will work across a NAT boundary. Usually, applications that contain IP addresses or application port information in the packet payload will not work across NAT as the NAT process modifies the IP address only in the IP header and does not modify anything in the packet payload. Without visibility into a packet, this kind of troubleshooting could be long and tedious.

Use of NAT can also lead to configuration complications on the end systems since the translated address is valid on one side of a NAT boundary and the actual IP address may be valid on the other side of that NAT boundary. For example, when printing to a printer whose IP address has been statically translated (one-to-one), let's say from the actual printer IP of 172.16.20.5 to an IP address of 10.81.249.23, the print job will need to be initially sent to the translated IP address i.e., 10.81.249.23. Consider another case where the printer IP address is translated from its actual IP address of 172.16.23.15 to 192.168.23.26 at the customer end and is further translated to 10.81.249.35 at the data center. In this case, the system at the data center will need to send a print job to 10.81.249.35 in order to print to the actual printer at 172.16.23.15.

Furthermore, in the case of many-to-one or many-to-few NAT, the network traffic can only be initiated in one direction as discussed earlier. This should be kept in mind when troubleshooting connectivity issues across such address translation boundaries.

Virtual Web Server Implementation

Data center customers occasionally implement what are called *virtual servers*. This means that multiple web sites are implemented on the same set of servers. One way to implement this is to use a different IP address for each web site, with each web server having multiple IP addresses. Each IP address corresponds to a different web site. For example, let us say a customer has three web servers (1, 2, and 3) and two different virtual servers, www.site1.com and www.site2.com. Each web server would have two virtual IP addresses, one for www.site1.com and another for www.site2.com. www.site1.com would map to an IP address which is load balanced between the three site 1 IP addresses. www.site2.com would map to a different IP address which is load-balanced between the three site 2 IP addresses. This configuration consumes eight public IP addresses.

Extending this logic for **m** web sites and **n** web servers, **m * (n+1)** public IP addresses are consumed. This is a very wasteful way to use IP addresses. A better implementation uses virtual sites for customers with only one IP address per server. The web server produces content depending on the HTTP host request header [17]. As a result, for **m** web sites on **n** servers, only **n+1** IP addresses are consumed (including one virtual address per site). The shortage of IP address space dictates using this technique. Some registries will not accept virtual sites as an excuse for requesting additional IP addresses.

ADDRESSING DESIGN DECISIONS AT INTEL® ONLINE SERVICES

In this section, we discuss some of the implementation choices made at Intel Online Services that were driven by IP address space concerns. We first talk about the choice of IP address space. We then discuss remote access implementations, and conclude with a description of virtual web site implementations.

Addressing Choices

As we mentioned above, some types of address space are more appropriate than others in different situations. At Intel Online Services (IOS) , we use a hybrid addressing approach for the data center network that uses the most appropriate type of address space depending on the purpose of the host. We use private IP addressing for the data center internal networks and for customer servers that do not need to talk to the Internet. Since our internal networks do not need to talk to the Internet, there is no need to use precious public space. Also, back-end servers that do not need to talk to the Internet gain a measure of security because they are impossible to access directly from the Internet. (They are not immune to attack, however.)

Links to transit ISPs and other ISPs that IOS is peers with uses the address space of those ISPs on the router interfaces of the links. Since the IP address will need to change if the ISPs change and will go away if the ISP is no longer used, use of their space in this situation is not a problem and helps conserve IOS public address space.

For data center hosts and routers that need to communicate directly with the Internet, we have used a variety of address spaces. In North America, we use a class B (/16) that was made available to us. Using available public space allows us to have address space independent of our ISP selection, and it makes multihoming to multiple ISPs much easier. For our data centers in Asia and Europe, we have obtained space from the regional address registries that we own. Using the available class B was not feasible because some ISPs will not accept router advertisements of pieces of traditional class B address space. If we choose to use this space, we would have had to use the same ISPs in Asia, Europe, and in North America in order to have any kind of data center-specific routing policy. The process of obtaining new space took significant effort, but it is well worth it to have address space independent of ISP choice and the ability to multihome.

Size of Address Space Advertised

While IOS data centers are designed to handle thousands of hosts, not all of the hosts need to communicate directly with the Internet. Each data center advertises /19 networks, providing address space for up to 8192 hosts. /19 is the longest prefix that some ISPs will accept. For the data centers in North America using parts of a traditional class B, we also need to advertise the entire /16 network out of two data centers in order to deal with ISPs that do not accept parts of traditional class B networks. These ISPs will route traffic to IOS's ISPs, which will then route it to the individual data centers.

Autonomous System Number Choices

These choices are similar to our choices of address space. For North American data centers, IOS uses an autonomous system number that it had available. Separate autonomous numbers for each data center were considered wasteful, and in the North American environment, not particularly necessary. For IOS Asian and European data centers, we have obtained AS numbers for APNIC and RIPE, respectively. This allows the data centers independence in ISP selection, and it avoids any possible routing policy conflicts with other data centers on different continents.

Address Allocation Policy

For security and ease of management, IOS has chosen to place each of its customers on separate segments. In doing so, IOS enforces requirements for address utilization that mirror those of the address registries mentioned above. This positions IOS to be able to meet the utilization requirements of the registries when IOS requires more space.

To meet those requirements, IOS uses variable length subnet masks extensively. VLSM allows IOS to assign the appropriate sized subnet to a customer while maintaining utilization requirements. One consequence of this is that for IOS infrastructure hosts that need routing information sent to them, a routing protocol that understands VLSM needs to be used. This eliminates RIP version 1 protocol [3] and basically limits the routing protocol used by hosts to OSPF [4] and RIP version 2 [15].

Remote Access Implementation

IOS data centers offer a variety of remote access options such as Virtual Private Network (VPN), ISDN, and dedicated leased lines (T-1, T-3 etc.) to customers as a way of providing access directly from their networks into their servers hosted at IOS. Some of these options, such as LAN-to-LAN VPNs and leased lines, create network channels into the data center across which customer-end network addressing, that may very likely be in the private IP ranges, becomes visible. In this section, we focus only on these remote access options.

Allowing customer-end IP addressing, whether it is in public or private IP space, inside the data center network makes routing extremely difficult. The data center routing policy needs to account for routing network traffic appropriately to multiple customers' home networks. Furthermore, across multiple customers, the networks at their ends can be spread all across the public and private IP address ranges, leaving little scope for summarizing networks and as a consequence leading to larger routing tables. In addition, private IP conflicts across customers as well as the data center networks need to be resolved. Considering all the above challenges, we made a decision to hide customer-end internal networks from data center internal networks.

In this light, let us discuss the salient features of IOS's remote access infrastructure, which is logically represented in Figure 3.

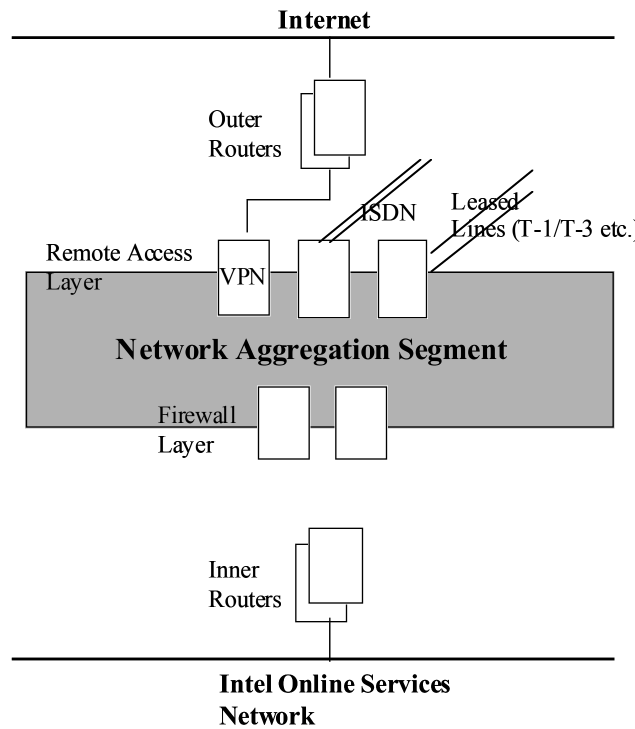


Figure 3: Intel Online Services remote access infrastructure for data center customers

The remote access layer constitutes the devices that provide the various remote access technologies supported at each data center. The firewall layer has devices that enforce security policies, among other things, on remote access traffic. The firewall layer also employs many-to-one as well as static NAT extensively to ensure that customer-end networks are not visible beyond the network aggregation segment. An exception to this rule is when a customer wants to run applications such as the Distributed Component Object Model (DCOM) [16] that does not work across NAT boundaries.

The actual design of the end-to-end network communication, across the above infrastructure, for various customers is dependent on the following factors:

- private IP addressing conflicts with other customers' networks
- where the network communication is initiated
- whether the applications that need to be run can work across NAT boundaries

Following is a discussion of the various remote access scenarios that can be encountered and how IOS supports end-to-end communication in those situations.

Scenario 1: Customer-end network does not conflict with any other customer's home networks or with data center internal networks. All network communication needs to be initiated from the customer end and inbound into the data center. All required applications work well with NAT.

This is a simple case and can be resolved by translating all of the customer network to one IP address (many-to-one NAT) from the data center's IP addressing range. Each customer network that fits this scenario is translated to a different IP address. As discussed earlier in the NAT section, many-to-one NAT does not allow for communication to be initiated in both directions. In this design, servers in the data center cannot initiate communication back to customer-end systems.

Scenario 2: This is the same as Scenario 1 but it requires that communication needs to be initiated from the data center to a few customer-end systems.

In such situations, we still create a NAT rule to translate all of the customer's networks to one single IP address. In addition, we create static one-to-one NAT relationships between specific data center's IP addresses and specific IP addresses of systems at the customer end. Hence, bi-directional initiation of communication is allowed only for specific customer-end systems.

Scenario 3: Customer-end internal networks conflict with another customer's home network but do not conflict with data center IP address space. Communication is always initiated from customer end to the data center and all required applications work well in spite of NAT.

To resolve the private IP address conflicts here, we ask customers to perform NAT at their end to hide their internal network from IOS behind either non-conflicting private IP address space or public address space. Furthermore, we translate that IP address space, at the firewall layer in the data center remote access infrastructure, to a single IP address from the data center's IP address space.

Scenario 4: This is the same as Scenario 3 except that communication needs to be initiated from the data center to some systems at customer end.

For allowing needed end-to-end communication here, we create static one-to-one NAT relationships for the specific systems at the customer end in addition to the solution implemented for Scenario 3. If customers are using many-to-one NAT on their end as well, they will need to set up corresponding one-to-one NAT relationships on their end in order to allow connections to be initiated from the data center to systems at their end.

Scenario 5: Customer-end networks have private IP conflicts with data center's private IP space.

With the exception of catering to applications that do not work across NAT, we have decided to allocate the data center's public IP space to the internal networks that such customers would need to reach at the data center. Customers in turn will need to perform NAT at their end to translate their networks to a non-conflicting IP space. This non-conflicting IP space is still hidden from the data center for reasons discussed earlier.

Scenario 6: Customer needs to run applications that do not work across a NAT boundary across one of the remote access channels.

In such cases, we must ensure that the data center's networks as well as the customer-end systems that need such communication use public IP addressing. In addition, we have to accommodate the routing for customer's public IP address range in the data center routing policy as it is not possible to hide customer-end networks behind NAT here.

Other Options: End-to-end connectivity design, across the remote access channels discussed in this section, can get quite complex and potentially difficult to troubleshoot. IOS also provides remote access options, such as client-to-LAN VPN and ISDN that provide direct connectivity from an individual desktop system to the data center network and are devoid of design complexities of the remote access options discussed so far. These remote access options hide the customer end IP addressing from the data center network, eliminating the possibility of address conflicts and the need for technologies such as NAT. Instead, these technologies assign an additional IP address from a designated pool of addresses in the data center's IP address space to the customer-end system while the system is connected. This IP address is used for all communication with systems within the data center. Applications that carry IP address or application port information in the packet payload work without any issue across these remote access channels, since the IP address and port information never changes anywhere between the customer desktop system and the systems in the data center.

This option works well for customers who travel a lot and access their systems from a number of locations or for customers who do not need continuous access to their systems. It does not work well for customers whose connectivity to their systems in the data center must be up all of the time.

Virtual Web Server Implementations

IOS uses the HTTP 1.1 host header technique [18] mentioned above for virtual web site public addresses. This means for **m** virtual web sites and **n** web servers, only **n+1** public addresses are used. With this implementation, the number of IP addresses required is not sensitive to the number of virtual web sites. Some IOS customers have web usage analysis packages that require that different virtual web sites have different IP addresses. IOS minimizes the impact on address space from such customers by mapping IP virtual addresses to web servers on ports other than the HTTP standard port 80. A different web server virtual IP address might map to port 81 on the web servers, while a different virtual site's virtual IP address might map to port 82 on those same servers. In this way, for **m** virtual web sites and **n** web servers, **m + n** public addresses are used. This is not as good as the host header implementation, but is much better than the IP address per web server implementation.

TRENDS IN IP ADDRESSING

IP address space is finite, and as the number of available addresses gets smaller, new addresses and AS numbers will be harder to obtain. As of February 2000, about half of the available IP addresses will be utilized [19]. We anticipate that depletion of addresses will accelerate as the Internet becomes more pervasive worldwide and as more and more devices (cellular phones, game consoles) are becoming able to communicate directly on the Internet.

One technical solution we are evaluating to reduce the number of addresses being used is to use NAT on public web servers. With NAT, a public virtual IP address could be mapped to multiple private addresses, reducing the need for a public IP address per web server. This solution has the potential drawback of complicating monitoring of web servers and debugging of problems since the individual web servers can no longer be individually contacted by the Internet. Also, various security schemes can be broken by using NAT [20].

A longer term solution to the depletion problem is for the Internet to move to IP version 6 [21]. The address space for IPv6 is much much larger, and many of the actions necessitated by IPv4 address scarcity will not be necessary. The address registries have already begun allocating IPv6 address space. Unfortunately, IPv6 is not backward compatible. At the moment, there is not enough economic incentive to undertake large scale conversion to IPv6. We anticipate that this may change as the amount of IPv4 address space is depleted.

CONCLUSION

IP address space is clearly one of the most critical resources that an Internet data center needs to manage. The fact that IP address space is a limited resource drives many technology and operational decisions. Even private address space, once thought to provide relief from addressing problems, can be the source of problems as two organizations find themselves trying to address private space address collisions. IPv6 holds some promise for relieving many of these problems, but the Internet has quite a ways to go before there is widespread adoption of IPv6.

ACKNOWLEDGMENTS

We acknowledge our technical reviewers, Sanjay Rungta and Matt W. Baker for their time and helpful feedback.

REFERENCES

[1] Postel, J (editor), "DARPA Internet Program Protocol Specification," *RFC 791*, September 1981, http://www.ietf.org/rfc/rfc0791.txt.

[2] Hawkinson, J., and T. Bates, "Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)," *RFC 1930*, September 1996, http://www.ietf.org/rfc/rfc1930.txt.

[3] Hedrick, C., "Routing Information Protocol," *RFC 1058*, June 1988, http://www.ietf.org/rfc/rfc1058.txt.

[4] Moy, J., "OSPF Version 2," *RFC 2328*, April 1988, http://www.ietf.org/rfc/rfc2328.txt.

[5] Rekter, Y., and T. Li, "Border Gateway Protocol 4," *RFC 1771*, March 1995, http://www.ietf.org/rfc/rfc1771.txt.

[6] Cerf, V., "IAB Recommended Policy on Distributing Internet Identifier Assignment and Recommended Policy Charge to Internet 'Connected' Status," *RFC 1174*, August 1990, http://www.ietf.org/rfc/rfc1174.txt.

[7] Nesser II, P., An Appeal to the Internet Community to Return Unused IP Networks (Prefixes) to the IANA, *RFC 1917*, August 1990, http://www.ietf.org/rfc/rfc1917.txt.

[8] Manning, B., and T. Pummill, "Variable Length Subnet Table For IPv4," *RFC 1878*, December 1995, http://www.ietf.org/rfc/rfc1878.txt.

[9] Mogul, J. and J. Postel, "Internet Standard Subnet Procedure," *RFC 950*, August 1990, http://www.ietf.org/rfc/rfc0950.txt.

[10] Rekhter, Y., B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, "Address Allocation for Private Internets," *RFC 1918*, February 1996, http://www.ietf.org/rfc/rfc1918.txt.

[11] Rekter, Y. and T. Li, "An Architecture for IP Address Allocation with CIDR", *RFC 1518*, September 1993, http://www.ietf.org/rfc/rfc1518.txt.

[12] Fuller, V., T. Li, J. Yu, and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy," *RFC 1519*, September 1993, http://www.ietf.org/rfc/rfc1519.txt

[13] Cisco Corporation, *Hot Standby Routing Protocol*. http://www.cisco.com/warp/public/619/index.shtml

[14] Evegang, K. and P. Francis, "The IP Network Address Translator (NAT)," *RFC 1631*, May 1994, http://www.ietf.org/rfc/rfc1631.txt.

[15] Malkin. G., "RIP Version 2," *RFC 2453*, November 1998, http://www.ietf.org/rfc/rfc2453.txt.

[16] Microsoft Corporation http://msdn.microsoft.com/library/backgrnd/html/msdn_dcomfirewall.htm

[17] Srisuresh P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," *RFC 2663*, August 1999, http://www.ietf.org/rfc/rfc2663.txt.

[18] Fielding, R., J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1," *RFC 2068*, January 1997, http://www.ietf.org/rfc/rfc2068.txt.

[19] Mankin, A., "Will there be a IPv6 Transition?" 1999 USENIX Annual Technical Conference, Monterey, 1999, http://www.east.isi.edu/~mankin/usenix99/tsld001.htm

[20] Carpenter, B, "Internet Transparency," *RFC 2775*, February 2000, http://www.ietf.org/rfc/rfc2775.txt.

[21] Deering, S., "Internet Protocol, Version 6 (IPv6) Specification," *RFC 2460*, December 1998, http://www.ietf.org/rfc/rfc2460.txt.

AUTHORS' BIOGRAPHIES

Jeff Sedayao is a network engineer at Intel Online Services. Between 1987 and 1999, he was the architect of and ran Intel's Internet connectivity. His primary interests are in Internet performance, security, and policy implementation. He also serves as Intel's representative to the Cross Industries Working Team's Internet Performance Team and has participated in the IETF's IP Performance Metrics Workgroup. His e-mail address is jeff.sedayao@intel.com.

Lynne Marchi received her B.S. degree in computer science from California State University, Sacramento. She began work at Intel in 1992 as a co-op and then joined the Corporate Information Security group in 1993. In 1996, she joined Internet Connectivity Engineering where she focused on the secure implementation of new firewalls. Recently, she has joined the Internet Firewall Engineering group of Intel Online Services. Her e-mail address is lynne.c.marchi@intel.com.

Sridhar Mahankali is an M.S. graduate in electrical and computer engineering from the University of Rhode Island, Kingston. He worked as a systems and network engineer with the Board of Governors of the Federal Reserve System before joining Intel's e-Business group in 1998, where he focused on firewall and intrusion detection implementation. Currently, he is part of the Internet Firewall Engineering group in Intel Online Services and specializes in engineering VPNs and network consulting. His e-mail address is sridhar.mahankali@intel.com.

Certifying Service Reliability in Data Centers

Michael Schlierf, Intel Online Services Quality and Reliability Engineering, Intel Corp.
Michael Fuller, Intel Online Services Quality and Reliability Engineering, Intel Corp.
Mike Kerr, Intel Online Services Quality and Reliability Engineering, Intel Corp.

Index words: data center, service, reliability, start-up, cookbook, ORC, operations, readiness, certification, SRCL, CIPA, WTP, OJR

ABSTRACT

Intel® Online Services (IOS) Quality and Reliability Engineering (QRE) applied the Virtual Factory and Copy Exactly (CE!) [1] Start-Up methodologies from Intel's Technology Manufacturing Group (TMG) to the challenge of starting up an IOS Data Center (DC). Our approach was to take existing methods for solving specific problems and modify them for service industry needs. The results are the Data Center Start-Up "Cookbook" and the Operations Readiness Certification (ORC) process. IOS uses Operations Readiness Certification to certify the reliability of systems and service offerings in each data center.

INTRODUCTION

Intel® Online Services is chartered to deliver managed services to host customers' unique e-Commerce applications. Our challenge is to leverage Intel's expertise in producing millions of units reliably and to apply it to rapidly developing and deploying new technologies, with a high level of customization.

The e-Commerce community is unforgiving of mistakes, and IOS cannot afford "black eyes" from poor execution. New data centers must be ready for customers within 54 days of construction being completed. The IOS business plan requires simultaneous data center start-ups with many new Intel personnel. On opening day, a data center must immediately provide reliable service consistent with other IOS data centers.

After opening the first data center (named IDS01) in Santa Clara in September, 1999, the QRE team organized a rigorous Lessons Learned process in which we identified hundreds of issues and suggested improvements. We developed specific action plans to resolve the major issues before opening the second data center (IDS02), which was scheduled for six months later.

As a result of the Lessons Learned and action plans, QRE developed an IOS Operations Start-Up Methodology consisting of 1) the DC Start-Up process (the Cookbook) and 2) the DC Operations Readiness Certification (ORC) process.

DATA CENTER START-UP COOKBOOK

The DC Start-Up Cookbook is a set of recipes for ramping up new data centers. The cookbook contains mechanisms for planning and scheduling, managing the project, and capturing lessons learned for subsequent start-ups.

Planning and Scheduling

The cookbook contains standardized phases and critical-path milestones from the following groups:

- Corporate Services (CS) Construction (limited to IOS-dependent milestones)
- Information Technology (IT) (limited to IOS-dependent milestones)
- Asset Management (AM)
- Intel Online Services Engineering (IOSE) Fit-up
- Staffing
- Human Resources / Employee Integration
- Organizational Resource Development (ORD)—Functional Training
- Customer Service Integration (CSI)
- Data Center Operations (DC Ops)
- Facilities Sustaining
- Customer Service Representatives (CSR)

The cookbook defines the key supporting tasks and their duration, sequence, and dependencies. It includes a detailed technical (IOSE Fit-up) sub-schedule that can be integrated into the master schedule.

Each phase/milestone has a standard duration relative to the "Go-Live" date. Representatives of all groups involved hold a full-day planning session to develop a detailed plan from the generic cookbook milestones. Each group defines its critical-path milestones to ensure that all groups understand and agree, particularly groups with direct interdependencies.

Managing the Project

Throughout the start-up phases, representatives of each involved group hold a weekly meeting to manage the IOS start-up milestones and group interfaces. Each week all group owners present a Weekly Risk Assessment template for their critical path milestones and assign a risk assessment (high, medium, or low) to meeting their milestones. They also present highlights/lowlights and plans, and they request help if needed. A Weekly Readiness Assessment Report is generated with Program Highlights, Lowlights, and assessed risk to the critical path.

Lessons Learned

The IOS Operations start-up methodology ends with a rigorous *Lessons Learned* process from the previous data center start-up. This is organized into *two* sessions:

- Session 1 (DC Operations) focuses on staffing, training, human resource allocation and utilization, management/planning, IOS Ops certification processes (SRCL Planning & Management, Customer Integration Process Audit, Walk-the-Pod, On-the-Job Readiness), and business processes (customer landing, build team, and change management).
- Session 2 (IOSE Fit-up) focuses on fit-up technical issues, planning and coordination of fit-up interdependencies to CS construction and IT, and SRCL technical content.

In these sessions, the stakeholders define Action Plans and assign Actions Required (AR) to specific owners or forums. Those that affect the data center start-up plans and schedules are included in the next version of the IOS Operations Start-Up Cookbook.

The purpose of the cookbook is to ensure reliable deployment of a service as measured by successful completion of the Operations Readiness Certification.

OPERATIONS READINESS CERTIFICATION (ORC)

The Operations Readiness Certification process validates that the data center is able to support customer sites with standard production processes and personnel. It consists of the following phases:

- The *System Readiness Checklist* (SRCL) verifies that the infrastructure is built correctly and that all systems and subsystems are operational.
- The *Customer Integration Process Audit* (CIPA) ensures that all business processes required to land a customer in the data center are in place.
- A series of *Walk-the-Pod* (WTP) exercises or *scripts* validate that the data center has the capability to support Service-Level Agreements (SLAs). The ORC team develops a risk assessment matrix to identify high-risk areas requiring immediate action plans.
- During On-the-Job Readiness (OJR), auditors observe all data center personnel performing their roles and responsibilities to validate that they have the required skills.

When all of these phases are completed successfully, the DC is certified as ready to "go live."

Systems Readiness Checklist (SRCL)

The first phase of IOS Operations Readiness Certification is the Systems Readiness Checklist (SRCL) process, based on the Intel Technology Manufacturing Group's Machine Readiness Checklist (MRCL) process. SRCLs are an integral part of starting up new data centers as well as supporting new service offerings.

SRCL deliverables are grouped into three levels:

1. The *build* checklist allows the system builder to capture completion of the build elements.
2. The *functionality* checklist validates all required system functions.
3. The *documentation* checklist verifies that all documentation required for transition to data center operations is in place.

The Build Checklist (SRCL Level 1) coincides with the IOSE Fit-up. The Functionality Checklist (Level 2) occurs the week after the Build Checklist. The Documentation Checklist (Level 3) begins the week after the Functionality Checklist and continues through "Go-Live."

A SRCL has several benefits:

- It provides the foundation for a standardized data center start-up process.
- It stipulates the requirements and deliverables for the system and ensures that those deliverables are met.
- It provides a standard qualitative assessment of the technical readiness of each system or subsystem.
- It creates a measurable endpoint for the system installers responsible for deployment
- It specifies what routine maintenance activities are required for DC Operations to assume ownership of and sustain the system.
- It is a means for DC Ops analysts and engineers to learn system details and reinforce technical training.

- It serves as a vehicle to commission the system to production in the data center. (The SRCL requires written sign-off by the system installer, DC Ops system owner, and ultimately the DC Ops manager.)
- It provides a framework for continuous improvement for future deployments.

There are a number of specific SRCL management processes in place to manage completion of SRCLs by service offering. There is also a specific management and team structure defined to complete SRCL execution.

The SRCL Functional Owner Matrix assigns groups of SRCLs to a DC Ops functional owner. This functional owner is responsible for the completeness and accuracy of technical SRCL content and also drives the completion of SRCLs within their functional area.

Customer Integration Process Audit (CIPA)

The CIPA process exercises the Customer Landing Process, as defined by Customer Service Integration (CSI). CSI determines which specific CIPA forms, processes, procedures, and roles need to be evaluated for each service offering then ensures that all business processes required to land a customer in the data center are in place.

The overall purpose of the CIPA is to certify the data center's ability to build a simulated customer *pod* (which typically consists of web application and database servers) then land the customer's e-commerce application around the clock, using only the skills and resources within the data center. The completed pod serves as the platform for the Walk-the-Pod simulations.

In a typical scenario, the implementation team

- blocks out a 72-hour period to conduct the CIPA,
- gathers all known procedural documentation,
- builds a faux pod (for a simulated customer) from Customer Enrollment through CSR Start-Up,
- presents Customer Build Request (CBR) to the Implementation Planning Council (IPC), Planning Meeting, and Change Assessment Team (CAT), and
- creates a library of all documents that need to be validated after the audit.

Throughout the CIPA, independent auditors observe every step.

The CIPA allows the implementation team to

- test all documented procedures, to remove unnecessary steps, and identify missing steps;
- identify procedures that are passed on through “tribal knowledge” or are not defined;
- identify processes that must be performed in serial or may be performed in parallel;
- test document control; and
- test revision control.

The CIPA also allows the implementation team to

- follow a Remedy ticket throughout the process, conducting warm handoffs between roles;
- uncover major issues in the landing/build/support processes;
- identify important training issues;
- test the project tracking tools;
- review the network design;
- ensure that all required asset, server, and network data are captured;
- develop Best Known Methods (BKM) with Customer Service Integration (CSI) and Information Security (InfoSec); and
- train other network engineers on the processes.

Once the implementation team has completed the Customer Integration Process Audit, the pod is ready for the Walk-the-Pod exercises.

Walk-the-Pod

Walk-the-Pod (WTP) is a concept adopted from Intel manufacturing that is used for validation of a new process or process change. The goal of WTP is to assess the operational readiness of a new/upgraded service or tool and validate that it is fully integrated into a production data center. Testing personnel or training readiness is *not* within the scope of WTP.

The purpose of WTP is to

- ensure that supporting technical and business documentation (such as production designs and run books) can be located and executed successfully;
- exercise procedures between support levels;
- exercise all monitoring, fault detection, troubleshooting, escalation, and problem resolution procedures; and
- ensure that the technology and processes support both customer and internal service-level agreements.

Planning Phase

During the planning phase, assigned team members create a set of simulation exercises, or scripts, covering a representative range of the promised capability. Each script describes a specific service scenario. WTP scripts are organized by the primary capability they evaluate and contain the following components:

- Targeted Primary Capability
- Required Procedures and Processes
- Required Preliminary Training
- Scenario Set-Up
- Expected Process Flow
- Expected Outcome

The best practice is to cover each primary capability and key process with several exercises (with primary and secondary focus). For example, a network-focused exercise will simulate a network outage monitored, corrected, and verified by specific management tools (network availability and capability). Resolving the network outage also verifies the problem management workflow.

In order to develop robust simulation scripts, the team members use a *capability matrix* that ensures that adequate tests are run against critical capabilities. The capability matrix also shows which capability, process, or tool is either over- or under-tested, thus maximizing the resources allocated to WTP.

Validation Phase

During the validation phase, team members perform a dry-run validation to make certain that the tools, processes, and workflow function as expected. Key questions to answer during the dry-run validation are as follows:

- Can the scenario script create the error needed?
- Is the physical infrastructure correctly described and tested?
- Are there missing processes or procedures?
- Is the operational business model correctly assumed?

During Walk-the-Pod validation, Data Center Operations personnel perform the simulation exercises to verify that individual components, tools, processes, procedures, and documentation required for landing, monitoring, and supporting a service offering exist and function as an integrated system.

Execution Phase

Before the execution phase can begin, the SRCLs and test pod must be completed, and the planned scripts must be validated in the specific environment to be tested.

As the Walk-the-Pod team runs each exercise, team members perform specific roles. During the execution of each key function, an auditor captures changes, issues, or needed action items. After each exercise is run, the team does a postmortem on the effort to determine if the issues found require a re-write and re-run of the scenario.

The final output of executing the simulation exercises is the Summary Report. It captures each scenario, associates a risk level (High, Medium, or Low) to each exercise and primary capability, and reports any open issues or required action items, as discovered through Walk-the-Pod.

On-the-Job Readiness

Once Walk-the-Pod verifies that the system is ready, On-the-Job Readiness (OJR) verifies that the people are ready to support the system.

The purpose of OJR is to ensure that command center personnel can

- learn the new service or tool in a simulated environment,
- execute realistic scenarios,
- perform all procedures between support levels,
- perform all troubleshooting, escalation, and problem-resolution procedures, and
- support both customer and internal service-level agreements (SLAs).

Planning Phase

During the planning phase of OJR, the implementation team re-uses the Walk-the-Pod exercises and categorizes them by job classification.

Validation Phase

During the validation phase of OJR, the implementation team members make certain that the scenarios and workflows reflect realistic situations.

Key questions to answer during the validation are as follows:

- Does the scenario apply to this (regional, non-regional, or tier-3) data center?
- Does the workflow identify all critical decisions?
- Does the command center person have access to all the resources required?
- Does the process support all related SLAs?
- Are there missing processes or procedures?

Execution Phase

During the execution phase of OJR, DC Operations personnel perform the simulation exercises to verify that they can follow all related procedures, perform the required tasks, and consistently make correct decisions, all within the constraints of the service-level agreement for the new service or tool in a production environment.

As the data center personnel run each exercise, performing their specific roles, an auditor captures changes, issues, or needed action items. After each exercise, the OJR team members evaluate the results to determine if the issues found require revising or re-running the exercise.

At the completion of the OJR, each data center analyst or engineer on each shift must be able to perform all required procedures. Full OJR status is achieved when all data center personnel on each shift have completed this skill validation.

The Post Deployment Summary details each scenario, associates a risk level (High, Medium, or Low) to each exercise, and reports any open issues or required actions.

CERTIFICATION DECLARATION

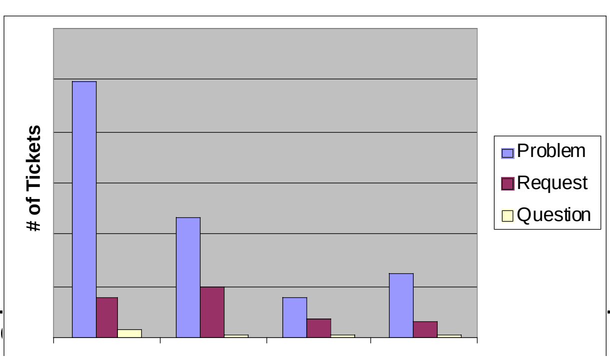
The Ops Readiness Certification process provides data center management with an objective assessment of their readiness to “go live.” The result of the SRCL, Walk-the-Pod, and On-the-Job Readiness exercises is a “punch list” of processes and procedures that may not be complete. Data center management makes an informed-risk decision regarding whether to “go live” immediately, or to complete the punch items before landing customers. When the data center has completed all phases of the Operations Readiness process, QRE issues a memo to the data center certifying that it is ready to support customers.

RESULTS

As a result of following the Data Center Start-Up Cookbook, IOS experienced near-flawless execution when opening the second data center (IDS02) in Chantilly, Virginia, in March, 2000. The Fit-up and Go-Live processes were completed on schedule, three weeks faster than in the prior start-up, in IDS01.

DISCUSSION

IDS01 (in Santa Clara, CA), IDS02 (in Chantilly, VA), and IDS03 (in Winnersh, UK) are similar data centers in the sense that they are the same relative size, have a similar number of customers, have comparable service offerings, had permanent staff during the start-up period, and their operating support systems were implemented in the local language. IDS04 (in Tokyo, Japan) is significantly different in these areas.



The declining total Ticket Count for the three data centers suggests that the Ops Readiness Certification process reduces the number of critical issues causing customer problems during the six weeks after data center start-up (see Figure 1.)

Figure 1: Ticket count

The progressive decrease in the problem tickets in the Operating Systems Service (OSS) category, where we can best measure the integration of technology and people, also furthers the conclusion that the ORC process increases the "Opening Quality" (see Figure 2).

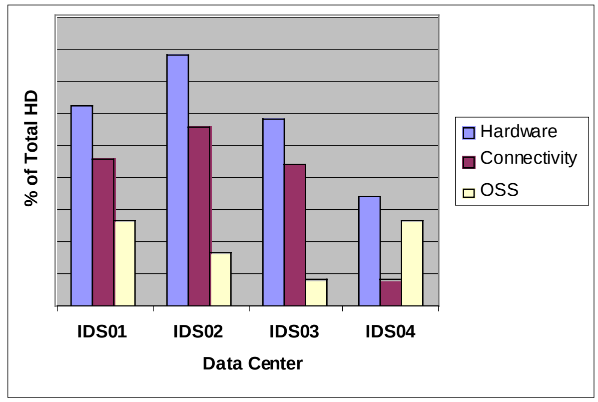


Figure 2: Category %

The decrease in the ticket Resolve Rate from IDS01, where we did not execute an ORC, to IDS02 and IDS03 where we did supports our increased confidence in our "Opening Quality" (see Figure 3).

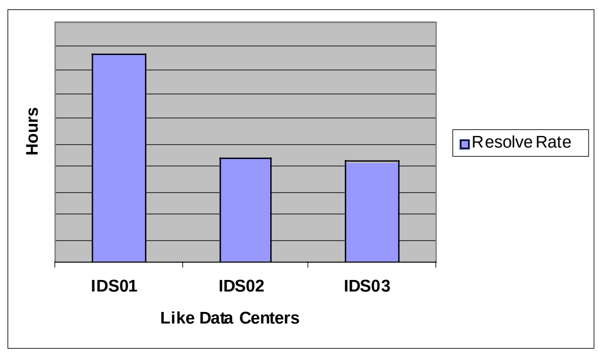


Figure 3: Resolve rate

CONCLUSION

Using the Data Center Start-Up Cookbook and Operations Readiness Certification (ORC) methodology, IOS has increased our confidence in data center quality at opening without lengthening the ramp-up time.

This confidence level prediction is supported by the following datasets: 1) the decreasing number of trouble tickets within each data center during the first six weeks; 2) the decreasing percentage of total tickets in the Operating System Support (OSS) category; and 3) an overall decrease in the time required to resolve problems. The data demonstrate that, between similar data centers (IDS01, 02, and 03), ORC produces a continuous improvement process.

ACKNOWLEDGMENTS

We acknowledge Steve Dickson, the QRE Deployment Manager, for creating the SRCL process and documenting a significant part of the Ops Readiness Certification (ORC) process. He has played a major role in applying the ORC methodology in IDS02, 03, and 04.

Stephen Dobbins played a key role in developing the SRCL process. Michelle Sugasawara and Michelle Griffiths were significant in executing the ORC process and interpreting the results, which improved the process for IDS03.

REFERENCES

[1] C.J. McDonald, " The Evolution of Intel's Copy Exactly Technology Transfer Method, Intel Technology Journal, Q4, 1998 at http://developer.intel.com/technology/itj/q41998/articles/art_2.htm.

AUTHORS' BIOGRAPHIES

Michael Schlierf is the Intel Online Services Quality and Reliability Engineering Manager. He has 20 years of experience with Intel in Industrial Engineering, Semiconductor Fabrication Operations Management, Factory Automation, and Information Technology. His e-mail is michael.c.schlierf@intel.com.

Michael Fuller is a QRE Program Manager with over 10 years experience in program analysis and program management, within both technology and applied behavioral analysis. His e-mail is michael.fuller@intel.com.

Mike Kerr is a professional technical writer with over 20 years of experience documenting user interfaces, programs, and processes. His e-mail is mike.kerr@intel.com.

e-Business Asset Management and Capacity Planning

Brad Coyne, IT e-Business Integration Engineering, Intel Corp.
Sonja K. Sandeen, IT e-Business Integration Engineering, Intel Corp.

Index words: asset management; capacity planning; facilities management; infrastructure management

ABSTRACT

This paper describes Intel's e-Business data center asset management and capacity planning programs including the business drivers that led to their creation, the technology developed to build and maintain them, and Intel's future plans for them.

As Intel Corporation's Internet presence grew from static content served from a single desktop PC to over 65 externally facing Internet applications running on over 850 servers in multiple states, the need for asset management and capacity planning programs became very obvious. This paper explains these programs in depth. We will first, however, give a brief overview of each program.

INTRODUCTION

Asset Management

Intel's e-Business asset management program is a single web-based source from which all the physical and configuration data for each of the servers in the e-Business production and pre-production environments can be obtained. This program was initially planned with four sequential phases and it is still evolving today. Effective asset management relies on both process and tools coupled with the discipline of the using audience.

There are specific processes created and implemented in order to maintain strong data integrity. Architectural design review boards review the requirements and approve them prior to any assets being procured for Intel's e-Business environment. With each of the processes explicitly outlined in the intranet web site, each engineer is responsible for updating the database when there is a change in the assets.

From a tools perspective, to make asset management easy for engineers, Intel's e-Business team has created a central web application for recording the perpetually changing asset information. This asset management application utilizes two back-end databases that capture physical information as well as configuration data. The two repositories are then seamlessly integrated into a single web portal for engineers to view and update vital server information. The overall infrastructure for the asset management application consists of one web server, two database servers, and a system-side agent for software and hardware inventory. The asset management application can be used by Intel's e-Business team for anything from determining where a system is located to what applications are running on that system. Future goals of the asset management application will be to integrate branding information (system specific documentation), system capacity information, and health monitoring controls and data. In e-Business, asset management is seen as more than just recording serial numbers. Our goal is to provide a comprehensive system that can be easily used to manage, administer, and evaluate Intel's e-Business assets.

Capacity Planning

Intel's e-Business is undergoing exponential growth rates due to the corporate emphasis on Internet commerce. As our Internet business grows, the need to forecast system and facility capacity requirements grows as well. Capacity planning is divided into two very large categories: system management (which includes server, network infrastructure, and storage area networks) and facilities/infrastructure management (which includes physical space, network connectivity requirements, and power and cooling requirements specifically related to the data centers and pre-production labs). Capacity planning is a crucial part of the entire planning process and integration for Intel's e-Business infrastructure because of Intel's demand for 99.999% application availability.

Every year Intel's e-Business more than doubles the number of systems used to support our Internet applications. To prevent our e-Business from outgrowing its facilities, system resources, and network infrastructure, we have to use a variety of methods to not only predict, but to also control capacity. Like asset management, Intel e-Business has invoked programs dedicated to both technical and process-based solutions to allow for effective capacity planning.

THE EVOLUTION OF ASSET MANAGEMENT AND CAPACITY PLANNING

Asset management and capacity planning are viewed as behind the scenes operational processes and tools not often represented in a very public manner. If one does not hear about asset management and capacity planning, it means that the processes and tools are working effectively and are optimally integrated to provide all of the essential business planning and analytical data required at any given time. This, we feel is the true success criteria of good operational asset management and capacity planning programs.

To understand the need for well-defined, integrated, and scalable asset management and capacity planning programs for any business unit is the first step to effective and efficient programs.

These programs are inherently evolutionary. As e-Business and the Internet grow, so too will the requirements to manage assets and plan for capacity.

THE TECHNOLOGIES AND DESIGNS THAT ENABLE ASSET MANAGEMENT

In order to successfully implement an asset management program within e-Business, a harmony must exist between the processes that define asset management and the tools that enable those definitions. This portion of the paper will explain the enabling technologies and designs that were modeled from the processes.

As the processes for asset management were constructed, based on an evaluation of the existing environment and the projected environment, careful consideration was given to the development of an application that would suit both realms. When designing an Asset Management Application (AMA), our first priority was to utilize existing data and processes. We then developed an application that would immediately leverage the current system in use as well as define a system that would enrich asset management as it evolved to the desired state. The following sections of this paper describe the retrofitting and development of a comprehensive application that enabled Intel's e-Business to manage its assets from both a corporate capital and business support perspective.

Re-Developing Asset Management

Effectively managing over a thousand capital valued systems is a challenge for any company or business. To meet that challenge within Intel's e-Business structure, we were chartered with not only retromanaging the existing environment, but also with controlling an exponentially growing farm of servers. Step one of the asset management challenge was to leverage existing data and applications.

The management of assets within Intel's e-Business has gone through as many revisions and evolutions as the Internet itself. The first attempt of asset management was a simple spreadsheet maintained by numerous engineers. Such a static effort was, at the time, all that was needed to manage a business consisting of only a handful of servers. As Intel's e-Business grew, and the value of asset management was recognized, more dynamic applications were implemented. The first draft of such an application consisted of a free-text web form with a simple one-table database. Although this development was pointed in the right direction, the demands of e-Business and Intel Corporation soon overcame its capabilities. To address the obvious needs at the time, management created an internal group that was chartered with corralling and coordinating Intel's e-Business assets. This newly formed entity consisted of both process and application developers.

Once processes were developed that outlined asset management, the application developers needed to implement a cost-effective solution. Third-party applications were considered briefly. However, the needed level of scalability and customization together with the inflated costs of outside applications quickly paved the way for an in-house solution.

It was determined that the first step in instigating asset management was to create a system that would not only host all the needed information, but also make the acquisition of that information as simple as possible for the users.

Creating a Dependency

To create such a system, first a database was designed that could hold physical asset information as well as integrate links to other systems and information that were used by engineers. The reasoning behind such cross-functional integration was the realization that an engineer rarely needs corporate-level asset information. By integrating asset management into more daily processes and applications, Intel e-Business created a dependency that would enforce the discipline required by asset management processes.

One logical opportunity to create such a dependency was the already existing process for system space and networking requests. In order for a system (asset) to land within Intel's e-Business environment, an engineer must have two things: physical space assigned to place the system in the data center and an IP address with the accompanying DNS entry. By taking advantage of these two necessities, the application developers were able to enforce the collection of asset information. In order to acquire rack space and networking needs for a landing server, an engineer must first pass through the asset management processes.

To facilitate this dependency, the developers formed an alliance with the facility and networking groups. This new alliance not only benefited the asset management process but also improved the efficiency of the joining partners. By integrating multiple processes and interrelating the data within, duplicate efforts were eliminated, and valuable data were joined.

Adding Features

Through dependencies, the asset management effort was assigned to the e-Business Integration Engineering group. However, to implement a successful application and process there needs to be tangible benefits to the user. As mentioned previously, the typical engineer will not likely benefit directly from asset management. Therefore, to truly create a business-wide need for asset management, the developers needed to create business-wide functionality.

An opportunity to do just this presented itself in the form of the integration of systems management. Intel's e-Business had an existing system for dynamically accumulating system and application data. The systems management infrastructure consisted of a system-side agent that inventoried software and hardware configuration data for each server within the e-Business environment. The asset management program developers realized that if they could provide a portal to the asset data within the asset application, they would be adding value to everyday operations. By altering the installation of the systems management tool to incorporate a logical link to the asset management database, a virtual conglomerate system was created. With the marriage of configuration and asset data, engineers now had an everyday use for the e-Business asset application.

The creation of dependencies and the addition of every day functionality to Intel's e-Business AMA has significantly contributed to the success of the asset management program. However, limiting the scope of the application and its user interfaces was not part of the application's design. Further development and interfaces are on the horizon.

FUTURE DEVELOPMENTS

Within Intel e-Business we are constantly pushing for tight integration of applications, of which asset management is no exception. Although the AMA is now coupled with our systems management data, there are even more opportunities for integration. The goal of the AMA is to provide a comprehensive portal to all and any system data. The portal will eventually be the one interface engineers and managers alike will visit in order to view any system information. The end result will be the transparent integration of all system data ranging from a system's physical and financial data to a system's configuration and build documentation.

The next step for the AMA is to link the interface and data to branding documentation. Intel e-Business maintains a document for each system within the environment. The branding documentation outlines all the steps used to build and configure a system. Branding information is crucial to controlling and building our production environment. Today, branding is viewed through shared directory browsing. A project is now in place to improve the branding process by initiating content control through version control. In addition to version control, branding documents will be managed through a central database.

The AMA developers see the upcoming branding application as yet another component that can be integrated into the AMA portal. By adding the branding component we will create even more dependencies and increase the functionality of the AMA for our end users.

With the linking of branding and systems management applications, the AMA will be an advanced infrastructure consisting of a web server and three back-end databases. The real benefit the AMA will provide will be the seamless joining of multiple data repositories. Where databases and the data once stood alone, they now will be fully integrated into one portal (see Figure 1).

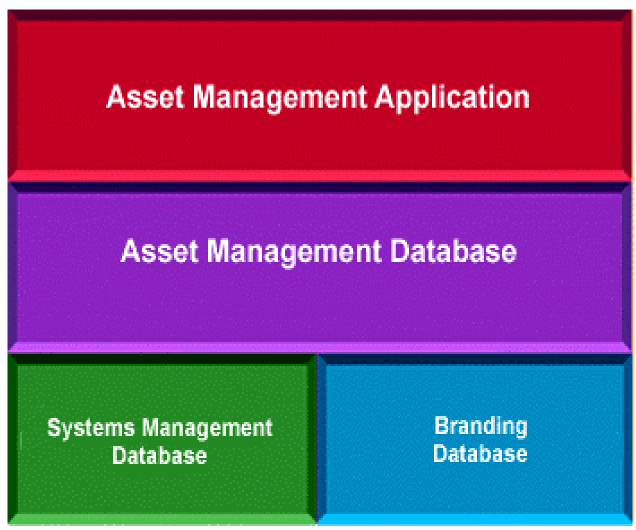


Figure 1: A view of the AMA components as they are scoped for today

After branding information is brought into the AMA there are even more possibilities on the horizon. Plans are being considered to also incorporate our capacity management and monitoring applications into the AMA portal. Assuming Intel e-Business can create this immensely conglomerated system, the AMA portal will undeniably be the one-stop interface for complete systems, asset, capacity, and monitoring management (see Figure 2).

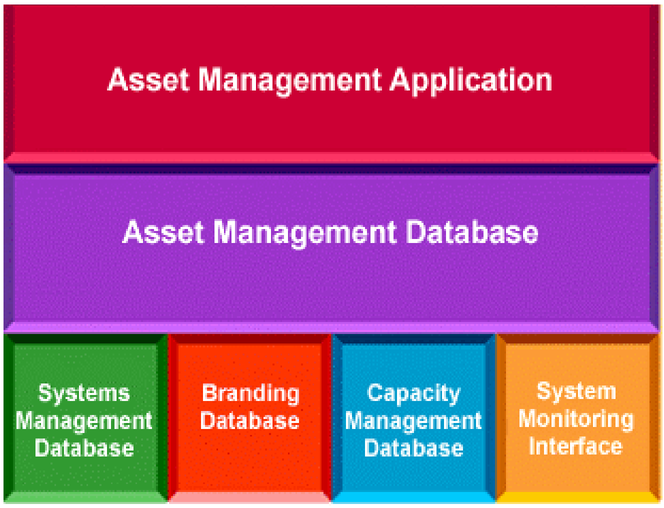


Figure 2: A view of the AMA components as they will be viewed in the future

CAPACITY AND SYSTEMS MANAGEMENT

Efficient management tools can greatly ease the management load in any computing environment, especially with an architecture that includes large numbers of systems. The Intel e-Business infrastructure includes over 850 servers, approximately 60% of which are part of the production environment. All servers are remotely monitored for CPU, memory, and disk utilization. This helps management staff plan for upgrades and provides useful baseline information for sizing new systems and applications. It also helps ensure that overtaxed systems are discovered before they suffer unnecessary failures or performance degradation.

In order to proactively monitor the capacity of systems within our e-Business environment, Intel's IT group has developed a server-side agent that captures dozens of system performance counters every five minutes. The performance data are then aggregated on several large databases exceeding a _ terabyte of performance data (Figure 3). Intel e-Business then runs daily reports for each of its systems, which are routinely reviewed by production engineers for potential capacity issues. The performance data and reports allow us to predict as well as prevent capacity problems within the environment (see Figure 3).

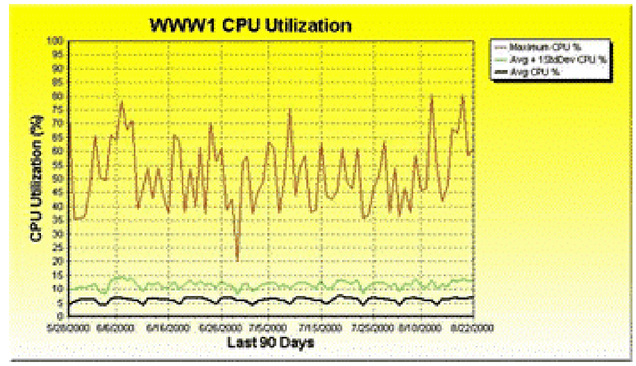


Figure 3: An example of reports used by e-Business to monitor capacity

In addition to monitoring system capacity Intel e-Business also uses a third-party agent and infrastructure to collect system configuration data. Knowing how a system is built and what applications are running on that system is an important component to managing and controlling a large e-Business site. E-Business systems are routinely scheduled to query their own hardware and software configurations and relay those data to a central repository (Figure 4). Once in the repository, engineers can query and review the dynamic data. This gives Intel e-Business the ability to identify, on a large scale, the systems that either need upgrades or configuration changes.

Intel IT has also developed a custom application called Metrios® that performs functionality testing. Metrios accesses e-Business systems in much the same way a user would and verifies system responses. It can run Active Server Page scripts and even dial out to test connectivity from the Internet. Metrios can be configured for automatic monitoring and will page out to notify staff of problems requiring immediate attention (see Figure 4). Once notified, Intel IT staff use Microsoft's Terminal Server* and the Intel® LANDesk® Server Manager 6.2 for remote system management. Staff members can dial in from their desks or homes to access and manage systems throughout the e-Business computing environment. These tools are essential components for maintaining the Intel e-Business infrastructure and enabling efficient control of widely distributed systems.

* Other brands and names are the property of their respective owners.

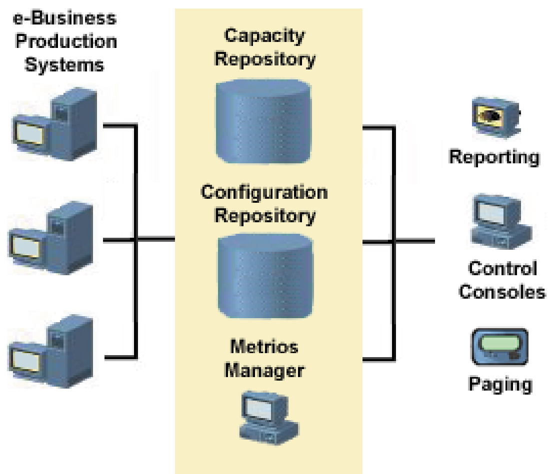


Figure 4: Comprehensive e-Business systems management obtained through an integrated infrastructure

FACILITIES AND INFRASTRUCTURE MANAGEMENT

The e-Business environment is becoming increasingly complex and in so many cases challenging Moore's Law. This growth is exponential in the speed of new applications being introduced and existing applications being upgraded. These requirements are being driven by the very intelligent end-user, who also happens to be the one developing the architecture. Along with asset management and systems capacity planning, facilities and infrastructure management was identified as a key contributor to planning Intel's e-Business success strategy.

Facilities and infrastructure management is defined as the physical space (data center and labs), the network infrastructure, and the power and cooling requirements to house all of the systems in a given space. These three major components are wholly dependent on each other in order to accommodate a single system. One cannot land a system in the data center or any of the pre-production/development labs without having all three components in place.

In summary, facilities and infrastructure planning is so tightly integrated with asset management and capacity planning in Intel's e-Business environment that it cannot be treated as a separate issue.

Challenges

The single most challenging problem with facilities and infrastructure management planning is how to plan for growth and how to decide how much growth to plan for based on cost and return on investment. It is a very peculiar and confusing state to be in. Do you plan for two, four, or six years? Can you even plan for any further than four years out? Based on trend analysis of system procurement and the change in systems, how much density do you plan for? What is your focus and how will this investment be returned? What are the success criteria of a data center or lab design? Once these questions have been answered, you can start the management process.

The problems with managing facilities and infrastructure are, in fact, in most cases the most costly and the most time consuming to react to and fix. We are trying to expand our power/cooling capacity in the data center to accommodate the amount of physical space and network capacity we currently have available. Power and cooling requirements seem very basic, but they can be the deciding factor in the installation of systems in a specific lab or data center. The design and planning of power and cooling capacity are essential and are very sensitive to our ever expanding e-Business infrastructure.

In order to understand the requirements of the pre-production labs and data center, we first prepared an historical trend analysis specific to the data center, production system growth, and pre-production system growth. We factored in the understanding that the growth rate of systems in our e-Business environment has been doubling every year, and the system platform physical vertical size has decreased to 1/3 or 1/2 of the size from one year ago. We also factored in our intention to make sure we would reduce the complexity of application deployment (support, launches), support processes, network infrastructure (less reliance on WAN, MAN, routers), instrumentation, metrics, analyses, security administration, and capacity management. In addition to reducing complexity we also wanted to avoid indirect costs of load balancing and network infrastructure optimization, increased support costs, and increased remote administration.

An overall facility and infrastructure roadmap was designed with a consistent physical capacity, network capacity, and power/cooling capacity architecture and framework. Each of the pre-production labs are now outfitted with cabinets and racks to accommodate all types of systems – tower or rack mounts – as well as with the network and power/cooling capacities which are scalable at a equal rate to each other.

The data center power/cooling capacity has been retrofitted to accommodate the physical capacity required by e-Business production.

RESULTS

The success criteria of facilities and infrastructure management is when the design of the lab or data centers factors in the correct ratio of physical space to network capacity to power and cooling capacity, and all of these are scalable at an even level. For example, the capacity remaining for network, physical space, and power/cooling is identical. In a facility design, you do not want to be in a position where you have a large amount of one component and none remaining of another component.

The design of the Intel e-Business data center and pre-production labs is complete. The pre-production labs are in operation and are using the new facilities and infrastructure strategy. The data center will soon be online with the new infrastructure strategy.

With a long-term business strategy in place and a complete understanding of the components of the facility and infrastructure, Intel e-Business will be able to scale the data center and all of its pre-production labs with zero interference to business.

The importance of proactive facilities management and the understanding of infrastructure scalability has been recognized and will always be an integral part of

e-Business strategy planning.

CONCLUSION

As e-Business strives to provide the most successful and available Internet presence to the world, behind the scenes are the operational processes and tools enabling it to do just that. Integrated, scalable, and very accurate asset management and capacity planning programs are essential. Without these programs in place, there is no platform on which to build the business strategy. The initiation and development of an asset management solution and capacity planning program will pay off in the most rewarding ways. With these programs in place you will be able to successfully reduce the complexity of the e-Business environment with consistent processes and tools. Resources will be used effectively (both assets and personnel); system and application performance and availability will increase; and consequently, time to market will decrease. The result is an overall reduction in cost for Intel.

ACKNOWLEDGMENTS

The contributions of Jeremy Murtishaw, an e-Business Lead Internet Gateway Engineer whose knowledge and experience in Data Center and Lab Network and Facilities Infrastructure Design, are gratefully appreciated.

AUTHORS' BIOGRAPHIES

Brad Coyne is an e-Business Architect with Intel Corporation. He graduated in 1998 with a B.S. degree in Business and an M.S. degree in Information Sciences from Ball State University in Indiana. He has been with Intel Corporation for almost two years now. Since joining Intel his engineering work has focused on server management and Internet development and architecture. This focus includes the development of new applications and methods for managing an IT organization's assets. His e-mail is bradley.r.coyne@intel.com.

Sonja K. Sandeen is an e-Business Logistics Analyst with Intel Corporation**.** She obtained a B.S. degree in Business Operations from the DeVry Institute of Technology. Sonja provides e-Business Integration Engineering with a processdriven program for all of the operational facets of a unit. Her primary focus is on asset management space followed by capacity planning of the CPS Data Center and six pre-production labs. She enables process improvements across all of the e-Business environments to ensure maximum efficiency through integration of e-Business data tracking tools and processes. Her e-mail is sonja.k.sandeen@intel.com

Intel® e-Business Engineering Release Management and Application Landing

Alan Hodgson, IT e-Business Integration Engineering, Intel Corp.

Index words: release, environments, engineering, prioritization, testing, milestones, automation

ABSTRACT

To meet the rapid growth in the number of e-Business applications to be deployed and the demand for increasing the frequency for application releases, Intel's e-Business release cycle has gone through many evolutions and has seen many improvements. This paper examines these changes from an IT e-Business Integration Engineering perspective. The e-Business release cycle may range from 6 weeks to 20 weeks depending on the functional and technical complexity of the release. A release may include up to 6 applications and in general, each application releases 4 or more times a year. There are over 65 core e-Business external-facing applications.

In this paper, we focus mostly on Business-to-Business (B2B) direct sales and marketing, the most active e-Business area, and we deal only with the e-Business Integration Engineering group's role in e-Business release planning and management.

INTRODUCTION

Since 1998, the Intel® IT e-Business Integration Engineering organization has grown from a small group of 25 engineers supporting 50 servers running a handful of external-facing Internet applications to 80+ engineers supporting approximately 850 servers running 65 external-facing Internet applications. This server farm is split between pre-production, which in general covers development, QA test, and stress test servers; and production, which includes production, failover, and production support servers. The challenge has been to provide on-time, reliable engineering activities to support the rapidly increasing number of applications being released in the external-facing, Intel® e-Business space and to keep pace with the increasing application release frequency.

The Intel IT e-Business Integration Engineering group has progressed, out ofnecessity, from adhoc releases requested with minimal lead time by the application groups to a very close partnership with the business application groups that includes early engagement and notification of release priorities and plans. Out of this partnership has evolved a comprehensive release process that incorporates planning and prioritization, resource and environment management, engineering management and release management.

Intel is dedicated to delivering new and improved functionality to its customers in a timeframe that keeps Intel and its customers competitive in the e-Business market place. To ensure reliability and speed in the release process, Intel's IT e-Business Integration Engineering group is in the process of automating many of the engineering build activities. This has reduced environment build times by up to 50% and increased accuracy and reliability. Automation is continuing with all aspects of instance designs and code migrations.

The engineering activities vary with the technical complexity of the release, but in general include instance design activities, server build activities, stress testing, day 1 testing (a rehearsal of the deployment activities), and deployment of the application into production. A complex release may also include purchasing and assembling new hardware and possibly the development and implementation of new reference designs. These engineering activities are now closely managed and aligned with the application development lifecycle activities to ensure that development, testing, and deployment milestones can be met. Daily management of release milestones has become a necessity for meeting release deadlines.

This paper covers a broad range of issues and activities relating to the management of the release and deployment of an external-facing (where Internet communication passes through a firewall) e-Business application at Intel. It is written from an e-Business integration engineering perspective and looks at the Intel e-Business release process with respect to how the e-Business Integration Engineering and the e-Business Application Development teams handle the release process for over 60 applications a year.

Intel's e-Business applications fall into one of the following business areas:

- B2B Sales and Marketing Group (SMG)
- B2B Indirect Channel
- B2C (Business to Consumer)
- B2C Core Services (e.g., Search, Registration, etc.)

- B2B Supplier
- Other non-mission-critical e-Business applications

In most cases this paper refers to the release process for the most active and mission-critical e-Business area, Intel's B2B Sales and Marketing Group.

The Engagement Process

For new applications and major enhancements to existing applications there are two phases of customer engagement with the e-Business Integration Engineering group. The first phase, shown in Figure 1, deals with finding strategic responses to business initiatives and involves participation in the Architecture Design Group (ADG). In this phase, the e-Business Integration Engineering group provides expertise in infrastructure and architecture to help find the best strategic response. This may include piloting and validating a recommendation before sanctioning its acceptance.

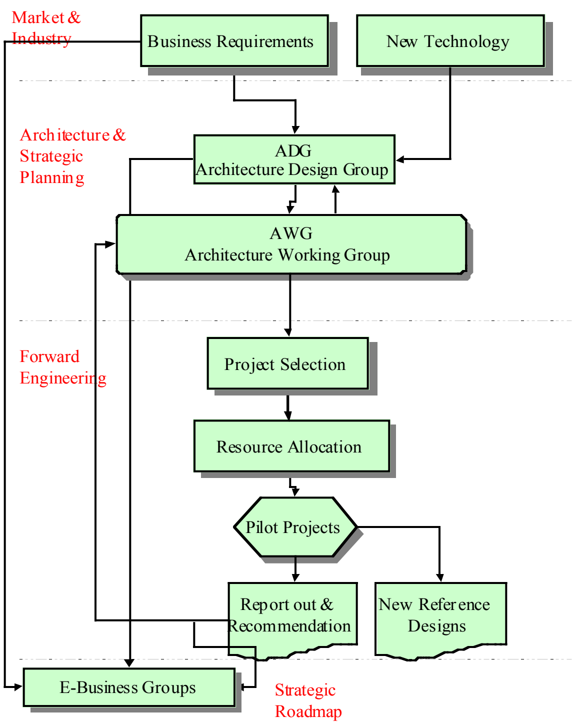


Figure 1: Initiative study and strategic architecture design

The second phase, shown in Figure 2, occurs when a project has been funded and is ready to be prioritized in the release process. In this phase, the technical complexity of the project is assessed, resources are assigned, and there is participation in the design, development, and deployment of the application.

This paper focuses on the second phase of the engagement process dealing with e-Business release planning and management.

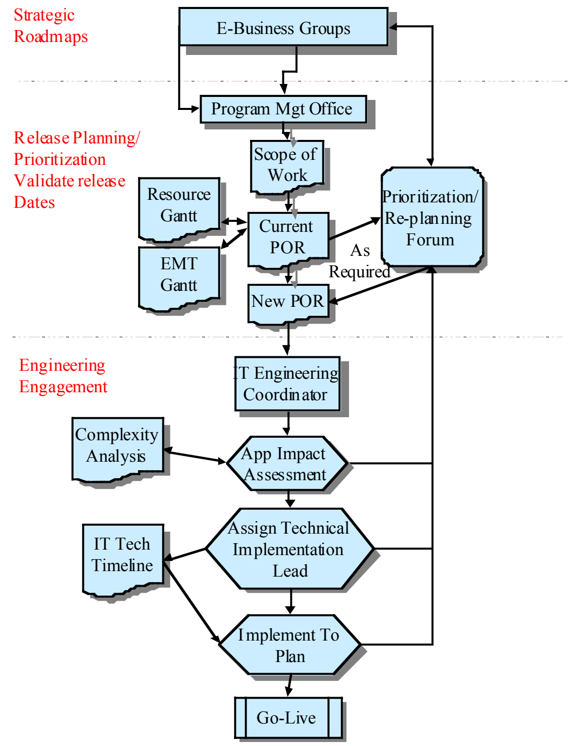


Figure 2: Release planning/management

CHALLENGES

As the number of e-Business applications increased and the need to release more functionality more frequently, there was an ever-increasing demand for hardware and engineering resources. This demand often outpaced the hardware and resources available. Environments and servers were being shared among applications and needed to be constantly rebuilt for the next release. Releases would follow so closely upon one another that a missed release date could very well have an impact on the date of the next release. Engagement with the IT e-Business integration engineers would happen towards the end of the application design phase leaving very little time for engineering to get acquainted with, and prepare for, the new technology requirements. In many cases, new hardware needed to be purchased and new reference designs developed. In addition, there was little or no business prioritization of the applications and associated releases. All this resulted in unrealistic expectations from the users that a release could always be resourced and delivered on time. From an engineering standpoint this was a no win situation and resulted in overloaded resources and missed schedules. This paper addresses the release planning and management processes that were developed and implemented to change this scenario yet support this fast-paced dynamic environment.

ENGINEERING RESPONSIBILITIES

First, let's take a look at the Intel IT e-Business Integration Engineering organization and its role in the release process. Support for our release cycle environments requires a team of engineers who can plan, design, build, test, and maintain the environments. These engineering activities are in the critical path of the release and are dependent on certain development activities being completed on time. Likewise, the QA test teams are dependent on the e-Business integration engineering activities being completed on time. The Intel e-Business Integration Engineering group has categorized these activities as follows:

- Design Engineering
- Factory Engineering
- Pre-Production Engineering
- Production Engineering

Design Engineering includes those activities relating to adding or changing infrastructure designs. These design additions or changes may be as a result of new software, new hardware, or new releases of operating systems and databases. Design engineering may include designing new infrastructure architecture, prototyping and testing the changes, and ultimately developing reference designs that can be used to consistently set up and install the new hardware, software, or release. These reference designs are used by other engineers to build servers for new application releases.

Factory Engineering includes those activities relating to ensuring a successful deployment of the application release from an engineering perspective. These activities include developing and managing the engineering project plan and designing and documenting the builds for all servers in the environment. These design documents are called Instance Designs (often referred to as branding documents), and they provide the complete specification for building and configuring a server. They include, in part, reference designs for the base builds of the OS and databases, but in addition include directory structures, application installation and configuration instructions, and Common Object Model (COM) specifications. Factory engineering requires the engineer to be closely involved with the application development team to understand the new application's technical designs. The engineer should be a partner in the design solutions. A factory engineer will, in most cases, use the build of the proto environment as an engineering prototyping opportunity to test out and develop the instance design documentation. This documentation will ultimately be passed on to the pre-production engineer and the production engineer to build and maintain the testing and production servers.

Pre-Production Engineering includes those activities relating to the building and maintenance of servers in the pre-production environments. These engineers are expected to build from the Instance Design documentation to ensure consistent and reliable server builds. In the release cycle, these engineers are most impacted by any slippage in application design and development activities, since they are still expected to deliver test environments to the QA and Stress test teams on time.

Production Engineering includes those activities relating to building the production servers, deploying an application release into production, and providing adequate production support for the application infrastructure. Production engineering is also a primary recipient of the Instance Designs developed by the factory engineer and validated by the pre-production engineer. A production engineer is also a primary participant in building the stress and day 1 environments and performing the day 1 execution. The day 1 execution is in effect a dress rehearsal of the deployment of a release into production and usually takes place the week prior to production deployment. There is an additional burden on the production engineer to voice concern if the application does not meet the entrance criteria for production stability and performance. Once a deployment has taken place the production engineer becomes responsible for on-call support, production environment maintenance, data center management, and troubleshooting of production issues.

Server Name	PROD	Stress 1	Stress 2	Stress 3	Stress 4	Stress 5	Stress 6	QA	PROTO
CYAN	S26	NA	S26	S26	S26	S26	S26	S26-RM	S26-RM
TEAL	S26	NA	S26	S26	S26	S26	S26	HP	S26-2
PLUM	S26	NA	S26	S26	S26	S26	S26	HP	S26-RM
ROSE	S26	NA	S26	S26	S26	S26	S26	HP	S26-RM
IVORY	S26	NA	S26	S26	S26	S26	S26	HP	S26
DIAMOND	S26	NA	S26	S26	S26	S26	S26	HP2	S26
EMERALD	S26	NA	S26	S26	S26	S26	S26	HP	S26-2
OCEANIC	S26	NA	S26	-	PC-360	S26	S26	S26	PC-350
CPSBATCH01	S26	NA	PC360	PC360	PC-360	S26	S26	HP	S26-2
CPSAMS01	4300	NA	4300	4300	4300	S50	S50	HP2	S26
JFSCODR	S26	NA	S26						
JFSIBL01-ORCL	4300	NA	S50-1	4300	S50	S50	S50	S26-RM	S26-RM
SW	S26	NA	S26	S26	S26	S26	S26		
ECLIPSE	S50	NA	S50	S50	S50			S26-RM	S26-RM
CAMELOT	S50	NA	S50	S50	S50			S26/S50	S26-RM
ECITRDB	S26	NA			S26			Consolidated on QADB	
PYRITE	S26	NA			S26			S26	
KALPLANA	S26	NA			S26			S26	
CELTIC	S26	NA	S26-RM	S26-RM	S26-RM			S26	S26-RM
SORRENTO	S26	NA							
CPSUSI	S26	NA	S26	S26	S26				
MIDAS	4400 (2 4300 (1	NA	4300					4300	
MIDASD2W	S26	NA	S26-RM					S26-RM	S26-RM
IBPDB1	4300	NA						S50	
GREEN	S26	NA						S26-RM	S26-RM

KEY: Landed/Online; Received/Not installed; To be Ordered; Order Approved: ESD: Estimated Shipping Date; Requires Upgrade

Figure 3: Sample pre-production environment server map

RELEASE ENVIRONMENTS

Each functionally dependent set of applications is developed, tested, and deployed on an infrastructure of servers that has been designed and configured to support the various applications. This infrastructure typically consists of groups of servers forming environments that are dedicated to phases of the release cycle.

- A development environment consists of a set of development servers used by developers to develop and test their applications.
- A proto environment consists of servers used to prototype or test out the designs.
- The pre-production QA environment consists of servers used by an independent QA test team to perform functional and integrated testing.
- The pre-production stress testing and day 1 testing environment is the most dynamic environment. It actually comprises up to five environments called stations, which are built on an as-needed basis to stress test applications within a release. An environment station may be used for stress testing Web Order Management one week and rebuilt to stress test a channel application the next week.
- The production servers host the application in production and typically include a redundant set of fail-over servers. In addition there may also be a production support environment where production bug fixes can be made and tested without impacting a new release that is being tested in the pre-production environment.

The complete set of phased environments, including production, is called a pipe. A pipe is, in most cases, specific and dedicated to an application area,(e.g., B2B sales and marketing. A phased environment typically consists of one or more web, database, and application servers. The numbers will vary depending on how close the phase is to production. The QA test environments will be more representative of the production architecture than the development environment and will include back-end system environments that make up the integration architecture. At Intel, the e-Business, B2B Direct SMG pipe requires between 20 to 26 servers in each of the test environments and around 50 servers in production, including failover. The pipe consists of around 140 servers that need to be maintained and supported for a release. See Figure 3.

The new or changed application is moved from environment to environment as development is completed and testing begins. At each stage, entrance criteria need to be met, as shown in Figure 4, before the code can be migrated to the next environment.

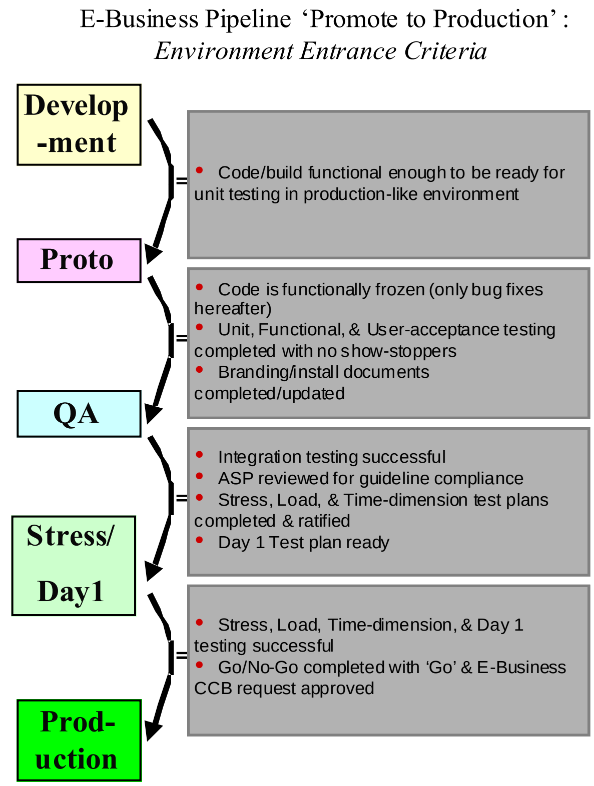


Figure 4: Environment migration and exit criteria

The Release Cycle

The release cycle for an e-Business application consists of the standard phases of planning, requirements, design, development, testing, and deployment. However, unlike the large back-end systems that often span years of development, an e-Business release takes anywhere from six months to six weeks depending on its functional and technical complexity.

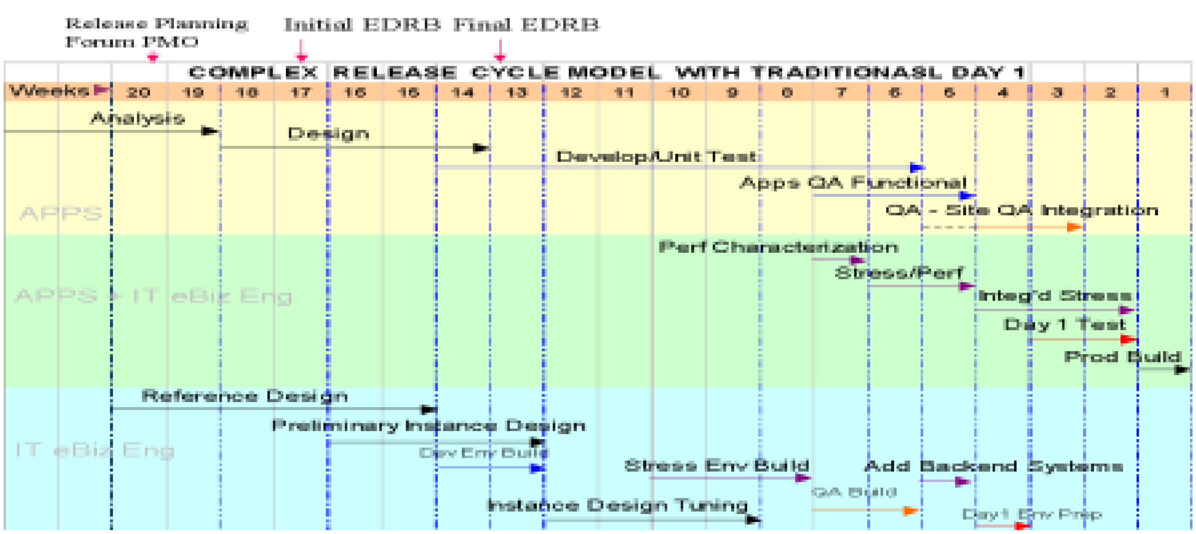


Figure 5: Complex release model

The Complex Release Model shown in Figure 5 reflects significant infrastructure changes that would include new servers and associated reference and instance designs.

A Medium Complex Release Model would not necessarily require new equipment and reference designs, but would require significant changes to instance designs with corresponding build and test activities.

The Simple Release Model shown in Figure 6 may require significant testing of functionality but would require very little engineering. In this model, the environments stay dedicated to the application and do not need to be rebuilt or changed.

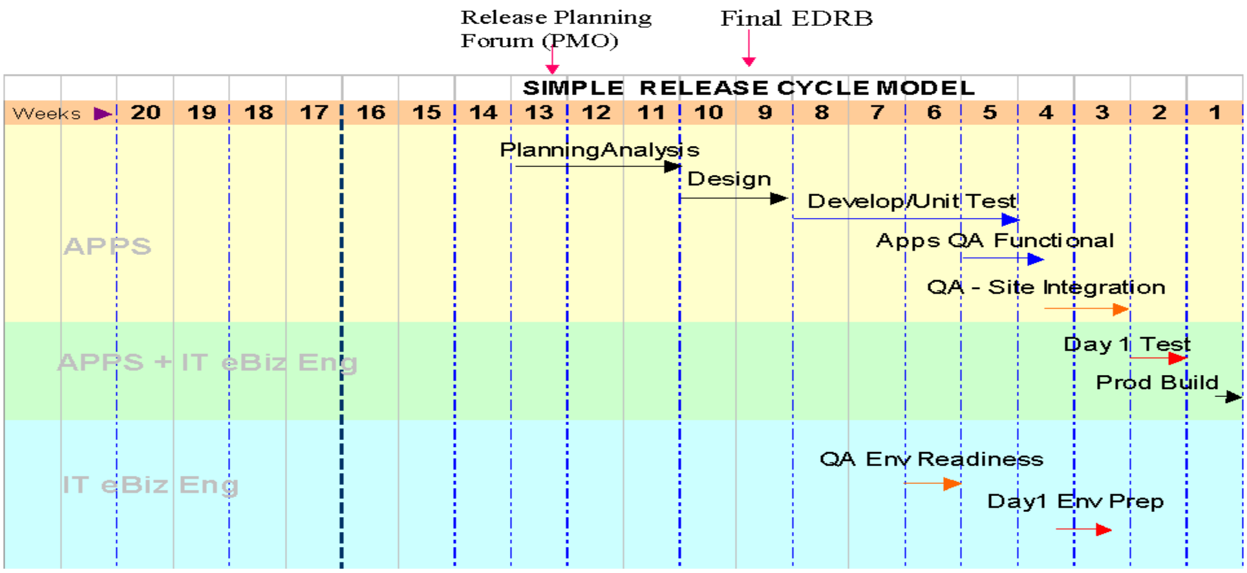


Figure 6: Simple release model

The Release Management Process

In the early days of e-Business development at Intel, the planning horizon for releases was no more than six months, and releases would occur somewhat sequentially within each business group. A release could consist of up to six applications. The dependencies between those applications were shared functionality, test environments, shared resources, or just a shared release date. This approach did have some success, but success was always dependent on the weakest application. If an application failed to pass testing criteria, the entire release might have been pushed out. This in turn had an impact on future releases that were dependent on the same resources and environments. This constant flow of critical path activity was the most stressful for the environment engineers and the QA test teams.

Figure 7 illustrates the four major processes that need to take place to manage a release.

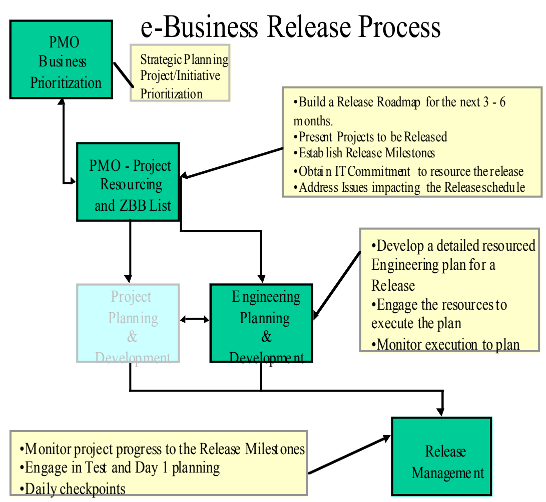


Figure 7: Major release management processes

Business Prioritization and Resourcing

In our largest e-Business area of B2B, the business organization has established a Program Management Office (PMO). The PMO is chartered with the responsibility of establishing a forum for business owners to submit release requests and join with other business owners to prioritize all requests relative to Intel's B2B e-Business priorities. Other participants in the forum include e-Business Integration Engineering management and QA/Test Management. The process is simple. The projects are assigned tentative release dates and priorities by the business owners. Each resource manager assigns resources to the projects until all resources are allocated. In the case of e-Business integration engineering, it is also necessary to determine if the shared test environments are available as required by the prioritization and release date. The business owners then determine if the projects left unresourced can be postponed. If they can, then no priority changes are made. If they cannot, then further prioritization is done or funding is allocated for additional resources or environments until the correct business prioritization is achieved. This process has brought visibility to the ratio between release demand and resource availability to implement a release and has resulted in all the key players working from the same prioritized list.

Engineering Planning and Development

From an engineering perspective, there is a lot to do in a very short space of time. We may identify a set of resources to work on the release in the PMO forum, but at that time we don't know much about the technical requirements. In the past, the application teams presented their technical requirements to the e-Business integration engineers at the end of their design phase, at the e-Business Design Review Board forum (EDRB). This late engagement has proved to be a major issue for complex releases where new hardware needs to be purchased and possibly new reference designs developed. Today, we assign an Integration Technical Lead (ITL) to the release after the PMO forum has given the green light for the release to proceed. It is the role of the ITL to partner with the application team to get an understanding of the complexity of the application and then to put together an engineering plan to meet the requirements. The ITL is also expected to be a partner with the application development team to develop the technical design and jointly present it at the EDRB.

Daily Release Management Meetings

In the past, the daily release management meetings focused on issues surrounding the release at a particular point in time. This served a purpose and allowed bottlenecks and issues to get addressed very quickly. However, it did not lend itself to looking at the big picture or even looking a week or more ahead. Today, we still have a daily meeting but the release is managed through the implementation of milestones. These milestones,(see Figure 8 below) reflect major dependent events that need to be completed on time to ensure that the release stays on track.

Daily monitoring of milestones has proved to be quite successful and provides advanced warnings of potential showstoppers. A missed milestone results in more attention being focused on the issues rather than an immediate halt to the application or release. A continuation of missed milestones for an application may result in that application being pulled from the release.

Application Name	High Level Design	Reference Design	Detailed DesignRvw	Proto Env Ready	Prelim Inst Design	StartFunct Test	eARL Ready	Start Char'n Test	QA Env Ready	Funct Code Freeze	PROTO Exit	Char'n Testing	StartInteg Testing	Inst Design Freeze	Day 1 Env Ready	QA Exit	Integ Stress Env	StartInteg Testing	Int Stress Complete	Day 1 Plan	Start Day 1	Management Update	Day 1 Complete	Go/NoGo
Kihei	w15	w26	w20	w30 7/17	w30	w30 7/17	w34 8/14	w34 8/14	w34 8/14	w34 8/14	w34 8/14	w36 8/28	w34 8/14	w34 8/14	w36 8/28	w36 8/28	w33 8/7	w34 8/14	w37 9/5	w37 9/5	w37 9/5	w38 9/14	w38 9/13	w38 9/13
IBL db																								
Vulcan 2San																								
Larry+ InfoDesk															w38					w38 9/12	w38 9/12	w40	w39	w40
AM/FDBL Content Publishing Role															w38					w38 9/12	w38 9/12	w40	w39	w40

Figure 8: Milestone tracking for a release

INCREASING THE RELEASE FREQUENCY

Engineering and Business Partnership

The e-Business world is constantly changing as customers require new functionality. The industry is constantly improving on technology and functionality, and on server technology; and operating systems and databases are issuing major releases almost every year. This is not a time for bureaucracy and process for process sake. It is a time for flexibility, compromise, and speed. These are most effectively accomplished by an alliance, or a partnership, with all the groups participating in the release cycle. The goal is to jointly deliver the release on time and not to demonstrate the shortcomings of the other groups' deliverables. However, the realistic requirements of each group still need to be respected and delivered with flawless execution.

In the Intel e-Business world, the partnership that exists today between the business organization and the IT e-Business Integration Engineering group is being enhanced with a much closer resource alignment to the business groups. This will provide more focused, dedicated environments and resources for each business group.

Automation

Automation is beginning to play a significant role in speeding up the release process. In the e-Business integration engineering world, we have started to automate server build scripts that can dramatically reduce the time and the amount of engineering resources it takes to build a server. Today, we have all of our OS and database base builds completed and in operation. In the future, we will be automating as many of the instance design scripts as possible.

In addition, we are also bringing in testing tools that have been successful with the back-end systems for test script automation and execution for both functional and stress/performance testing.

Pipeline Server Management

Dedicated pipelines of servers for each business group will reduce server conflicts between business group releases, but will most likely double and triple the number of servers over the next two to four years. Managing these large server farms is going to be a major challenge, ensuring redundancy for failover and reliability rates in the 99.6% – 99.8% range. To complicate the picture even more, Intel is planning a distributed data center architecture across North America, Asia, and Europe. This architecture is seen as a means to improve connectivity and performance in Asian and European countries. The release process may ultimately include multiple releases in different time zones.

MANAGING E-BUSINESS GROWTH AND CHANGE

Resource Constraints

One of the most difficult challenges facing the e-Business Integration Engineering group today is hiring enough experienced people who can hit the ground running when they join Intel. The e-Business Integration Engineering group has grown from a 30-person organization two years ago to a 100-person organization today. Windows* NT and other operating systems, Oracle, and SQL engineers are the primary resources. The e-Business Integration Engineering group has developed boot camp classes for new hires and puts a high priority on training. We are also looking at opportunities for outsourcing the more standardized tasks to free up our experienced employees to work on the more creative design and engineering tasks.

Outsourcing

The key areas of opportunity for outsourcing that are being evaluated are application development and support, infrastructure engineering environment builds, production hosting, and first-, second-, and third-level support. Many of the outsourcing opportunities can be outsourced within Intel but there may be situations where it makes more sense to use external vendors.

Infrastructure Releases

Keeping the infrastructure current and consistent has always been a challenge in this fast moving ever-changing e-Business world. A year ago there were 75 different reference designs, many of which had minor service pack or version differences. Today, through a concerted effort to standardize and reduce the number of reference designs, the number is down to around 40 and is still going down. Scheduling an infrastructure release for upgrades to servers, operating systems, and databases has always been difficult because of the impact upgrades have on the business release schedule. To lessen the impact, infrastructure releases have been aligned with functional releases. This allows the infrastructure release to leverage the environment engineering resources and the QA test resources. This strategy has been successful, but it is sometimes viewed as an unnecessary overhead and involves a risk to a business release. As a result, Intel's e-Business engineering group is looking into having one infrastructure release a year. The business is supporting this idea because it means a more stable and reliable environment.

RESULTS

Over the last three months that these improved release management processes have been in place, there have been no release misses in the B2B space, and the IT e-Business Integration Engineering group has delivered as required. The early engagement both with the PMO and the ITL has caused teams to plan ahead and adequately resource the releases. Time will tell, however, with respect to increasing the number of releases and whether automation and outsourcing can add further value.

In the next six months there will be approximately 12 releases involving approximately 21 applications in the e-Business B2B business area. The majority of these applications have been prioritized and resourced. This is twice as many releases as in the same period last year.

DISCUSSION

The processes and improvements that have been addressed in this paper are bearing fruit, but are just the beginning of the change that needs to take place. The e-Business world is fast paced and ever changing and requires constant monitoring for new and better ways of doing business.

As we move forward with our releases it is clear that there are limits to the number of servers and people that can be thrown at the problems. The key success factor in delivering on schedule is staying closely partnered with our customers. We all have to make sure that our resources are expended on the highest priorities. For Intel, the PMO forum is a major success factor and needs to be expanded to include all e-Business groups.

It will be difficult for the business to stand by the decision to dedicate one quarter to an infrastructure release. However, without it we stand the chance of falling behind on significant improvements to our ever-growing environment.

Finally, outsourcing both internally and externally may turn out to be a key opportunity to increase the release frequency without incurring a major increase in resources and hardware.

CONCLUSION

Intel's executive management has set the goal to have 100% of its customer business Internet enabled by early 2001. To meet this goal, it has been necessary to develop and deploy e-Business applications in the shortest possible time and with an acceptable level of quality. To achieve this, we are changing the paradigm for building, testing, and deploying systems. Business-focused teams, partnerships between business and engineering groups; automation of environment builds and application testing; outsourcing of suitable functions; and a flexible but managed release management process where engineers and developers feel each others pain when milestones are missed, are all essential to our success.

ACKNOWLEDGMENTS

Thanks to Jeff Engman for his significant efforts in driving change into the release process and steadfastly following the process.

Thanks to David Roff who has helped drive the automation of the base builds.

Thanks also to Bruce Theurer who has driven change into the release process from the user side. Bruce is responsible for both the PMO and the daily release management meetings.

AUTHOR'S BIOGRAPHY

Alan Hodgson graduated from John Dalton College, Manchester, England in 1971 with an equivalent B.S. degree in mathematics, statistics, and computing. He is a 30-year veteran in information systems and has seen the good, the bad, and the ugly in release management over the years. He specialized for many years in testing and joined Intel 3 1/2 years ago as the Test and Integration Manager for a large SAP project. Alan is currently an Engineering Manager in the IT e-Business Integration Engineering group. His e-mail is alan.hodgson@intel.com.