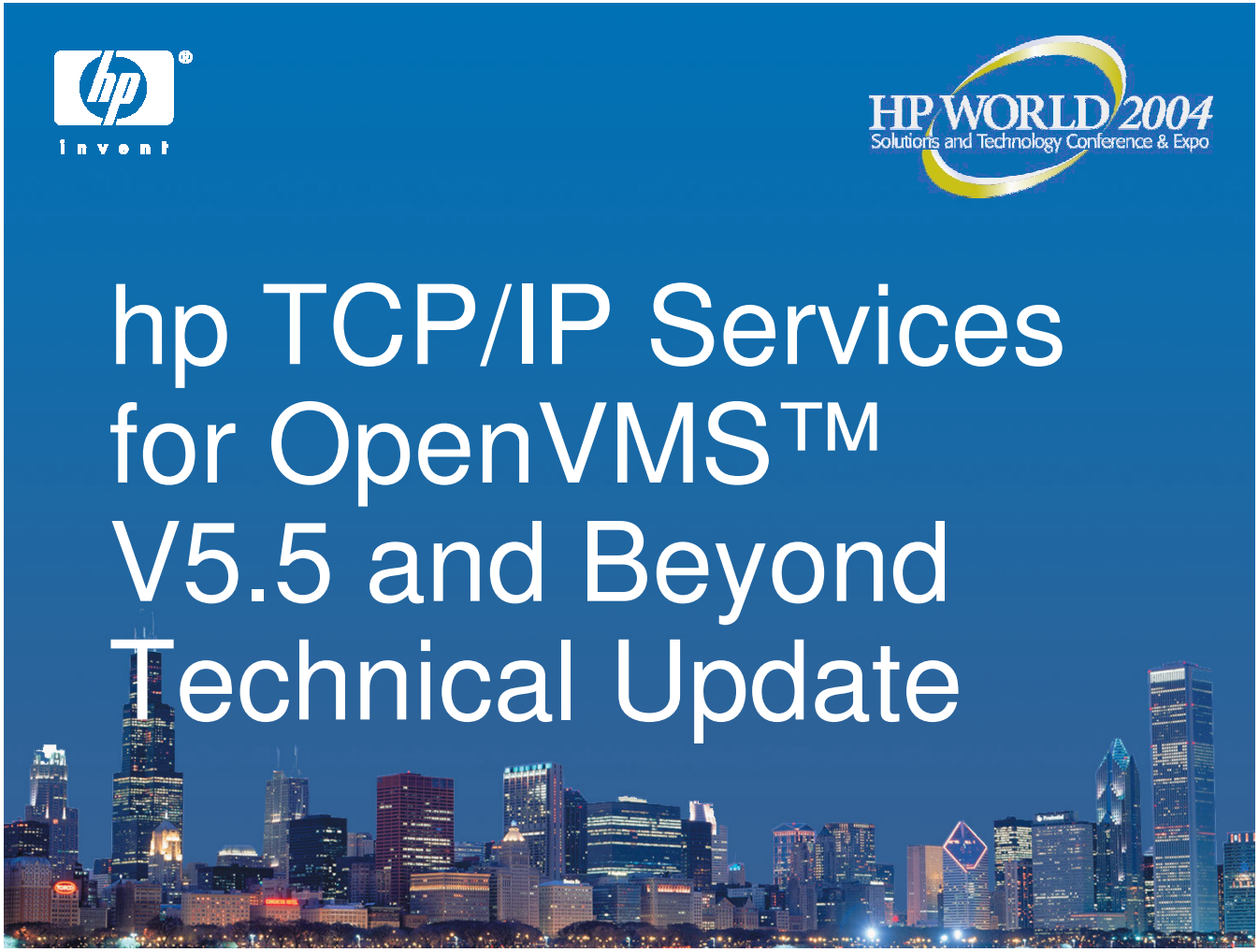


hp TCP/IP Services for OpenVMS™ V5.5 and Beyond Technical Update

Yanick Pouffary
Networks Technical Director

TCP/IP Services

- TCP/IP Services V5.4
  - FCS 02-DEC-2003
  - Alpha only release
- TCP/IP Services V5.5 for OpenVMS V8.2
  - FCS H2, CY 2004
  - Supported on both OpenVMS Alpha and OpenVMS Industry Standard 64 (I64) systems
    - NFS server unlikely on IA64
    - PPP unlikely on IA64

TCP/IP Services V5.4 Features List

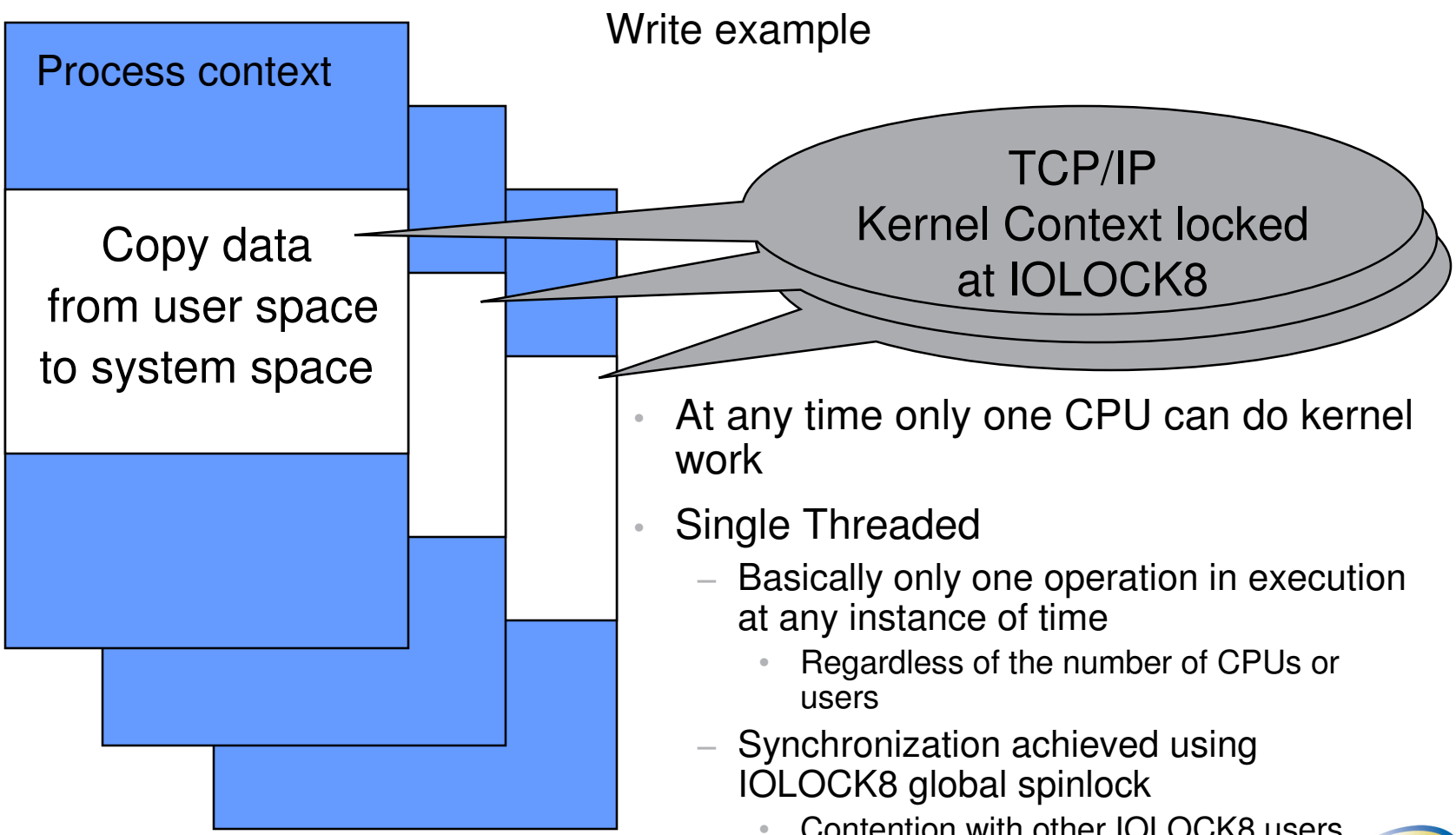
- failSAFE IP
  - IP address fail over capabilities within a host and/or a cluster
- Secure shell (SSH) v2 client and server
- Secure Socket Layer (SSL) for POP
- ISC BIND 9.2.1 Server
- tcpdump
  - Support to provide both dump analysis and packet capture
- IPv6 Software update
- Scalable Kernel to provide increased scalability for symmetric multiprocessing (SMP) (Require V7.3-2)
- Telnet Server performance and scaling enhancements
- NFS Server Performance enhancements
- INET driver performance enhancement
- Support for up to 32K BG devices
- Fast BG device creation and deletion

TCP/IP Services V5.5 Features List

- IPv6 Updates and Enhancements
  - failSAFE Support for IPv6
  - PWIP Driver Support for IPv6
- NFS Server Supports Case-Sensive file Lookups
- NFS Symbolic links support
- Support for NTP V4.2
- Support for TCPDUMP Version 3.7.2
- Update to SSH to V3.5.2

TCP/IP Kernel Options

Classic TCP/IP Kernel

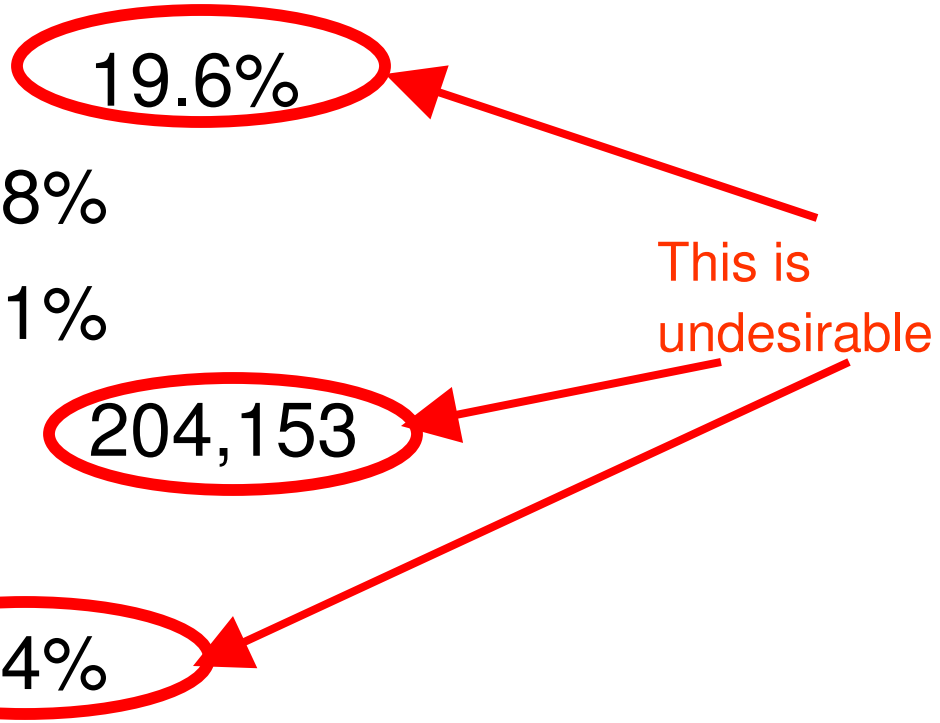


- At any time only one CPU can do kernel work
- Single Threaded
  - Basically only one operation in execution at any instance of time
    - Regardless of the number of CPUs or users
  - Synchronization achieved using IOLOCK8 global spinlock
    - Contention with other IOLOCK8 users (DECnet, LAN drivers, SCS, etc.)
  - Does not scale well in Multi-CPU system

Scaling Tests on GS1280 Classic Kernel

- Fork% system wide 19.6%
- Hold % all locks 54.8%
- Spin % all locks 24.1%
- Locks per second 204,153

- IOLOCK8 hold time 31.4%
- IOLOCK8 spintime 19.1%



Scalable Kernel

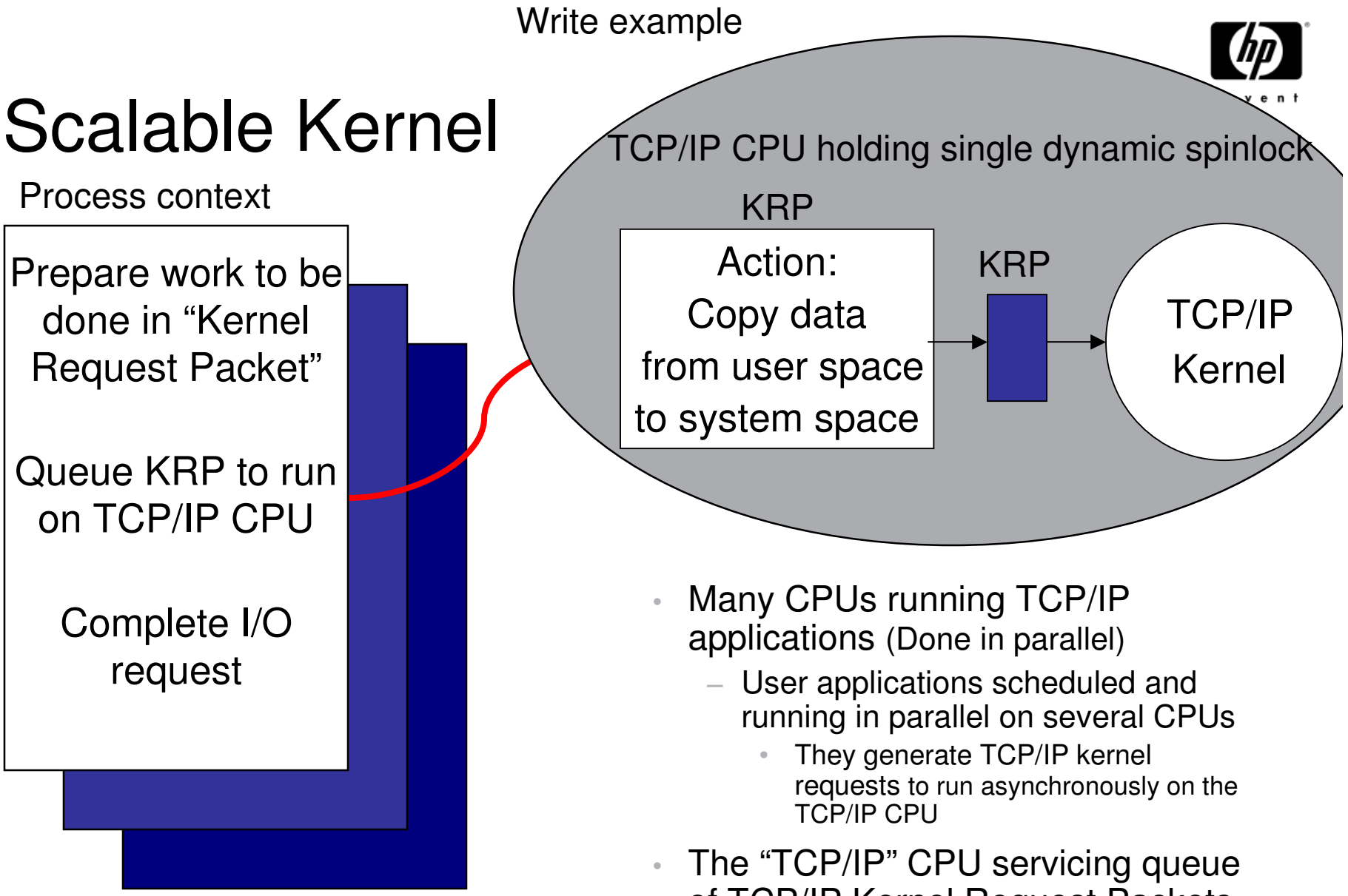
- Design Goals
  - Scale close to linearly over large number of CPUs
  - Reduce MPSYNCH to near zero
  - No longer use IOLOCK8

- Design Considerations
  - Synchronization of internal data must be done with little or no CPU contention (I.e. no MPSYNCH)
  - Transmits and Receives must proceed in parallel
    - Overwhelming majority of CPU cycles consumed in Transmit and Receive

Reduce CPU Contention

- Direct all processing requiring locking of internal data to a single designated “TCP/IP” CPU
  - removes CPU contention with other TCP/IP users
- Use dynamic spinlock to lock main internal database instead of IOLOCK8
  - removes CPU contention with other OpenVMS IOLOCK8 users
- Use several “mini” spinlocks to lock small subsets of database for small numbers of cycles
- For optimal performance LAN processing is done on a different CPU

Scalable Kernel



- Many CPUs running TCP/IP applications (Done in parallel)
  - User applications scheduled and running in parallel on several CPUs
    - They generate TCP/IP kernel requests to run asynchronously on the TCP/IP CPU
- The "TCP/IP" CPU servicing queue of TCP/IP Kernel Request Packets
  - Runs as a fork thread at IPL 8

Turns network I/O into asynchronous transactions

Scaling Limits

- As number of “user” CPUs grows the number of TCP/IP Kernel Requests requests grows
  - The amount of load each “user” CPU puts on the “TCP/IP” CPU is application dependent
  - Function of amount of application processing per TCP/IP Kernel Requests
- Adding more CPUs to the configuration of the system scales almost linearly until TCP/IP CPU approaches saturation

Scaling Tests on GS1280 Classic vs Scaling Kernel

• Fork% system wide	19.6%	22.7%
• Hold % all locks	54.8%	35.2%
• Spin % all locks	24.1%	4.4%
• Locks per second	204,153	162,033
• IOLOCK8 hold time	31.4%	Less than 5%
• IOLOCK8 spintime	19.1%	
• TCPIP lock hold time		13.4
• TCPIP lock spin time		0.0

This is good

Excellent Sign

Same amount of work with fewer locks per unit of work This is good

CPU is not busy This is good

TCP/IP Kernels Support

- "Classic" Kernel
  - Synchronizes using IOLOCK8
  - Similar to previous releases of TCP/IP prior to 5.3
  - Default in V5.4, Not present in V5.5
- "Scaling" Kernel
  - Highly parallel operation, No longer uses IOLOCK8
  - Requires V7.3-2 and above
    - In V5.4 to enable the scalable kernel, add to SYLOGICALS.COM before the command to start TCPIP$STARTUP.COM
      - $ DEFINE/SYSTEM/EXECUTIVE TCPIP$STARTUP_CPU_IMAGES "PERF=ALL"
  - Default in V5.5
    - Logical name TCPIP$STARTUP_CPU_IMAGES is now ignored

TCP/IP High Availability Solutions

- failSAFE IP
  - High Availability of IP addresses
  - Address Failover to alternate interfaces
  - IP Cluster Alias - Superseded by failSAFE IP
- Load Broker/Metric Server
  - High Availability of DNS Alias
  - DNS alias name dynamically updated with available addresses
- LAN Failover
  - High Availability of MAC address

What is failSAFE IP?

- Provides IP address redundancy
  - One instance of each IP address is active at any time
  - Other duplicate IP addresses are in standby mode
    - The OpenVMS distributed lock manager is used to ensure only one instance of an IP address is active across a cluster
    - Standby IP addresses may be configured on multiple interfaces within the same node or across a cluster
- Removes NIC as SPOF
  - Typical failures include NIC failure, disconnected or broken cable, or a dead port on the switch
- failSAFE service monitors the health of each interface and takes appropriate action upon failure or recovery of the interface

Configuring failSAFE IP

- To assign the same IP address to multiple interfaces
  - Use the TCPIP$CONFIG Core Environment menu to assign an IP address to multiple interfaces
    - Only one instance of the address is active, others are standby
  - Or use the ifconfig utility, which provides a greater degree of management control
- Enable the failSAFE IP service to monitor the health of interfaces
  - Enabled thru TCPIP$CONFIG Optional Components menu

failSAFE IP service

- Monitors the health of interfaces by periodically reading their “Bytes received” counter
- Logs all messages to TCPIP$FAILSAFE_RUN.LOG. Important events are additionally sent to OPCOM
- Generates traffic to help avoid phantom failures
- Invokes a customer written command procedure at the transitions
- Maintains static routes to ensure they are preserved after interface failure or recovery
- NOTE: If service is not enabled, then provides identical functionality to the IP Cluster Alias

Home Interfaces

- Creating IP addresses with home interfaces helps to maintain the spread of IP addresses across multiple interfaces
  - This is important for load-balancing and gaining higher aggregate throughput
- Upon home interface recovery after a failure, the addresses may return to their recovered home interface, thus maintaining the spread of addresses across the available interfaces
  - Note that an address will not migrate toward a home interface if it will result in dropping TCP/IP connections

failSAFE IP – Failure and Recovery

- Upon NIC failure
  - IP addresses and static routes on failed interface are removed
  - Standby IP address becomes active
    - IP addresses preferentially failover to an interface on the same node in an effort to maintain existing connections
    - If an address is not configured with a standby, then the address is removed from the failed interface until it recovers
  - Static routes on the failed interface are also removed and migrated to any interface where their network is reachable
- Upon NIC Recovery
  - IP addresses may be returned to the home NIC
    - Note: IP addresses will not return to a home interface if it means connections will be lost

Site-Specific Customization of failSAFE IP

- A user-defined procedure may be invoked during selected transitions of the failSAFE IP
- These transitions describe one of three events:
  - When the interface first appears to have stopped responding. This is the first warning that a problem may exist, but no action to failover IP addresses is taken yet
  - When an attempt to generate traffic on the interface fails. After the retry limit is reached, the interface is deemed dead, and IP addresses will be removed from the interface. Failover occurs
  - When the interface recovers
- The site-specific procedure is SYS$MANAGER:TCPIP$SYFAILSAFE.COM
  - May be defined by the logical name TCPIP$SYFAILSAFE

Static and Dynamic Routing (1)

- When an interface fails, failSAFE IP removes all addresses and static routes from the failed interface
- The static routes are reestablished on every interface where the route's network is reachable
- This may result in a static route being created on multiple interfaces and is most often observed with the default route.

Static and Dynamic Routing (2)

- Dynamic routing may need to be restarted to ensure the dynamic routing protocol remains current with changes in interface availability
- To do so use the TCPIP$SYFAILSAFE procedure restart the routing process
  - For example, for GATED:
    - $ TCPIP STOP ROUTING /GATED
    - $ TCPIP START ROUTING /GATED
    - Or use GATED scaninterval option to scan interfaces
    - Or force scanning thru $ TCPIP SET GATED/CHECK_INTERFACES

failSAFE IP - IPv6 support TCP/IP Services V5.5 (1)

- failSAFE IP has been upgraded to support IPv6
- Currently configuring standby IPv6 addresses requires manual intervention
  - This will be addressed in future
- Only link-local addresses need to be configured with standby addresses
  - To determine the IPv6 addresses assigned to each interface use $ ifconfig -a

failSAFE IP - IPv6 support TCP/IP Services V5.5 (1)

- Add to SYS$STARTUP:TCPIP$SYSTARTUP.COM with the commands below or executed interactively, at the DCL prompt, after TCP/IP has been started

```
$! IPv6 failSAFE Addresses
$!
$ ifconfig ie1 inet6 alias fe80::202:a5ff:fe60:a368
$ ifconfig ie0 inet6 alias fe80::202:a5ff:fe60:a369
```

- Then Restart failSAFE to pick up IPv6 address changes thru @SYS$STARTUP:TCPIP$FAILSAFE_SHUTDOWN and @SYS$STARTUP:TCPIP$FAILSAFE_STARTUP

TCP/IP High Availability Summary

- High availability requires careful planning and understanding of the failures that must be protected against
  - As a result, one or more high availability solutions may be required.
- failSAFE IP provides high availability of IP addresses for both incoming and outgoing new connections as well as existing traffic flow
  - IP Cluster Alias has been superseded by failSAFE IP
- DNS Alias with Load Broker and Metric Server provides high availability of a DNS Alias name and so benefits incoming connections only
- LAN Failover provides high availability of a hardware MAC address and benefits all LAN protocols

Comparing High Availability Technologies

	failSAFE IP	IP Cluster Alias	DNS Alias (Load Broker / Metric Server)	LAN Failover
Protects	All IP addresses	Single IP address designated as the cluster address	DNS Alias with list of most available IP addresses	MAC Address
Protocols	IP only	IP only	IP only	All LAN protocols
Scope	Interfaces within a node or cluster	Single interface per node in a cluster	DNS name lookup	Interfaces within a node
NIC	Independent of interface type	Independent of interface type	Not applicable	DE600 and DEGXA
Load Balancing	All interfaces active, balance outgoing connections, higher throughput	One interface in a cluster is assigned the cluster address, no load balancing	Load share inbound connections across DNS alias addresses	One interface in a node is active others are standby, no load balancing
Detects	Failure and recovery: interface, cable, switch, node	Node failure	Most available nodes	Failed interface, cable, and switch

SSHv2


invent


HP WORLD 2004
Solutions and Technology Conference & Expo

What Is SSH (Secure Shell)

- Application-level security solution
- SSH secures:
  - Terminal sessions, file copy, and remote command execution
  - Other protocols via port forwarding:
    - POP, FTP, X, SMTP, IMAP, even VPNs
- Consists of client, server, and support programs
- SSH is de-facto standard, turned internet standard
- Unix-oriented but available on many platforms

SSH Capabilities

- Features implemented:
  - Remote logins
  - File transfer (stream_lf, fixed 512)
  - Remote command execution
  - Key generation
  - Agent (key storage functionality tested)
  - Port forwarding (including ftp)
  - X forwarding (native mode)
  - Password, hostbased, and publickey client authentication
  - Multiple encryption algorithms
- SSH Components:
  - SSH client for login, remote command execution, forwarding (port, ftp and X11)
  - SSHD SSH2 server
  - SSH-KEYGEN key generation facility
  - SSH-AGENT holds keys in memory
  - SSH-ADD maintains keys inside agent
  - SSH-SIGNER digital key signer
  - SCP/SFTP file transfer clients
  - SFTP-SERVER2 file transfer server

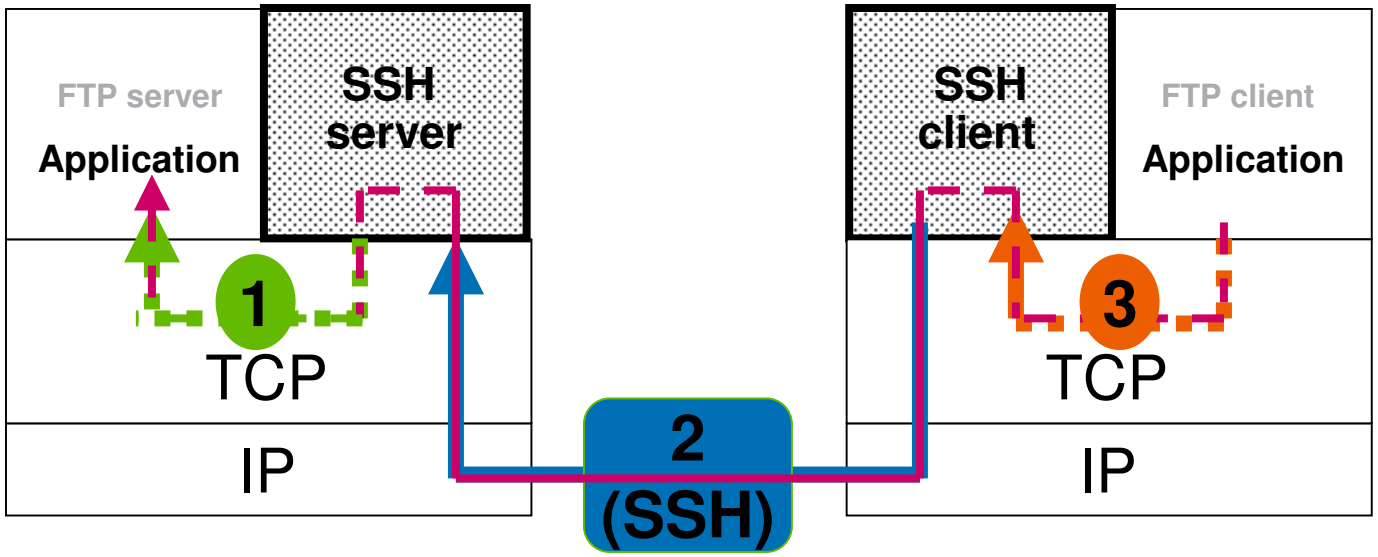
SSH in TCP/IP Release Details

- V5.4
  - Code base: SSH Communications Security, Inc.
  - Current support for SSH2 V2.4.1 on OpenVMS.
- V5.5
  - Update to SSH to V3.5.2
- Fully integrated in TCP/IP Services for OpenVMS
  - Configurable using TCPIP$CONFIG.
  - Use SSH through UNIX-style commands.
  - Compatible with OpenVMS auditing and access control.
  - Uses ASCII configuration files (same as UNIX)

SSH Authentication Setup

- Password Authentication (medium security)
  - Just turn it on
- Public-key authentication (most secure)
  - Generate, move public/private key files
  - Update identification and authorization files
  - Configure as needed (e.g., Authentication method, options)
- Hostbased authentication (least secure)
  - Update shosts.Equiv file and/or shosts
  - Copy server public key to client
  - Configure as needed (e.g., Authentication method, files)

SSH Port Forwarding (FTP Example)



1. SSH server: ACP enabled (no action required; Already running)
2. SSH client: (ssh <hostname> "-L ftp/2222:localhost:21")
3. Application: (ftp localhost 2222)

X11 Port Forwarding (Native Mode)

- To enable X forwarding
  - Client - set the value for forwardx11 to yes in config file ssh2_config
  - Server set the value for allowx11forwarding to yes in the config file sshd2_config
  - Alternatively, +x or –x command line options can be used enable and disable X forwarding on the client
- Requirements:
  - Native X11 forwarding uses Xauth for authentication
  - So Xauth should be installed on the system
  - Make sure you have the display variable for your display on the ssh client

OpenVMS SSH Startup

- Enable SSH via TCPIP$CONFIG
- Start SSH via TCPIP$CONFIG or server and client startup command
- Server startup and shutdown command
  - $ @SYS$MANAGER:TCPIP$SSH_STARTUP
  - $ @SYS$MANAGER:TCPIP$SSH_SHUTDOWN
- Client startup and shutdown command
  - $ @SYS$MANAGER:TCPIP$SSH_CLIENT_STARTUP
  - $ @SYS$MANAGER:TCPIP$SSH_CLIENT_SHUTDOWN

Auditing

- Update sysuaf, intrusion db
  - Sam behavior as telnet, set host
- Apply by default to password based authentication method only, can be configured to apply to others
- Application to remote command execution…
- Login failure but no auditing if no Account, expired, restrcited access, pwd_expired flag
- Auditing yes if
  - Invalid password, wrong keys, expired pw
  - Disuser or autologin, Intruder already

SSH Basic Troubleshooting

- Configuration and startup
  - SSH server and/or client
  - TCP/IP Services (restarted since upgrade?)
- Existence, contents and protections on:
  - Key files
  - Configuration files
    - Be sure values of AllowedAuthentications on client and server are compatible
- Status of user accounts (client and server)
  - $ mcr authorize
  - Access restrictions, expirations, …
  - File protections
  - Key files
- $ ASSIGN/SYSTEM 1 TCPIP$SSH_SERVER_DEBUG to get log level 99 when run usual way or
- Stop SSH server and run interactively with control over –d …. Capture log

SSH File Transfer Restrictions

- Stream_lf and fixed 512 (executables) supported (resulting files all stream_lf)
  - Not all variants of UNIX and OpenVMS directory paths supported
- Consider ssh ftp port forwarding as an alternative (see documentation)
- See Release notes, for known restrictions on file transfer, compatibility with other platforms, etc.

Other Features

Devices handling

- BG devices V5.4
  - Support for More Than 10,000 BG devices
    - Useful on very busy systems like web servers
    - Enabled thru sysconfig by setting ovms_unit_maximum greater than 9999 (< 32K)
  - Faster UCB creation and deletion
    - Support systems where large numbers of BG devices are continuously being created and deleted or where the number of BG devices has been increased above the 10,000 device unit limit
    - Enabled thru sysconfig by setting ovms_unit_fast_credel
      - This attribute can affect the amount of virtual memory used
- BG and TN devices V5.5
  - Use native VMS 8.2 fast UCB handling and creation up to 32,767 devices

TELNET/TELOGIN Driver Performance & Scalability Improvements

- Since V5.4 on Multi-CPU systems
  - No longer uses IOLOCK8
  - Added support for multiple concurrent I/O thru TN devices
- Reduced timer maintenance overhead
- Amount of overhead required for maintaining the TN devices has been reduced

NFS Performance Improvements (1)

- Support for a name cache for faster lookups (ODS-2/5)
  - Reduces the number of QIO operations required by the NFS server to look up files by name
  - Logical name establishes the size of the cache TCPIP$CFS_NAME_CACHE_SIZE
- Support for a file system directory cache (ODS-2/5)
  - Retains information about sequential files that will require record format conversion
  - Logical name establishes the size of the cache TCPIP$CFS_ODS_CACHE_SIZE
- These caches increase the virtual memory requirements of the NFS server

NFS Performance Improvements (2)

- Enable NFS server to take advantage of the directory and name caches by using the NFS attribute ovms_xqp_plus_enabled in SYSCONFIGTAB.DAT
  - This attribute is specified as a bit mask
- Internal performance improvements
  - New hashing algorithms, call reduction, code streamlining
  - Buffer alignment for faster moves
  - Increased number of NFS threads
- Ability for NFS to run on its own CPU on Multi-CPU system

V5.5 NFS Server Case-Sensitive Lookups

- The management ADD EXPORT command has two new options, CASE_BLIND and CASE_SENSITIVE
  - CASE_SENSITIVE enables UNIX-like case sensitivity for NFS server file lookups.
    - For example, NFS would preserve the case in the file names AaBBc.TXT and AABBC.TXT, regarding them as two different files
  - For UNIX clients lookup case-sensitivity is determined by the current ADD EXPORT/OPTION
  - For OpenVMS-to-OpenVMS mode
    - If running TCP/IP v5.5 or later, lookup case-sensitivity is determined by the OpenVMS DCL SET PROCESS/CASE_LOOKUP setting
    - If older version lookup case-sensitivity is determined by the setting of the ADD EXPORT/OPTIONS

V5.5NFS Symbolic Link Support

- Symbolic links are files that contain a link to another file or directory. When accessed the target file is accessed and deletion of the link has no effect on target file. Links can span disks and can span systems with NFS support
- Requires changes in CRTL, RMS and NFS
- NFS provides symbolic link support for both NFS client and server

V5.5 IPv6 Updates and Enhancements (1)

- IPv6 Configuration Enhancements and fixes
  - Can successfully configure 6to4 tunnels, all routes required for a 6to4 relay router, automatic tunnels, IPv6 over IPv6 manual tunnels, and manual routes
- ifconfig now documents how to manipulate IPv6 addresses
- IPv6 Neighbor Discovery process now Supports Dynamic Update Requests for ip6.arpa DNS Reverse Zone
  - If you still need to support delegations based on the ip6.int zone you can use DNAME to rename ip6.int
  - For more information, refer to Section 3.1.3,of the HP TCP/IP Services for OpenVMS Guide to IPv6

V5.5 IPv6 Updates and Enhancements (2)

- Several programming functions provided in earlier Early Adopter Kits (EAKs) were deprecated. These programming functions will no longer be supported after V5.5.
  - The following table lists the functions and their replacements:

Deprecated Function	Replacement Function
getipnodebyname	getaddrinfo
getipnodebyaddr	getnameinfo
freehostent	freeaddrinfo

V5.5 NTP V4.2

- Support for NTP V4.2
  - NTP V1 has been removed because of security vulnerabilities
- Still support authentication using symmetric key cryptography
  - Does not yet support public key cryptography
- Support for IPv6
  - Both IPv4 and IPv6 can be used at the same time
  - Please refer to the release notes for more details

Packet Tracing - TCPDUMP

- TCPDUMP
  - Provides native packet tracing and file based tracing
  - Native tracing in copyall mode
    - It only sees what the TCP/IP kernel sees
    - No promiscuous support (yet)
  - Boolean-based filter expression
    - This example shows tracing of both the ftp control session and ftp data session
    - $ tcpdump ip host lassie and (port 21 or port 20)

TCPDUMP versus TCPtrace

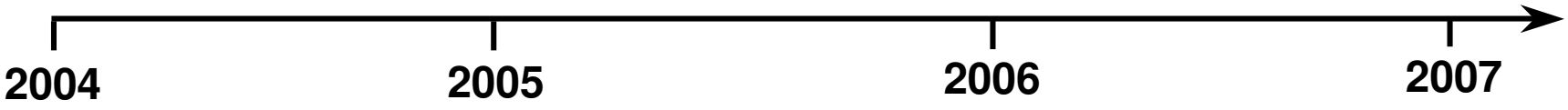
- TCPDUMP
  - standard UNIX packet trace analysis tool
  - binary file
    - compatible with Tru64 UNIX TCPDUMP
    - readable from other libpcap applications like Ethereal
- TCPtrace
  - traditional VMS style command interface with “dump” output
  - continued support for users familiar with this tool

V5.5 TCPDUMP and libpcap

- TCPDUMP has been upgraded to V3.7.2
- For more information about the changes in the new version of TCPDUMP, see the www.tcpdump.org web site
- libpcap API is provided for Early Adopters
  - An example program is included in the directory pointed to by the logical name TCPIP$LIBPCAP_EXAMPLES
  - The libpcap object library resides in the directory pointed to by the logical name TCPIP$LIBPCAP
    - The directory pointed to by the logical name SYS$SHARE contains an executable file

TCP/IP Services 5.next And beyond

TCP/IP Services for OpenVMS



TCP/IP V5.4 released
- **SSHv2 client functionality**
- **failSAFE IP (IP fail over)**
- **Scalable kernel**

TCP/IP V5.5 on Itanium® and Alpha - H2CY04 support of OpenVMS V8.2
- **NFS symbolic link support**
- **Libpcap library and TCPDUMP updates**
- **IPv6 configuration enhancements**

TCP/IP (Next)
(H2 2005)
Continued focus on performance & security
- **Updated SSH, SSL, Kerberos**
- **IPsec**
- **BIND 9 resolver**

IPSEC
EAK
H2 CY04

What is IPsec?

- Standards based IP-level solution to Security
- IPsec secures everything above IP Layer
- Provides:
  - ESP (Encapsulated Security Payload)
  - AH (Authentication Header)
  - IKE (Internet Key Exchange)
- Security policy dictates what is encrypted and which algorithms are available during IKE dialog
- When selected, can protect every packet
- Works for both for IPv4 and IPv6

IPsec Components

- Complex set of protocols, mechanisms, tools:
  - Engine: processes incoming and outgoing packets in real time
  - Interceptor: interface to the engine
  - Policy Manager: maintains a security policy database
- Applications:
  - Digital Certificate utilities
  - Cryptographic utilities
  - LDAP utilities
- ISKAMP/IKE
  - Security Association and Key Management

IPsec Implementation Plan

- Developed by SSH Communication Security, Inc.
- Encryption obtained using OpenVMS CDSA

IPsec vs. Secure Application Layer

- SSH, SSL/TLS:
  - Built into each application
  - Controlled by the application
  - Only applies end-to-end
- IPsec:
  - Applies to all network traffic
  - Controlled by the system administrator
  - Part of network infrastructure (VPNs)

TCP/IP Services for OpenVMS Pointers and Contacts

- HP OpenVMS Network Transports Home Page:
  - http://www.hp.com/products/OpenVMS
- Documentation Online:
  http://h71000.www7.hp.com/doc/tcpip.html
  Software Products Library
- OpenVMS V7.3-2 Documentation CD
- FailSAFE - OpenVMS Technical journal V2
- Scalable Kernel - OpenVMS Technical journal V4
- Contacts:
  - Product Management
    Lawrence.Woodcome@hp.com

Thanks for Listening!
....any questions?

General Feedback


hp
invent