HP-UX 11.X Technology and Beyond

DAL

John Fenwick
Hewlett-Packard
19447 Pruneridge Avenue
Cupertino, CA 95014 USA
408-447-4976
FAX 408-447-4364
John_Fenwick@hp.com

HP-UX 11.X Technology and Beyond Program Outline

DAL

- Introductions and Overview
- "Largeness" Extensions to HP-UX:
  - Filesystems, Files, UIDs, File Descriptors, Process Space
- 64-Bit HP-UX
- Developing for 64-bits
- Kernel Threads
- Additional technologies:
  - DLKM, Large Pages, Ignite-UX, System Recovery, others
- Networking enhancements
- HP-UX 11.X and Extension Pack releases
- Review

HP-UX 11.X Technology Review Program Outline

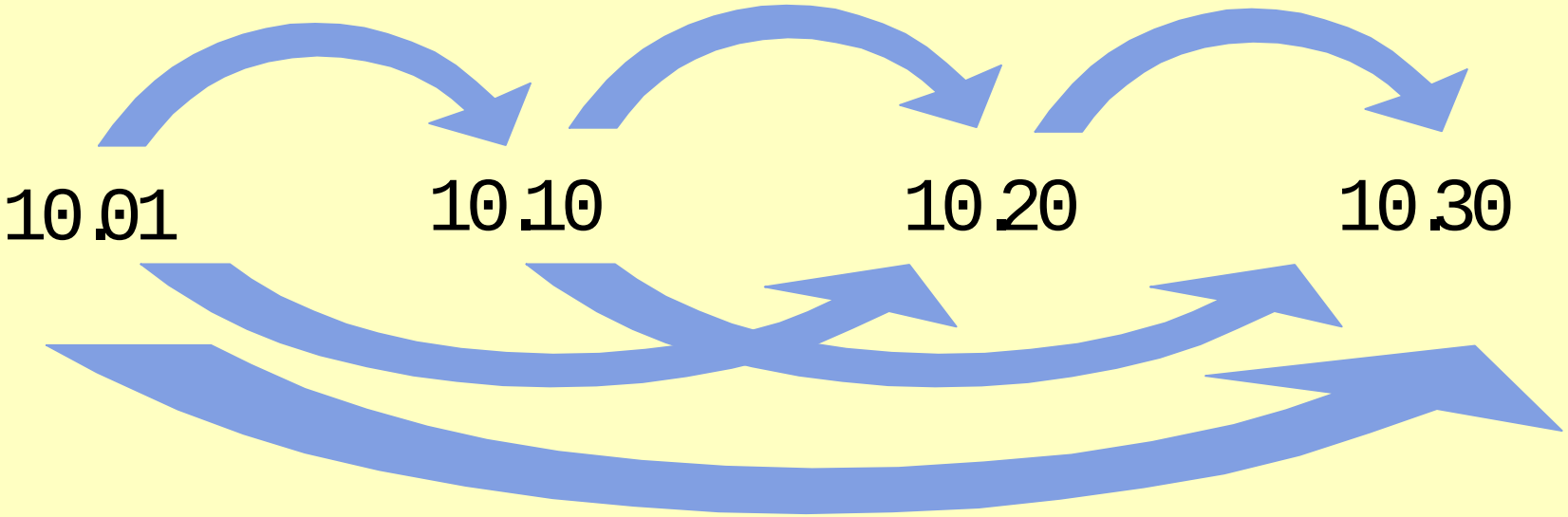
DAL

- Technical level of presentation
  - familiarity with System Administration tools
  - familiarity with Software Development concepts
  - familiarity with Operating System fundamentals
- Format of presentation
- References for further study
- Questions

"Largeness" Features in HP-UX: HP-UX 10.X Releases

DAL

- Each release is a superset of the previous
- HP-UX 10.01 is the "gateway" to the family
- Upgrade when and if you need



HP-UX 10.10 : CQ1 1996
Main new features

DAL

- 128 GB file system
- 3.75 GB RAM (T500, 0.75 GB cards)
- 1.9 GB process data space
- 60K File descriptors/process
- Shared LVM (SLVM) - for OPS
- Spec 1170 (UNIX 95)
- CDE - Common Desktop Environment
- 4 byte EUC commands
- DHCP server (including SAM management)
- SAM management of NIS
- 48 LAN card support

high-end focus

standards focus

SIZE EXTENSIONS

HP-UX 10.20 : CQ3 1996
Main new features

DAL

- Performance
  - PA-8000 optimization, Fibre Channel net/storage
  - MP tuning of transports, stacks, sockets, drivers
  - LVM tuning
  - Processor Affinity support
- Large files (local) - 128 GB files
- >60K UIDs - enable 4 billion user IDs
- 64-bit register math (PA 8000)
- 2.75 GB Shared Memory via patch
- Technical application performance
- DHCP client
- Distributed Print Services (Palladium)

high perf

high-end focus

standards focus

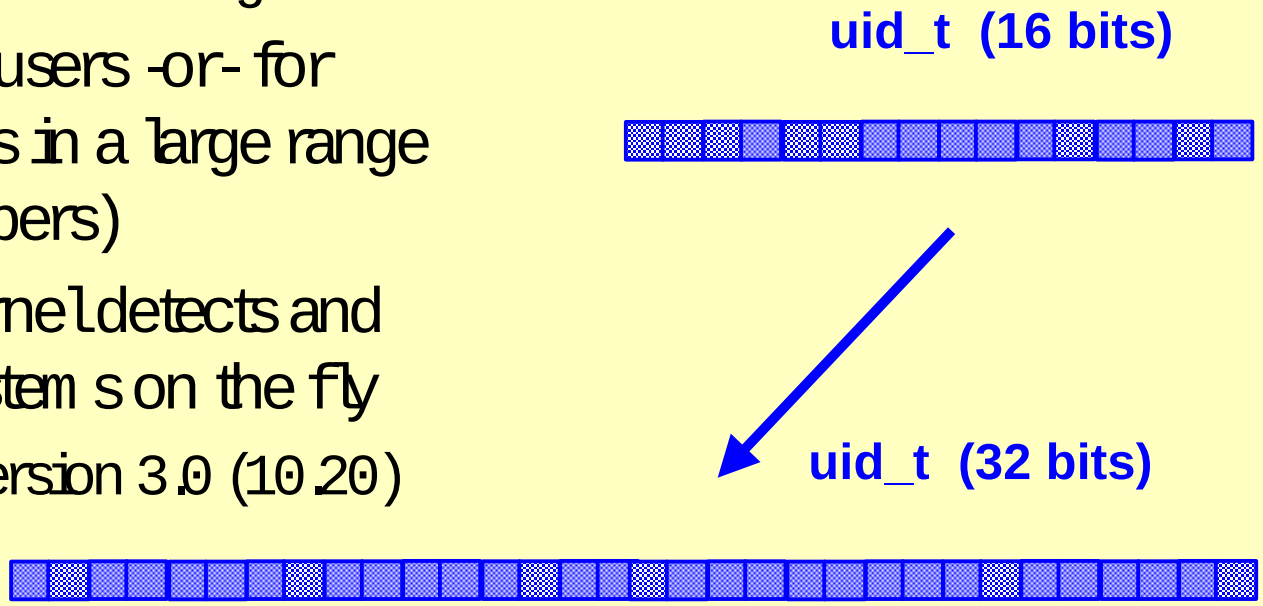
SIZE EXTENSIONS

Full performance for PA-8000 based systems

Large UIDs

DAL

- Base type of uid increased from 16 to 32 bits
- Changes to kernel, filesystems, libraries. commands, APIs
  - requires recompile to use Large UIDs
- For large number of users -or- for sparsely mapped uids in a large range (e.g.. telephone numbers)
- HFS: the HP-UX kernel detects and converts HFS filesystems on the fly
- VxFS: supported in Version 3.0 (10.20)



Using Large UIDs (continued)

DAL

```
 < /dev/vg00/lvol4 mounted on /tmp >
% fsadm /dev/vg00/rlvol4
file system          :     /dev/vg00/rlvol4
magic number         :                95014
feature bits         :                    1
file system supports : nolargefiles, longfilenames

% touch lg_uid_file
% chown 99999 lg_uid_file
% ls -l lg_uid_file
-rwxrwxrwx   1 99999   sys   0 Jan 21 15:09 lg_uid_file

% fsadm /dev/vg00/rlvol4
file system          :     /dev/vg00/rlvol4
magic number         :                95014
feature bits         :                    5
file system supports : nolargefiles, largeuids, longfilenames
```

<kernel has changed UID structures on the filesystem>

HP-UX 10.30 (limited release)
Main new features

DAL

- Performance
  - PA-8000 (continued)
- 1x1 kernel threads
- 128 GB files (networked)
- NFS Pv3, NIS+
- libc versioning
- Native OpenGL
- Support for new systems & peripherals
- Streams based TCP/IP
- BIND 4.9.3
- Year 2000 clean
- T600

high perf

high-end focus

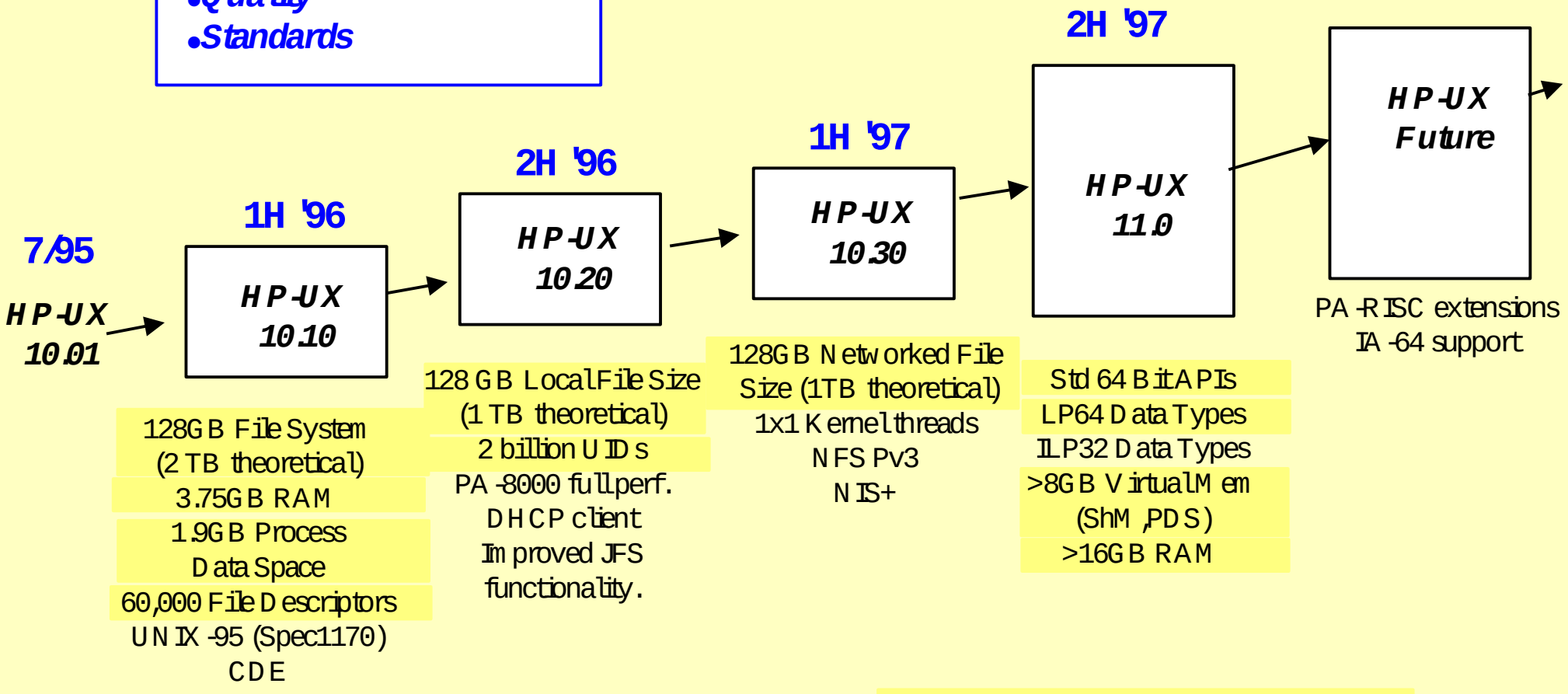
standards focus

SIZE EXTENSIONS

HP-UX Operating System Roadmap

DAL

- Performance
- Binary Compatibility
- Quality
- Standards



Extend OS Size Capability

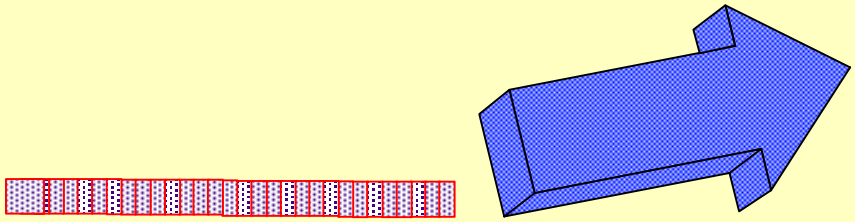
Market drivers to 64 bits

DAL

- Some Database vendors and technical software developers are leading the movement to 64-bit computing.
- They developed databases and applications that handle very large memory and enable access by many more users.

HP's 64-bit strategy: overview

DAL



HP-UX
HP PA-RISC

HP-UX
HP PA-RISC, HP/Intel IA-64

Key theme:
Implement evolutionary -- not revolutionary -- product strategies to deliver key new features and protect customers' software investments.

Elements of HP's evolutionary 64-bit strategy

DAL

Objective	Why important	Strategy
Lead industry in hardware and software features and performance	Meets customers' increasing functionality and performance demands	Collaborate with partners in developing next-generation UNIX and chip technologies
Provide smooth upgrade path	Ensures investment protection; minimizes upgrade costs in time and money	Continue policy of forward binary compatibility; 32- and 64-bit apps to coexist and communicate
Continue HP's long-standing commitment to standards	Gives partners flexibility in platform selection	Lead standards development and adherence

Benefits of 64-bit computing

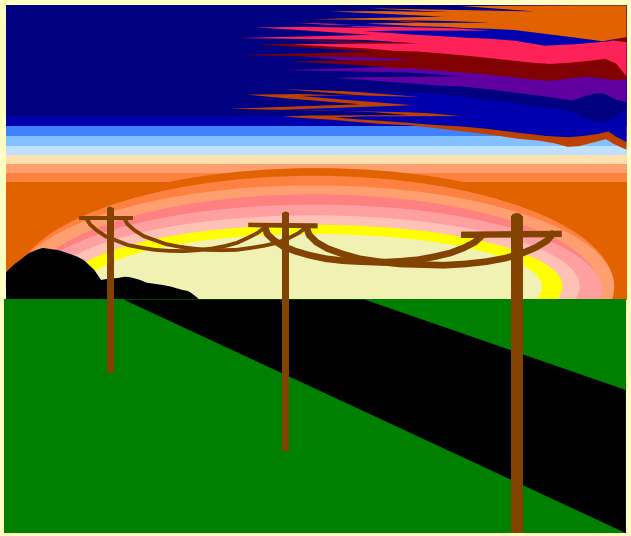
DAL

- Scalability
  - Larger applications and data and more users
- Potential gains in performance
  - Much larger amounts of data can reside in RAM, resulting in performance gains due to much less (time-consuming) swapping to disk
- These attributes make 64 bits well suited for certain high-end applications
  - very large DB and Decision Support
  - OLTP with 10's of thousands of users
  - complex technical simulations

Smooth upgrade to 64-bit environment

DAL

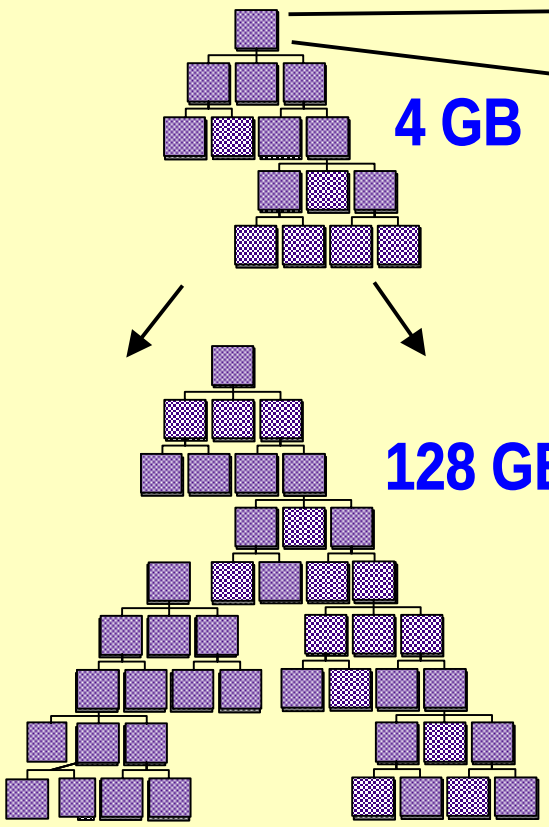
- Investment protection through forward binary compatibility
  - 32-bit applications may run unmodified on 64-bit HP-UX
- No migration; minimal end-user effort
  - No forced recompile
  - No forced recode
  - No data reload



Evolutionary introduction of 64-bit functionality into HP-UX

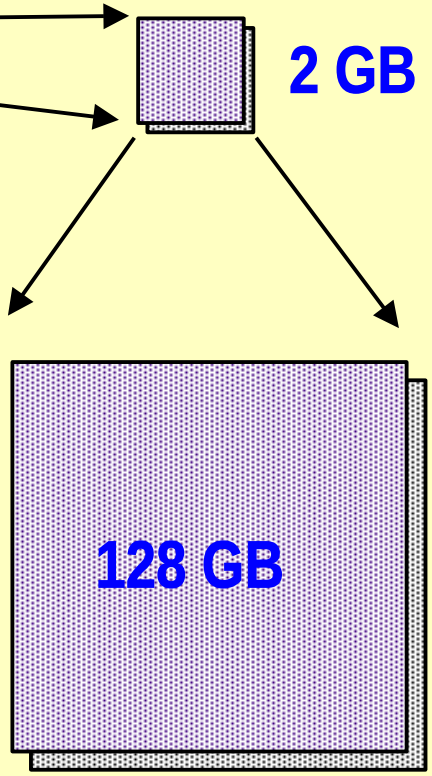
DAL

HP-UX 10.10, 2/96



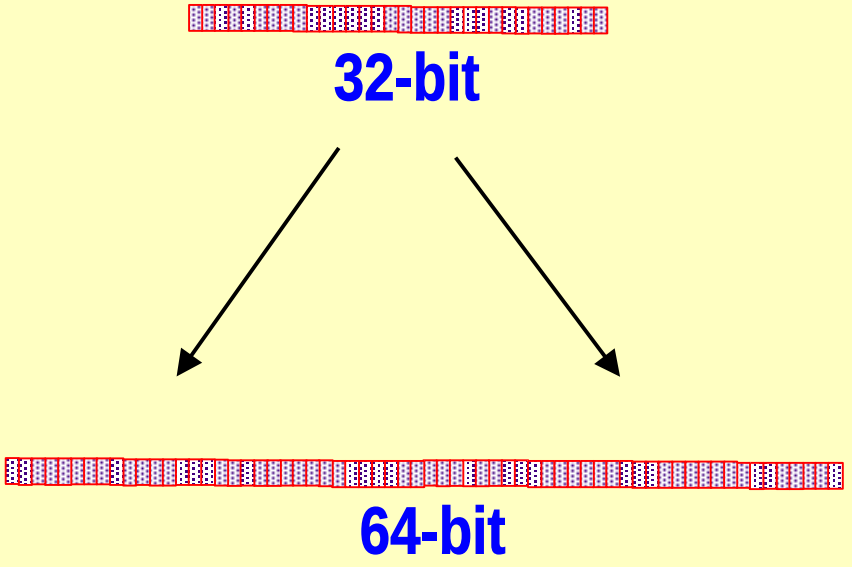
Large file system

HP-UX 10.20, 8/96



Large file size
64-bit register math

HP-UX 11.00, 1997



Large memory/addressing
64-bit data model

H

Operating System Data Models: ILP32 and LP64

DAL

	ILP32	LP64
Integer	32 bits	32 bits
Long	32 bits	64 bits
Pointer	32 bits	64 bits

- HP-UX 10.0 is ILP32 as are many other UNIXes
- HP-UX 10.[10,20,30] extended the OS capabilities
- HP-UX 11.0 comes in two versions:
  - ILP32
  - LP64
- can cross-develop between 32- and 64-bit OS versions
- can execute both 32- and 64-bit applications on 64-bit kernel
- can execute only 32-bit applications on 32-bit kernel

HP-UX Operating System : Specifications by Version

DAL

Attribute	HP-UX 10.01	HP-UX 10.10	HP-UX 10.20	HP-UX 10.30	HP-UX 11.00/32	HP-UX 11.00/64
Introduced	Jun 95	Feb 96	Aug 96	Aug 97	Nov 97	Nov 97
File system	4 GB	128 GB	128 GB	128 GB	128 GB	128 GB
File size	2 GB	2 GB	128 GB local,2GB network	128 GB local and network	128 GB local and network	128 GB local and network
Physical RAM	2 GB	3.75 GB	3.75 GB	3.75 GB	3.75 GB	4 TB
Shared Mem	1.75 GB	1.75 GB	2.75 GB	2.75 GB	2.75 GB	8 TB
Process data space	0.9 GB	1.9 GB	1.9 GB	1.9 GB	1.9 GB	4 TB
# File Descriptors	2,000	60,000	60,000	60,000	60,000	60,000+
# User IDs	60,000	60,000	2	2	2 billion	2 billion
Threads model	User	User	User	User and Kernel	User and Kernel	User and Kernel
Y2K Ready	patch	patch	patch	yes	yes	yes

HP-UX 11.0 Specifications

DAL

Attribute	32-bit version	64-bit version
CPUs supported	16	32
File system size	128 GB	128 GB
File size (local and networked)	128 GB	128 GB
Physical RAM	3.75 GB	4 TB
Shared memory	2.75 GB	8 TB
Process data space	1.9 GB	4 TB
File descriptors	60,000 plus	60,000 plus
User IDs	2 billion	2 billion
Threads model	User and Kernel	User and Kernel

HP-UX 11.0 - Supported Systems (Nov 1999 Extension Pack)

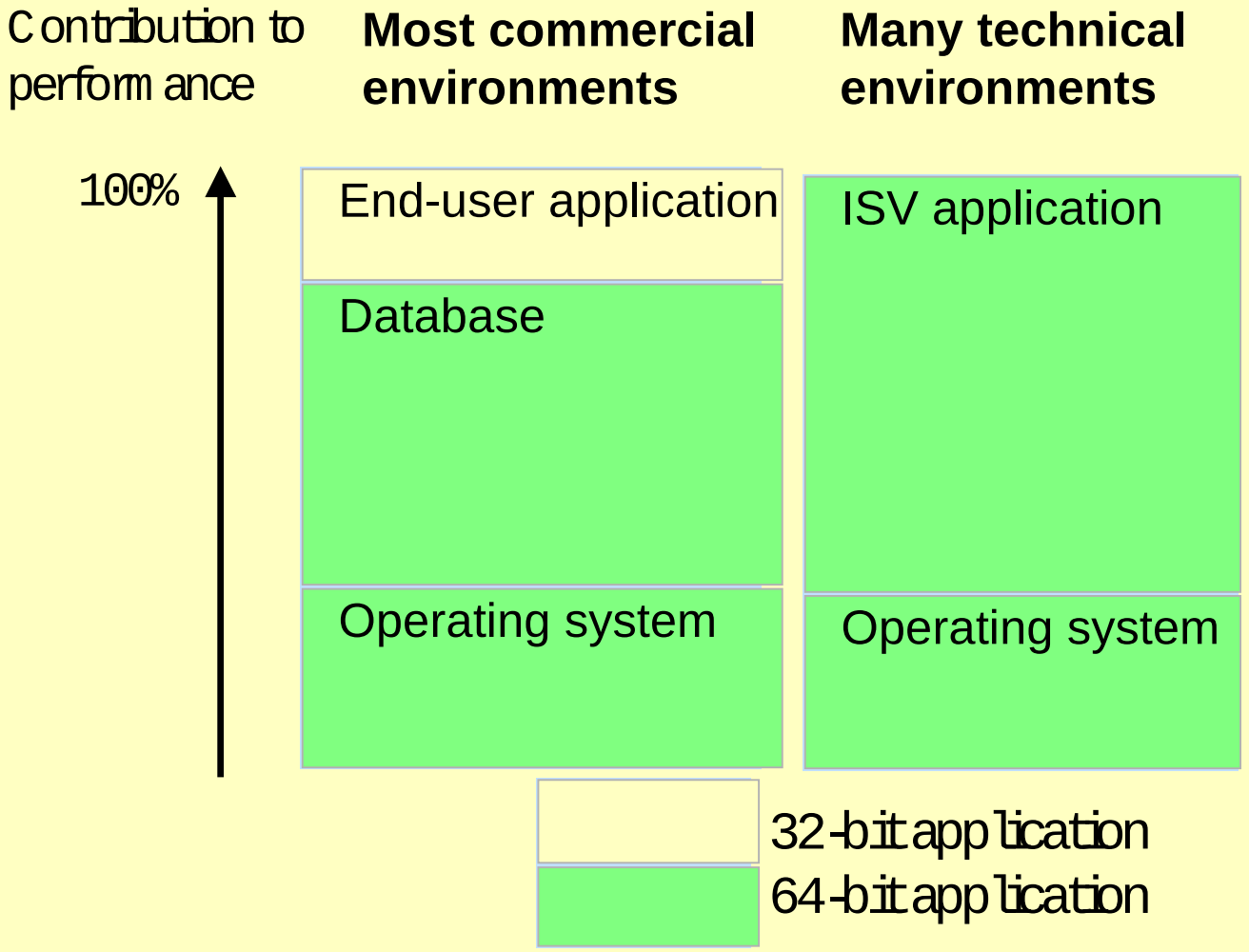
DAL

Series	Model	32-bit	64-bit
A-Class	A180, A180C	X	
D-Class	Dx10, Dx20, Dx30, Dx50, Dx60	X	
	Dx70, Dx80, Dx90	X	X
E-Class	E25, E35, E45, E55	X	
F-Class	F10, F20, F30	X	
G-Class	G30, G40, G50, G60, G70	X	
H-Class	H20, H30, H40, H50, H60, H70	X	
I-Class	I30, I40 I50, I60, I70	X	
K-Class	K100, K200, Kx10, Kx20	X	
	Kx50, Kx60, Kx60, Kx70, Kx80	X	X
L-Class	L1000, L2000		X
N-Class	N4000-[36,44]		X
R-Class	R380, R390	X	X
T-Class	T500, T520	X	
	T600	X	X
V-Class	V2200, V2250, V2500		X
EPS	EPS22, EPS23	X	X
	EPS40		X
700 Series	712, 715/[64,80,100,100XC], 725/100	X	
B-Class	B132L, B132L+, B160L, B180L	X	
	B1000		X
C-Class	C100, C110, C160L	X	
	C160, C180[XP], C200, C240, C360	X	X
	C3000		X
J-Class	J200, J210, J210XC	X	
	J280, J282, J2240	X	X
	J5000, J7000		X
J-Class	J280, J282, J2240	X	X

HP-UX users may gain performance increases without needing to recompile their application

DAL

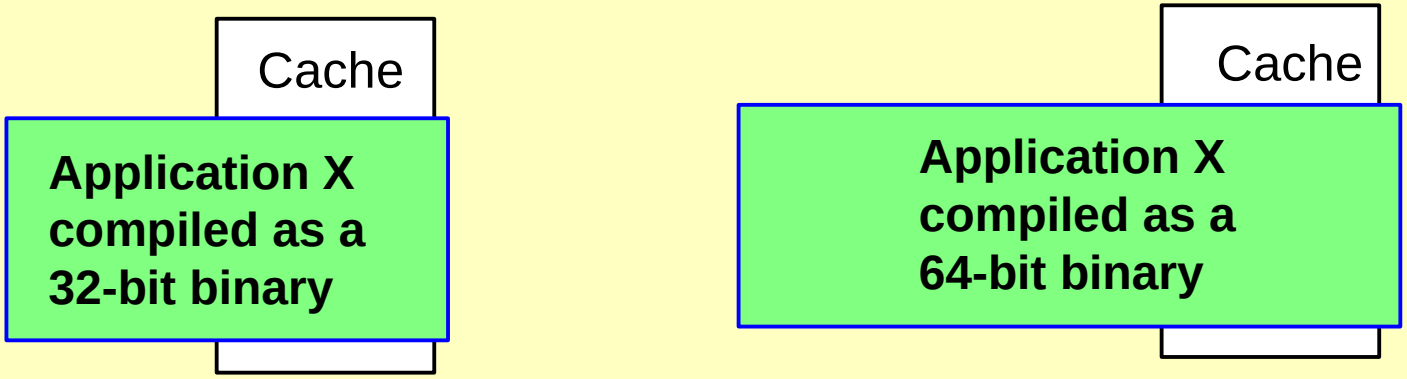
- HP-UX will allow an existing 32-bit end-user application to interact with a 64-bit database
- Most of the total performance gain will come from recompiles of key DB and ISV applications
- Customers need not recompile their applications



Mixing 32- and 64-bit applications

DAL

- 32-bit applications may run faster than recompiled 64-bit versions, due to "cache-fit" effect

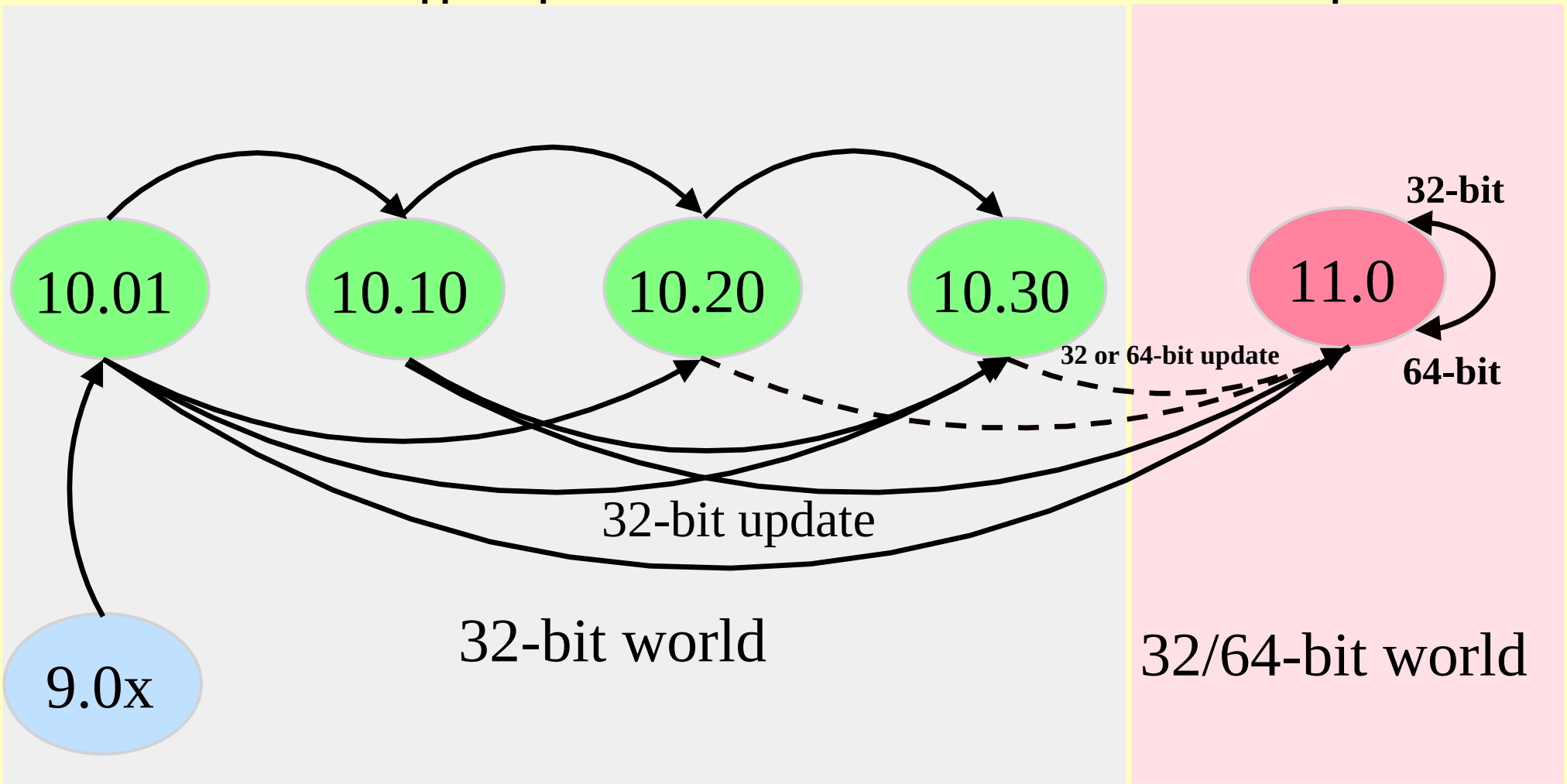


Best overall performance comes through 32- and 64-bit coexistence, which HP-UX provides

Upgrade Paths to HP-UX 11.0

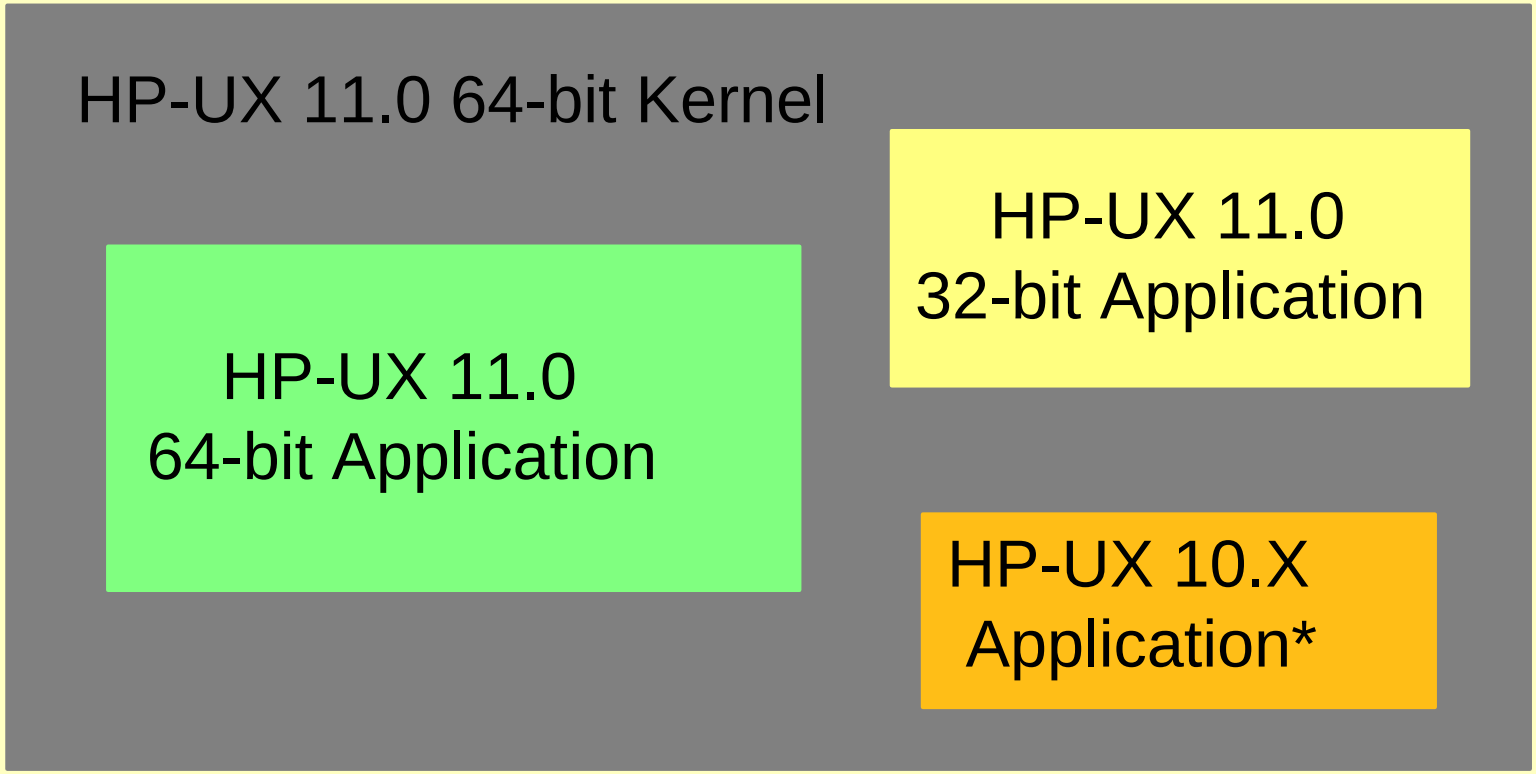
DAL

Software Distributor support updates from HP-UX 10.x to HP-UX 11.0 in one step



Applications that run on 64-bit HP-UX 11.0

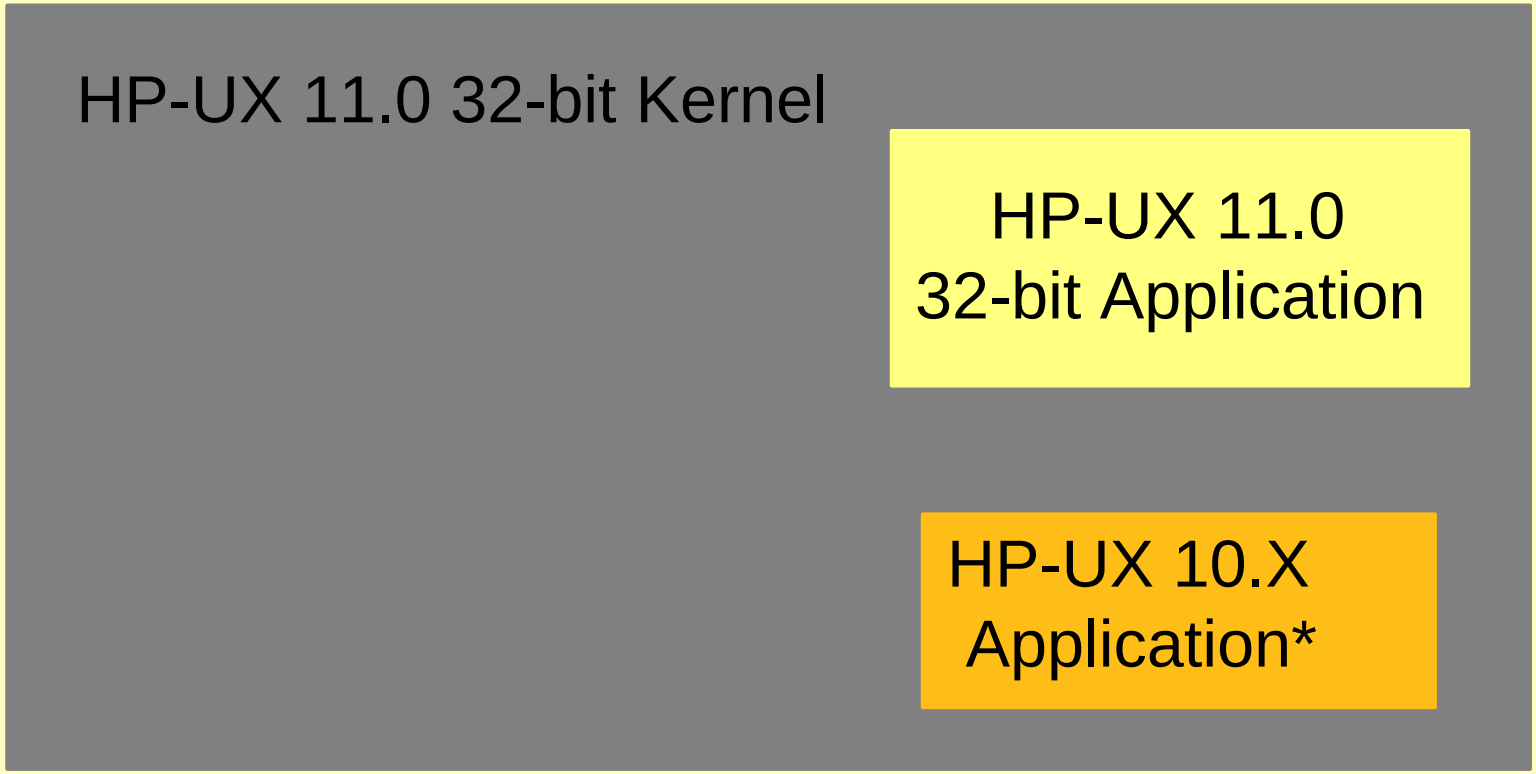
DAL



- Well-behaved applications that run on HP-UX 10.X run on HP-UX 11.0. (See Compatibility Guidelines)

Applications that run on 32-bit HP-UX 11.0

DAL

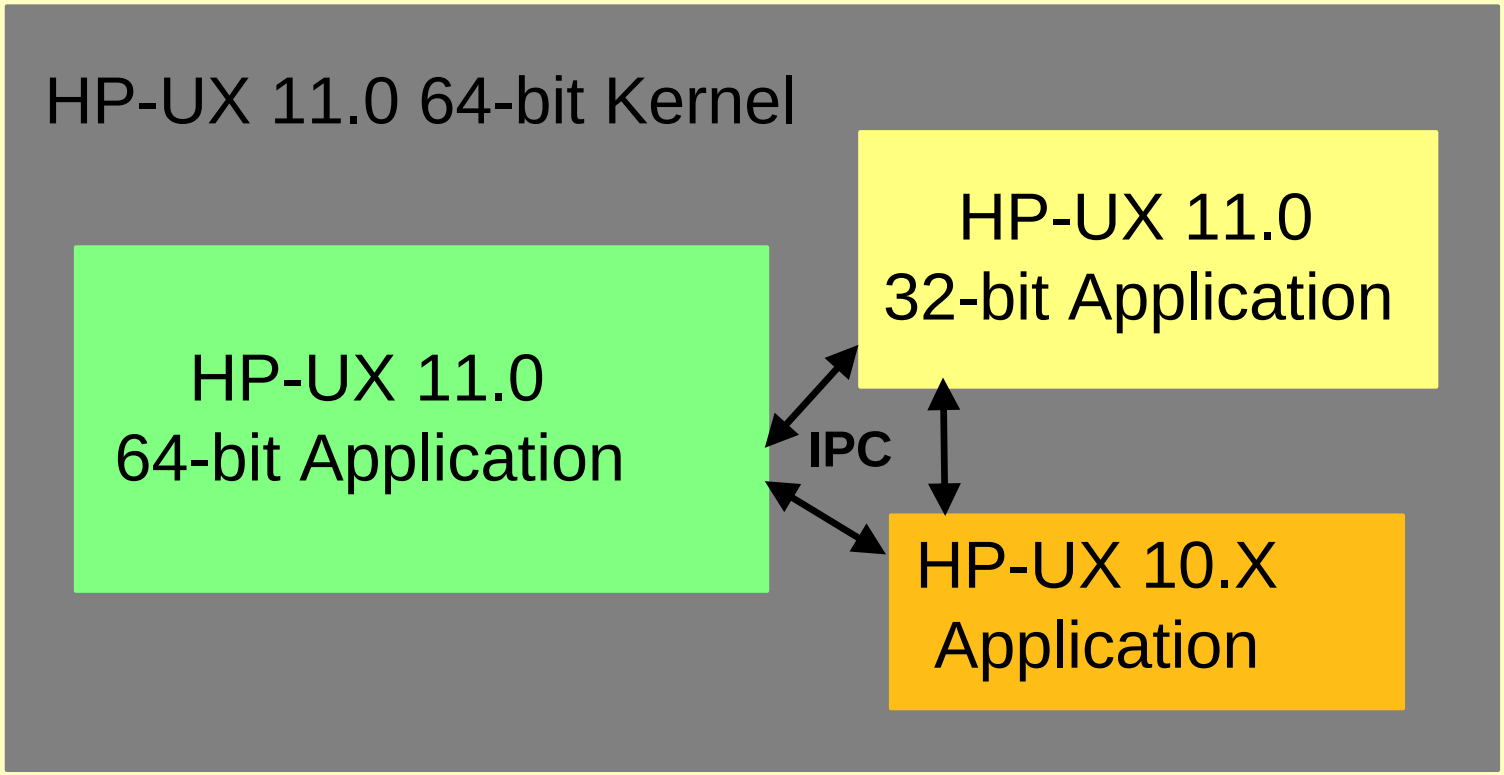


- Well-behaved applications that run on HP-UX 10.X run on HP-UX 11.0. (See Compatibility Guidelines)

Applications Interoperability on 64-bit HP-UX 11.0

DAL

- 32- and 64-bit applications can interoperate on 64-bit HP-UX using standard IPC mechanisms:
  - Shared Memory
  - Mapped Files
  - Sockets
  - Signals
  - Message Queues
  - Pipes
  - RPC



HP-UX Compilers

DAL

- Available for both 32- and 64-bit development
  - HP C
  - HP aC++
  - HP FORTRAN 90
  - HP Assembler
  - HP DDE
  - HP PAK
  - HP Linker toolset
- Available only for 32-bit development
  - HP FORTRAN 77
  - HP Pascal
  - HP MicroFocus COBOL
  - HP C++ (cfront)

Compiler Option/Hardware Architecture Run-time Compatibility

DAL

Compiler Option	PA-RISC 1.1 32-bit Systems	PA-RISC 2.0 64-bit Systems
+DA1.1	X	X
+DAportable	X	X
+DA2.0		X
+DD64 or +DA2.0W		X

+DD64 is a HP C option for compiling in 64-bit mode.

+DA2.0W is the HP aC++, HP Fortran90, and HP C option for compiling in 64-bit mode.

Compiler Options to Develop for 32- or 64-bit Applications

DAL

Compiler Option	What it Does
+DA1.1	Produce PA1.1 code
+DA2.0	Produce PA2.0 code (32 bit)
+DA2.0W	Produce PA2.0 code (64 bit)
+DD32	same as +DA1.1
+DD64	same as +DA2.0W

- all options are supported on both 32- and 64-bit systems
  - can cross-develop for either platform
- on 32-bit systems make sure you have installed the 64-bit library fileset (OS-CORE:C-MIN-64ALIB)

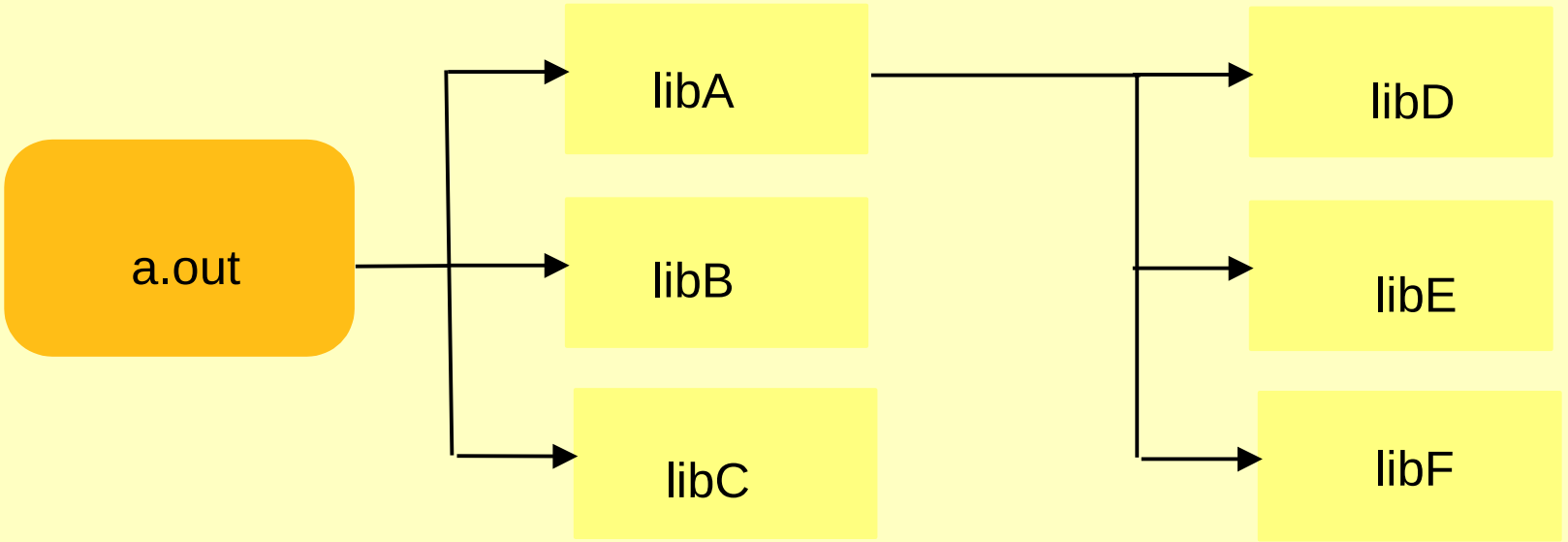
HP-UX Linker Changes

DAL

- A number of new/changed options to HP-UX linker - try "ld +help" or review STK White papers
- Apps compiled and linked in 32-bit mode will see no change in behavior.
- Apps compiled and linked in 64-bit mode use a runtime startup model similar to other SVR4 systems.

New Linker search order for 64-bit applications

DAL



- 64-bit mode: Breadth-first search:

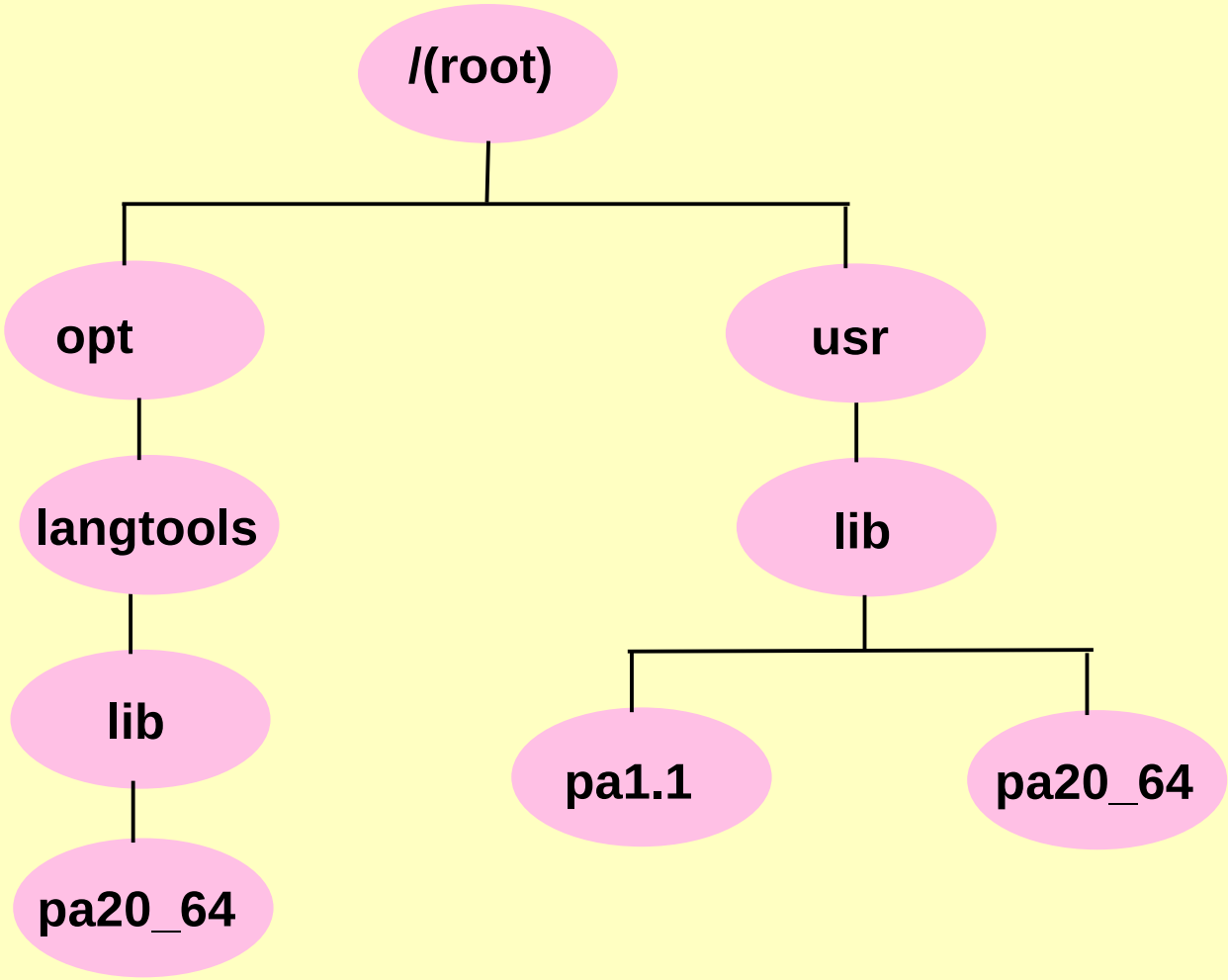
  a.out -> libA -> libB -> libC -> libD -> libE -> libF

- 32-bit mode: Depth-first search:

  a.out -> libA -> libD -> libE -> libFD -> libBE -> libC

New Library Paths

DAL



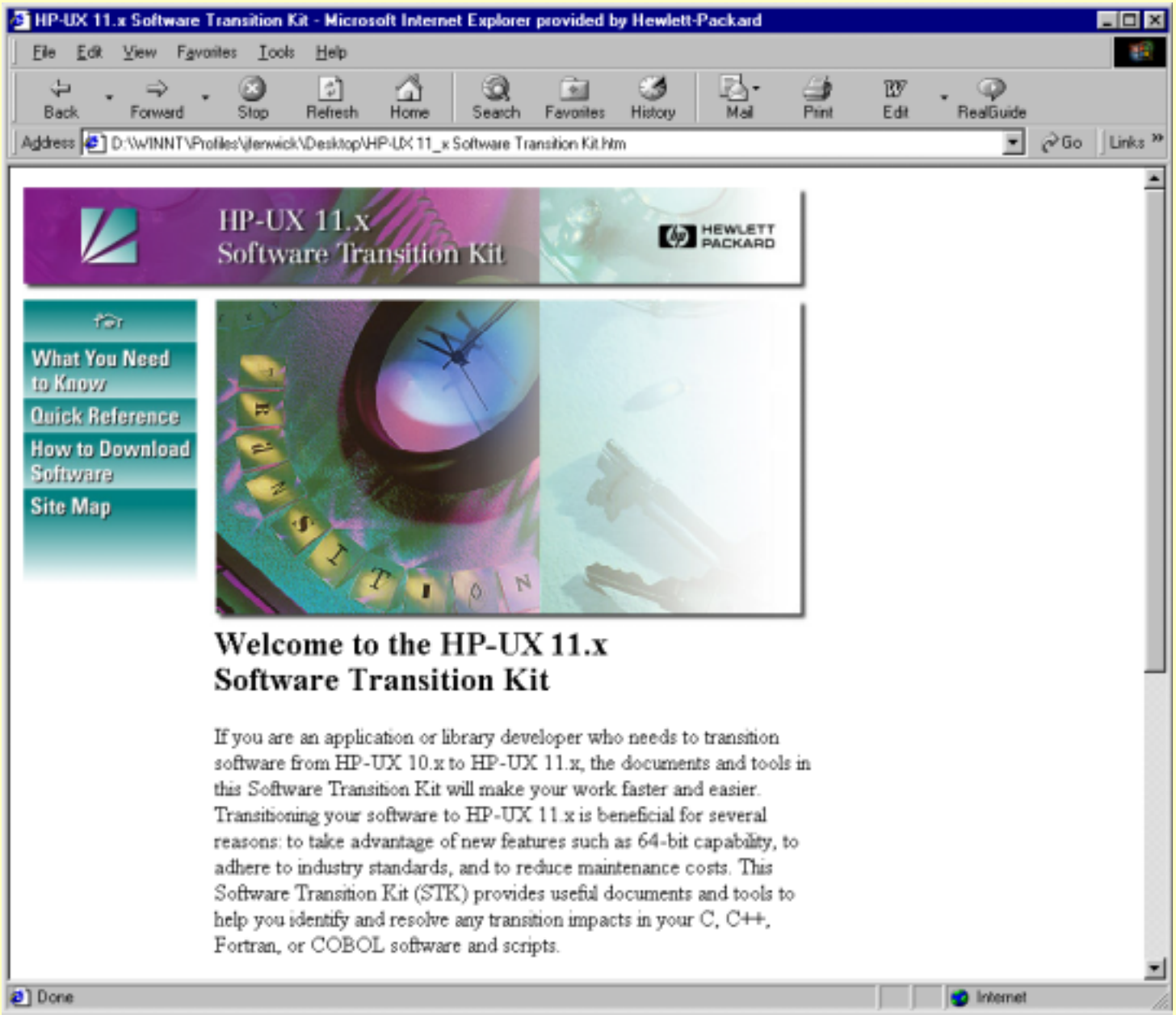
HP-UX Transition Tools

DAL

- HP-UX Software Transition Kit (STK). Current: 1.3.1
  - tools and documentation to discover changed APIs
  - works on C and C++ source code, scripts, makefiles
  - http://www.software.hp.com/STK/
- HP C compiler
  - both lint and the C compiler provide options to help transition to the 64-bit data model.
- HP-UX 11.0 Release Notes
  - documents system header file changes, system library changes, and lists 64-bit versions of system libraries.
- FlexeLint
  - identifies non-portable constructs in C and aC++ programs.

HP-UX 11.0 STK

DAL



HP-UX 32-bit and 64-bit Base Data Types

DAL

Data Type	ILP32 size(bits)	LP64 size(bits)
char	8	8
short	16	16
int	32	32
long	**32**	**64**
long long	64	64
pointer	**32**	**64**
float	32	32
double	64	64
long double	128	128
enum	32	32

ILP32 to LP64 Porting Concerns

DAL

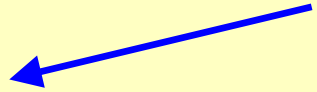
- Fundamental changes:
  - longs and ints are no longer the same size
  - pointers and ints are no longer the same size
  - pointers and longs are 64 bits and are 64-bit aligned
  - Predefined types size_t and ptrdiff_t are 64-bit integral types
- Potential impact:
  - data truncation
  - data type promotion
  - constants
  - enumerated types
  - pointers
  - data alignment and data sharing
  - bit shifts and bit masks
  - bit fields

Example: code works on 32-bit and will fail on 64-bit

DAL

```
int main ()
{
  int *buffer;
  buffer = malloc(sizeof(int));      *buffer = 1234;
  printf("Buffer address: %p\n",     &buffer);
  printf("Buffer contents: %p\n",     buffer);
  printf("Dereferenced value: %d\n", *buffer);
  return 0;
}
```

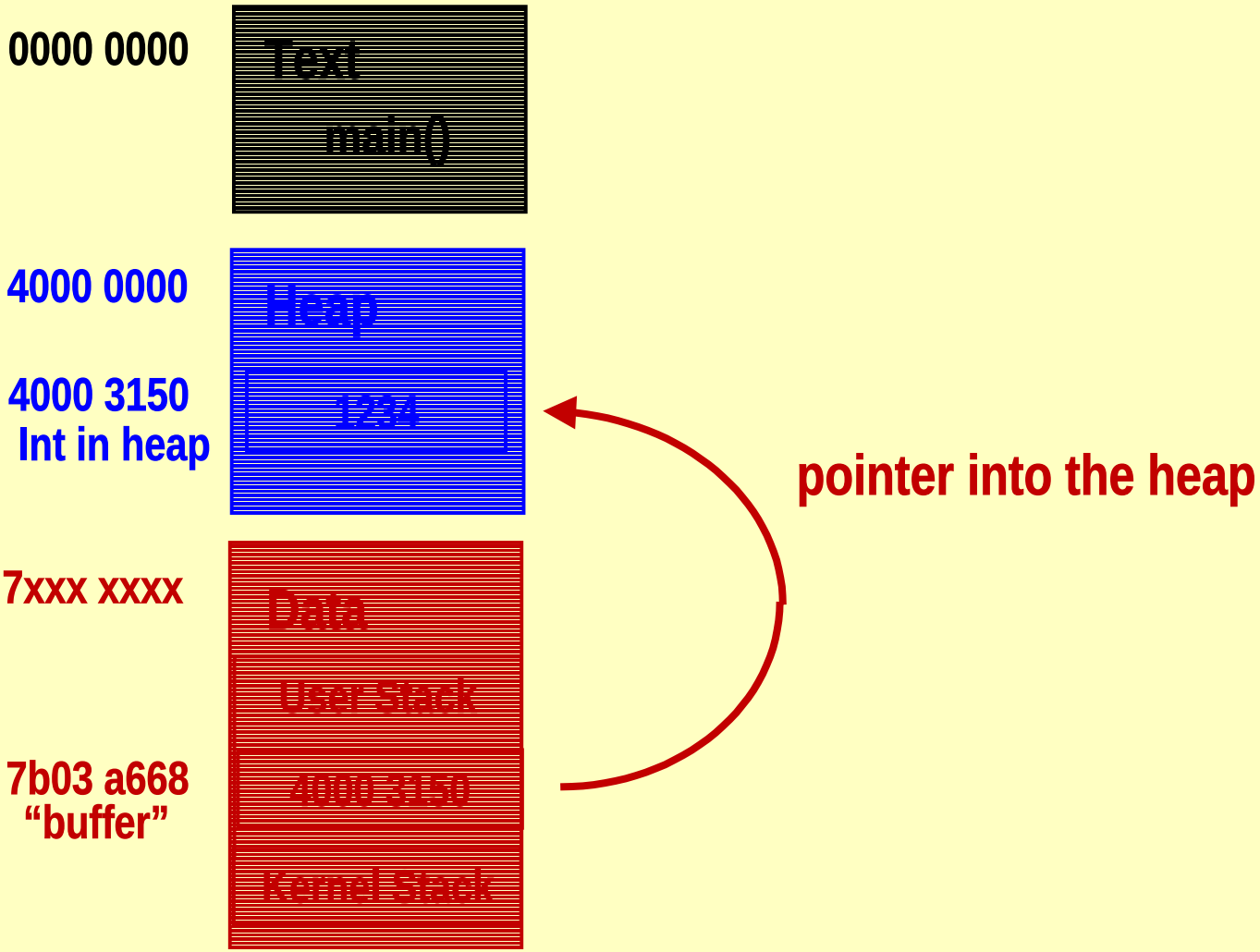
HP-UX 10.20
ILP32



```
$ ./malloc_return
Buffer address: 7b03a668   <-- address in data segment
Buffer contents: 40003150  <-- address in heap
Dereferenced value: 1234   <-- dereference ptr in heap
```

Example Program - Memory Map

DAL

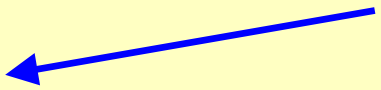


Example: code works on 32-bit and will fail on 64-bit

DAL

```
int main ()
{
  int *buffer;
  buffer = malloc(sizeof(int));
  *buffer = 1234;
  printf("Buffer address: %p\n",     &buffer);
  printf("Buffer contents: %p\n",     buffer);
  printf("Dereferenced value: %d\n", *buffer);
  return 0;
}
```

```
$ getconf KERNEL_BITS
64
$ cc +DD64 -o malloc_return
malloc_return.c
$ ./malloc_return
Memory fault(coredump)
```



HP-UX 11.0 LP64

Example: code works on 32-bit and will fail on 64-bit

DAL

```
int main ()
{
  int *buffer;
  buffer = malloc(sizeof(int));
  *buffer = 1234;
  printf("Buffer address: %p\n",     &buffer);
  printf("Buffer contents: %p\n",     buffer);
  printf("Dereferenced value: %d\n", *buffer);
  return 0;
}
```

- In C an undefined function return value is integer type 32 bits
- malloc returns a pointer type which is now 64-bits
- pointer.64 to integer.32 data truncation -> invalid pointer value
- invalid pointer dereference -> core dump

Example: code works on 32-bit and will fail on 64-bit

DAL

```
#include <stdlib.h>
int main ()
{
  int *buffer;
  buffer = malloc(sizeof(int));
  *buffer = 1234;
  printf("Buffer address: %p\n",     &buffer);
  printf("Buffer contents: %p\n",     buffer);
  printf("Dereferenced value: %d\n", *buffer);
 return 0;
}
```

- Correct this by including the appropriate function declaration from the include file <stdlib.h>
- lint would have detected this missing function declaration

Example: code works on 32-bit and will fail on 64-bit

DAL

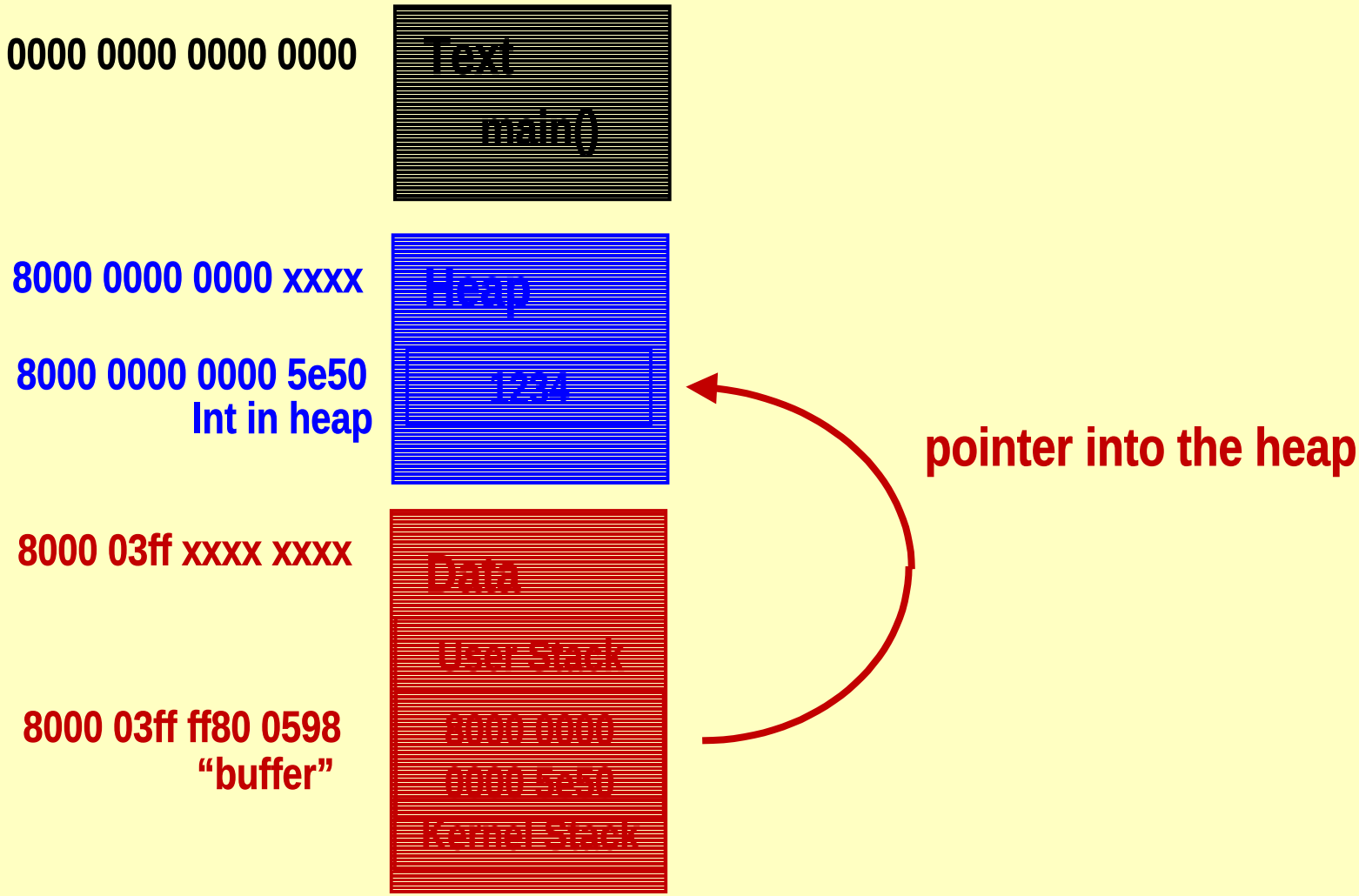
```
#include <stdlib.h>
int main ()
{
  int *buffer;
  buffer = malloc(sizeof(int));
  *buffer = 1234;
  printf("Buffer address: %p\n",
        &buffer);
  printf("Buffer contents: %p\n",
        buffer);
  printf("Dereferenced value: %d\n",
        *buffer);

 return 0;
}
```

HP-UX 11.0 LP64

```
$ cc +DD64 -o malloc_return2 malloc_return2.c
$ ./malloc_return2
Buffer address: 800003ffff800598  <-- data segment
Buffer contents: 8000000000005e50 <-- ptr in heap
Dereferenced value: 1234          <-- dereference ptr.
```

Example Program - Memory Map

DAL



Example: compiling and running on 32- and 64-bit systems

DAL

```
/*
 * test_basetypes: display maximum values of base data types
 */

#include <values.h>

main()
{
   printf("** Maximum values of base types on this system **\n");
   printf("Max int  = %ld\n",MAXINT );
   printf("Max long = %ld\n",MAXLONG);
}
```

Example: compiling and running on 32- and 64-bit systems

DAL

```
$ uname -a
HP-UX te887-01 B.11.00 A 9000/887 312087281 two-user license

$ getconf KERNEL_BITS
32

$ cc -o test_basetypes test_basetypes.c          <-default compile: 32-bit

$ file test_basetypes
test_basetypes: PA-RISC1.1 shared executable dynamically linked -not stripped

$ ./test_basetypes
** Maximum values of base types on this system **
Max int  = 2147483647
Max long = 2147483647

$ cc -o test_basetypes +DD64 test_basetypes.c    <- compile for 64-bit

$ file test_basetypes
test_basetypes: ELF-64 executable object file - PA-RISC 2.0 (LP64)

$ ./test_basetypes                               <- cannot execute 64-bit application
sh: ./test_basetypes: Execute permission denied.      on 32-bit system
```

Example: compiling and running on 32- and 64-bit systems (continued)

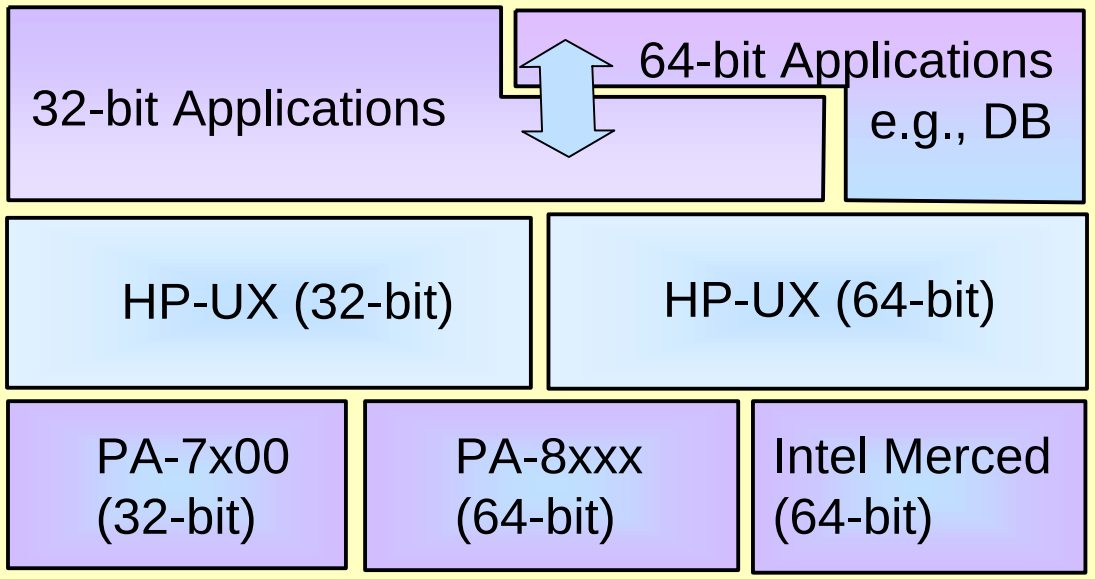
DAL

```
$ getconf KERNEL_BITS
64

$ cc -o test_basetypes test_basetypes.c          <-default compile: PA 2.0 32-bit
/usr/ccs/bin/ld: (Warning) At least one PA 2.0 object file (test_basetypes.o)
  was detected. The linked output may not run on a PA 1.x system.

$ ./test_basetypes                               <-run the PA 2.0 32-bit application
** Maximum values of base types on this system **
Max int  = 2147483647
Max long = 2147483647

$ cc -o test_basetypes +DD64 test_basetypes.c    <-compile for PA 2.0 64-bit

$ file test_basetypes
test_basetypes: ELF-64 executable object file - PA-RISC 2.0 (LP64)

$ ./test_basetypes                               <-run the PA 2.0 64-bit application
** Maximum values of base types on this system **
Max int  = 2147483647
Max long = 9223372036854775807

$ cc -o test_basetypes +DA1.1  test_basetypes.c  <-compile for PA 1.1 64-bit
```

Smooth Upgrade to 64-bit Environment

DAL

- End-users may recode and/or recompile their 32-bit applications at their leisure to access 64-bit functionality directly, or ...
- Run existing 32-bit applications unchanged and access 64-bit functionality of 64-bit databases or other ISVs



HP-UX will allow co-existence and interoperability of 32- and 64-bit applications

HP-UX 11.0/10.30: Major New Functionality

DAL

- Merged binaries and merged installmedia
- Kernelthreads
- Performance optimized page size
- Dynamically loadable kernelmodules
- Fastcore dump/recovery
- Fastpassword lookup
- IgniteUX
- New system recovery capability
- Year2000 safety
- Other new technologies

HP-UX 11.0 Feature Set New and Obsoleted

DAL

- For a complete reference and pointers to other documentation, please browse the HP-UX Web Site:
  - http://www.enterprisecomputing.hp.com/
  - http://www.unixsolutions.hp.com/
- Other relevant web sites:
  - http://docs.hp.com (http://www.docs.hp.com)
  - http://software.hp.com (http://www.software.hp.com)

Merged binaries and install media

DAL

- Merged install media
  - Install media now common for both S800 and S700
  - only 1 set of CD-ROMs
- Most S800/S700 dependencies removed from core OS
  - optional products (e.g., graphics) may still vary

- Merged install kernels
  - cold install kernel is the same for both S800 and S700

Merged binaries and install media (continued)

DAL

- Merged binaries in core OS for 32- and 64-bit systems
  - every command in the core OS is built in 32-bit mode and can operate on either a 32- or 64-bit system
  - some kernel-specific commands are both 32- and 64-bit capable (such as ps and dmesg)
- Machine IDs and specifiers
  - future machine IDs are to be S700/S800 neutral
    - remove any dependency on S700/S800 machine id
  - New V-class machine types:

	uname -m	model
V2200	9000/800	9000/800/V2200
V2500	9000/800	9000/800/V2500

New ELF object file format for 64-bit executables

DAL

- For 32-bit applications, the object file format remains SOM-32 (Spectrum Object Module).
- For 64-bit applications, the object file format is now ELF-64 (Executable and Linking Format)
  - ELF: a binary format developed by UNIX System Laboratories, used on Solaris, Linux, others.
  - New executable file startup mechanism
  - More efficient executable file format
- Why keep SOM for 32-bit and adopt ELF for 64-bit?
  - Maintain compatibility existing (32-bit) code
  - gain benefits for newly developed (64-bit) code

HP-UX Threads in the 10.X Releases

DAL

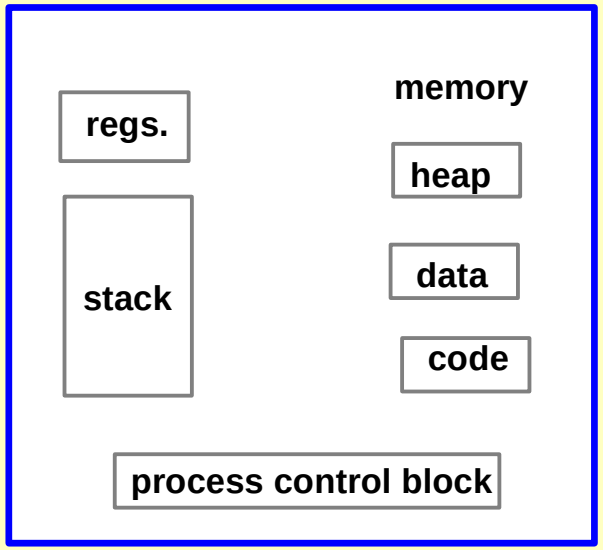
- HP-UX 9.X and 10.X offered User Space threads
- All Thread APIs conform to POSIX standards
  - (Pthread standard)
- Threads libraries, include files, and documentation delivered with the core DCE product
- Implementation based on OSF DCE Threads
- Actual thread libraries are contained in libcma
  - CMA = Concert Multithreaded Architecture

Thread Definition

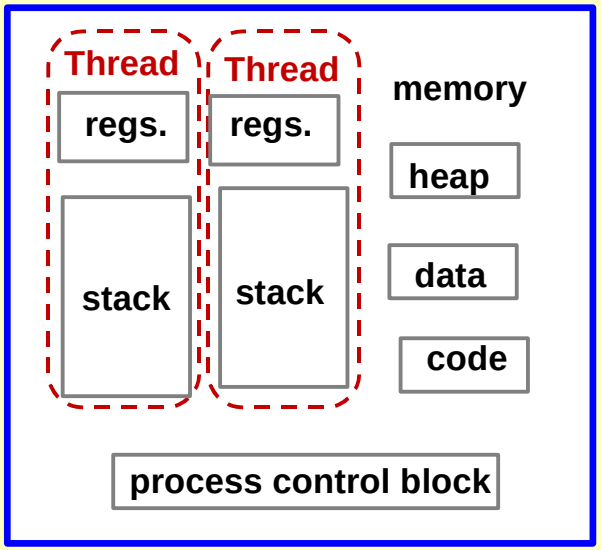
DAL

- A single sequential flow of control that can coexist with other threads in the same process.
- A thread is described by:
  - a unique identifier - Thread ID (TID)
  - shared address space with other threads in process
  - scheduling priority and policy

Traditional Process



Multi-threaded Process



User Threads and Kernel Threads

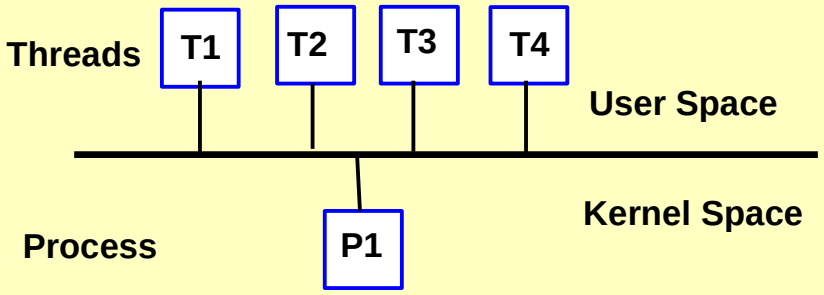
DAL

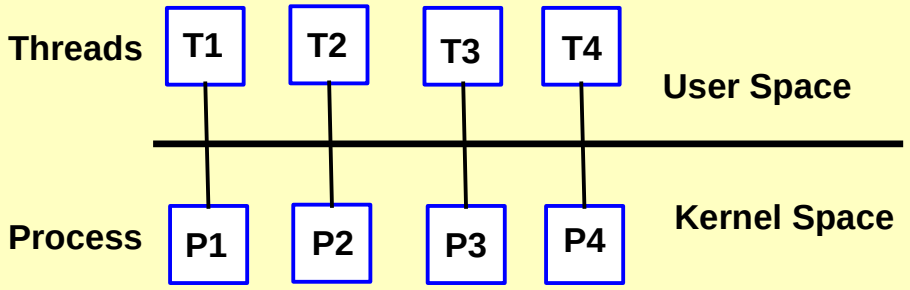
User Threads

- handled in User Space and controlled using the threads APIs provided in the threads libraries
- Mx1:Many to One Model. Many Threads exist inside 1 Process that is visible to the Kernel

Kernel Threads

- handled in User and Kernel space and are kernel schedulable entities visible to the operating system
- 1x1: One to One Model. one Thread in one Kernel process. HP-UX implements with LWP (Light Weight Process).





User Threads vs. Kernel Threads

DAL

M x 1 User Threads

- all threads mapped into one process
- not visible to kernel
- fast creation/control
- no kernel overhead
- if one thread blocks, all threads are blocked

1 x 1 Kernel Threads

- each user thread mapped to kernel thread
- each thread scheduled independently, so if one blocks others can still run
- some overhead due to kernel calls

Kernel Threads
HP-UX Implementation

DAL

- Kernel thread infrastructure delivered in the 10.20 Release:
  - thread structure in kernel
  - kernel scheduling done on thread objects
- Full implementation of Kernel Threads delivered in 10.30 and 11.0 Releases

Threads - Concurrency and Parallelism

DAL

- Concurrency - multiple threads may be in progress at the same time
- Parallelism - multiple threads are executing at the same time
  - On a MP system, KernelThreads may execute in parallel on different CPUs
  - potentially a huge performance win

POSIX Pthreads

DAL

- Pthreads can be implemented with Mx1, 1x1, or MxN models
- POSIX Standard: P1003.1c Portable API for Threads Extensions to POSIX 1003.1(a,b)
- HP-UX UserSpace (DCE) threads comply with POSIX Draft4.
- HP-UX Kernelthreads (HP-UX 11.0) comply with POSIX Draft10.

Programming with Threads

DAL

- Write programs to create and execute simple multi-threaded processes.
- Will demonstrate the basics of thread programming
- Will demonstrate Multi-Processor scaling and concurrency issues
- Test program will be a multi-threaded: "Hello, World!", then will become a form of compute-bound application

Programming with Threads program pseudo-code

DAL

```
/*
 *  Create and execute 2 threads (pseudo-code)
 */

main()
{
  pthread_create(thread1);          /* create first  thread   */
  pthread_create(thread2);          /* create second thread   */

  pthread_join(thread1);            /* wait for first  thread */
  pthread_join(thread2);            /* wait for second thread */
}

thread1()                           /* Thread 1 will run here */
{
  printf("Hello, World!\n");
}

thread2()                           /* Thread 2 will run here */
{
  printf("Goodbye, World!\n");
}
```

Programming with Threads - program code

DAL

```
/*
 *  Create and execute 2 threads (main program)
 */

#include <pthread.h>
void    thread1();                          /* forward declarations  */
void    thread2();                          /* for thread start addr */

main()
{
  pthread_t  tid1,tid2;                     /* Thread ID values */

  /* Create threads */
  pthread_create(         &tid1,            /* Thread ID (returned value) */
      (pthread_attr_t *)  NULL,             /* thread attributes */
          (void *(*)() )  thread1 ,         /* thread executes this func */
               (void *)   NULL   );         /* argument passed to thread */
  pthread_create(         &tid2,
      (pthread_attr_t *)  NULL,
          (void *(*)() )  thread2,
               (void *)   NULL   );

  /* wait for the threads to finish */
  pthread_join(           tid1,             /* wait on this thread */
               (void **)  NULL ) ;          /* thread's exit value */
  pthread_join(           tid2,
               (void **)  NULL ) ;
}
```

Programming with Threads - program code (continued)

DAL

```
/*
 * Functions where the two threads will start executing
 */

void thread1()
{
  printf("Hello, World!\n");
  pthread_exit( (void *)  NULL);          /* return a value and exit */
}

void thread2()
{
  printf("Goodbye, World!\n");
  pthread_exit( (void *)  NULL);
}
```

Programming with Threads - compiling and executing

DAL

- compile the program

```
% cc  +DA1.1  -lpthread  -o 2threads  2threads.c
```

  - compile +DA1.1 so the binary will be portable and will run on any supported 11.0 system
  - Pthreads library located at /usr/lib/libpthread.[a,sl]

    (for DCE threads link against libcma)
  - default compilation builds a shared-linked executable
- run the program

```
% ./2threads
Hello, World!
Goodbye, World!
```

Programming with Threads - order of execution

DAL

- run the program

```
% ./2threads
Hello, World!
Goodbye, World!
```

... OR MAYBE ...

```
% ./2threads
Goodbye, World!
Hello, World!
```

- Is the order of execution guaranteed?

  It is not guaranteed unless one uses the thread scheduling and synchronization APIs

Programming with Threads - compute-bound example

DAL

- Each thread will now execute a basic example of a compute-bound application
- Threads will execute concurrently as the kernel schedules each thread
- Threads may execute in parallel on a multi-processor system
- Default thread scheduling policy on HP-UX is to schedule threads across different processors.

Programming with Threads - compute-bound program code

DAL

```
#include <pthread.h>
void    busywork();                     /* where the threads execute  */

main()
{
  pthread_t  tid1,tid2;

  /* Create threads */
  pthread_create(          &tid1,       /* Thread ID (returned value) */
      (pthread_attr_t *)   NULL,        /* thread attributes          */
          (void *(*)() )   busywork,    /* thread executes busywork   */
               (void *)    10000 );     /* argument passed to thread  */

  pthread_create(          &tid2,
      (pthread_attr_t *)   NULL,
          (void *(*)() )   busywork,
               (void *)    10000 );

  /* wait for the threads to finish */
  pthread_join( tid1, (void **) NULL); }
  pthread_join( tid2, (void **) NULL);
}
```

```
void busywork(int maxcount)
{
  int count1, count2;
  for (count1=0; count1<maxcount; count1++)
    for (count2=0; count2<maxcount; count2++);
  pthread_exit( (void *)  NULL);
}
```

busywork: loop for (count*count) times

Programming with Threads - compute-bound execution

DAL

- Program compiled the same as before

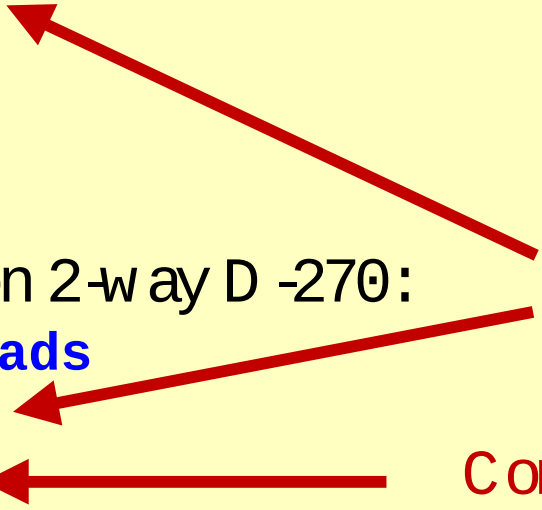
```
% cc +DA1.1 -lpthread -o 2busythreads 2busythreads.c
```

- execute the program on 1-way C-110:

```
% timex 2busythreads
real        17.10
user        16.79
sys          0.03
```

- execute the program on 2-way D-270:

```
% timex 2busythreads
real         5.06
user        10.05
sys          0.01
```



Real(wallclock)time

Compute time 2X realtime!

Programming with Threads - compute-bound example

DAL

- Realtime (wallclock time) is halved on the 2-way system
  - Each thread executes in parallel on a different processor
- This program MP-scales perfectly to a 2-way system, but would not scale any better on a system with more than 2 processors installed.
- This simple program example is not concerned with synchronization, global resources and contention, asynchronous events - many of the interesting elements of real-world programming.

Threads References

DAL

- Thread Time, Scott Norton and Mark DiPasquale, HP Press and Prentice-Hall, ISBN 013-190067-6
- Multi-Threaded Programming under HP-UX, A. Jyothy Ready, HP World August 1996
- HP-UX Whitepaper in the documentation directory /usr/share/doc/proc_mgt.[ps,txt]
- FAQ in comp.programming.threads

Large VM Page Size

DAL

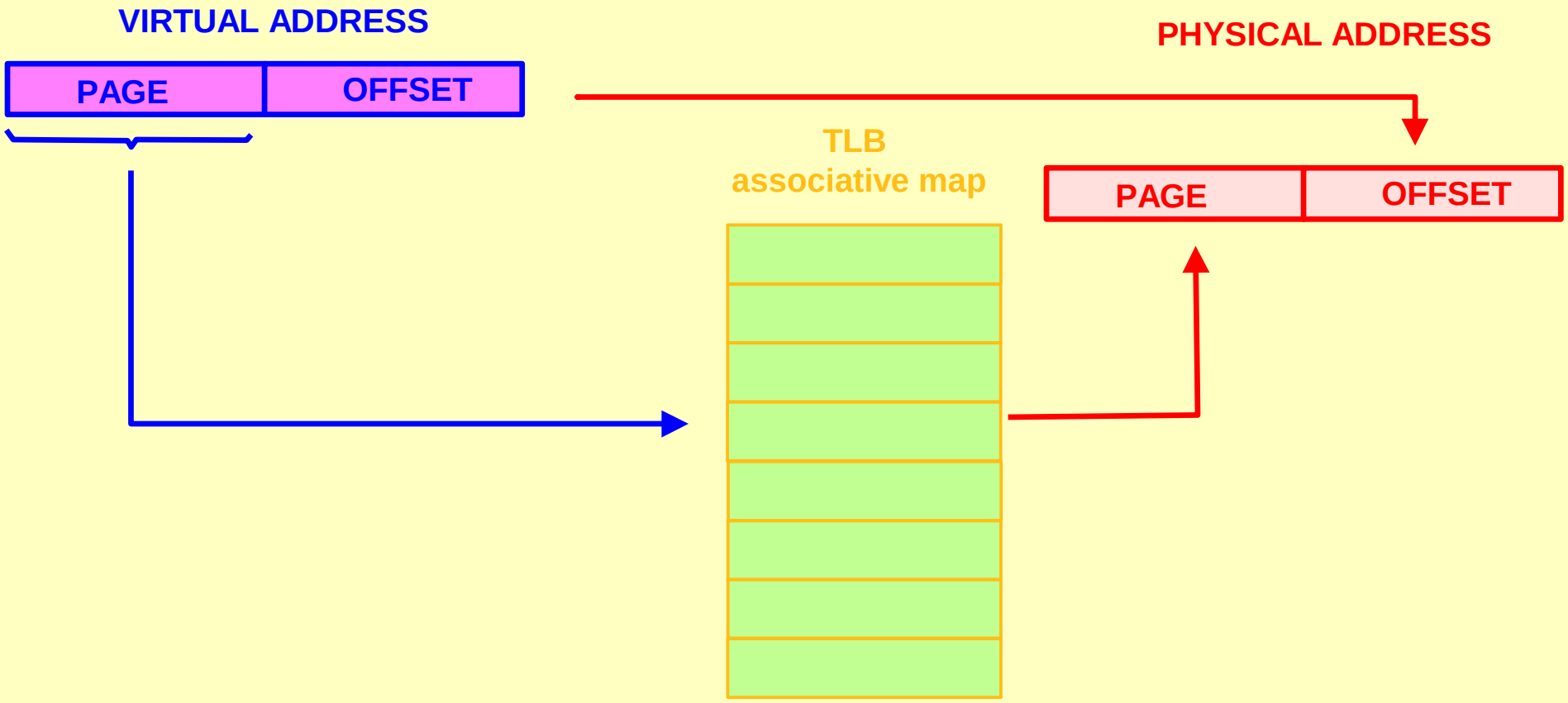
- Virtual Memory Page Size was fixed at 4 KBytes for all HP-UX releases prior to 10.20
- HP-UX 10.20 introduced Large Pages - VM page size for an executable process could be explicitly set using the chatr command to:

4K	1M	256M
16K	4M	L (use largest available size)
64K	16M	D (use kernel default size)
256K	64M	

- Page size could be set for Text, Data, Stack, Memory Mapped Files, and other memory objects.
- Only for PA-8000 processors

Virtual Memory Page Lookups and Translation Lookaside Buffer (TLB)

DAL



TLB is a hardware lookup (extremely fast) of virtual page addresses. If page entry is not in TLB, a software search (slower) of all virtual page addresses must be done by the kernel to get the physical page address.

Performance Optimized Page Size

DAL

- HP-UX 11.0 introduces variable page size (aka. Large Pages or Performance Optimized Page Size).
- The SysAdmin can request a page size using the chatr() command or the kernel can specify a page size.
- The kernel tries to honor the request but may use a smaller page size if there is competition for memory.
- Kernel tunable parameters to control operation:
  - vps_pagesize — default page size used by kernel
  - vps_ceiling — maximum page size used by kernel
  - vps_chatr_ceiling — maximum size a user can set

  (probably don't want or need to change these tunables)

Advantages to Performance Optimized Page Size

DAL

- Larger virtual address ranges can be mapped using fewer TLB entries, so there will be fewer TLB misses
- POPS will offer performance advantages for applications that:
  - are experiencing significant TLB misses
  - have large Reference Sets (e.g. large Data Segments or Text Segments)

Performance Optimized Page Size Tools and Documentation

DAL

- chatr command (see chatr(1) man page)

  chatr +pi <size> text page size

  chatr +pd <size> data page size

  size = 4K, 16K, 64K, 256K, 1M, 4M, 16M, 64M, 256M, L (largest), D (kernel default)

- kernel tunable parameters:

  vps_pagesize default page size kernel will use

  vps_ceiling max page size kernel will select

  vps_chatr_ceiling max page size a user can select

- /usr/contrib/bin/vps_stats
  - report page statistics
- White paper: /usr/share/doc/mem_mgt.[ps,txt]

Performance Optimized Page Size Example Programs

DAL

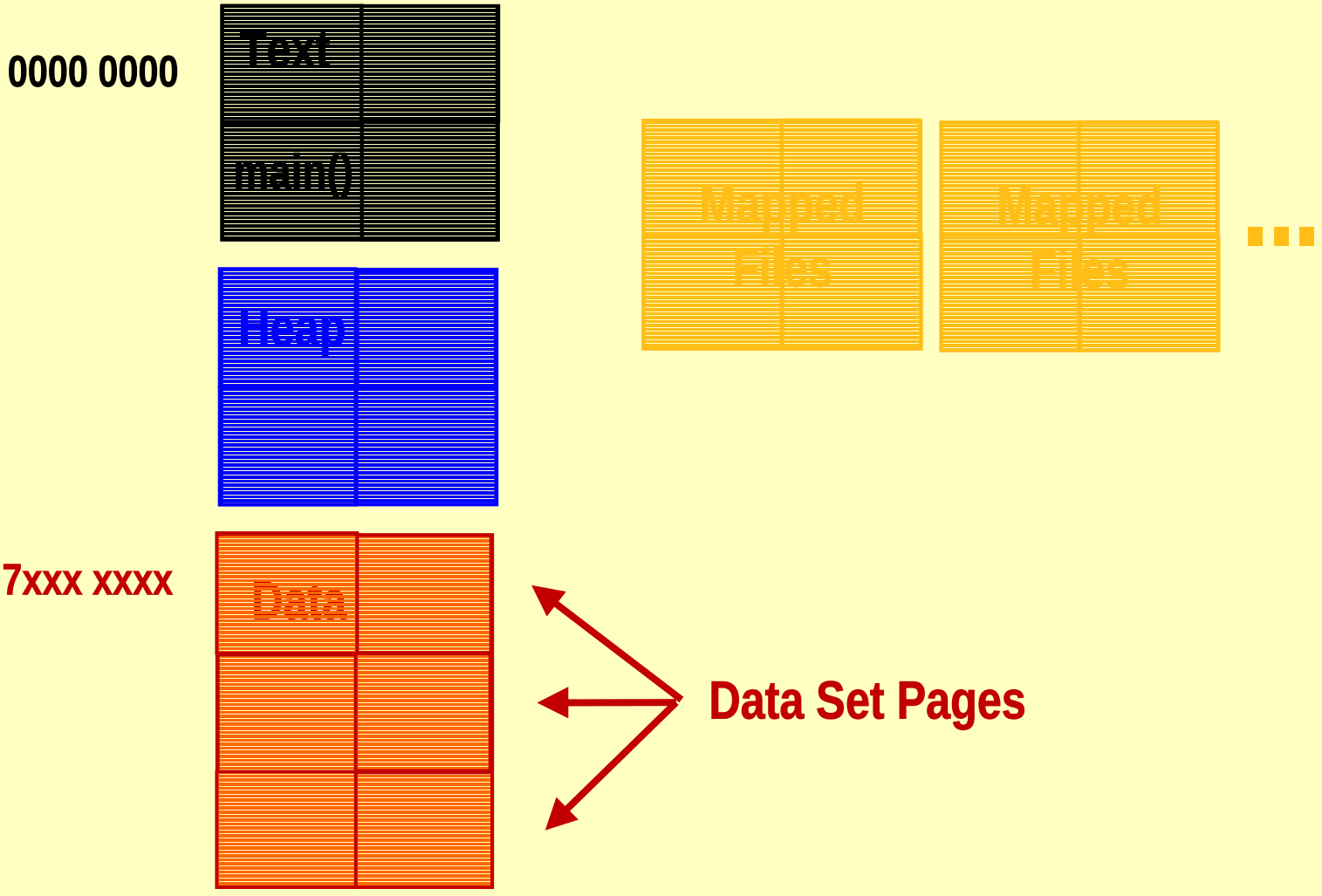
```
/*
 * Allocate/touch large data space - good locality
 */

#define SIZE 256
int big_array [SIZE] [SIZE] [SIZE];

main()
{
  int i,j,k;     /* array indices */

  for ( i=0; i<SIZE; i++ )
    for ( j=0; j<SIZE; j++ )
      for ( k=0; k<SIZE; k++ )
        big_array[i][j][k] = k;
}
```

65 MB Data Segment
16 Million 4-byte ints

Touch each element
16 Million iterations

Memory Map - Virtual Memory Pages

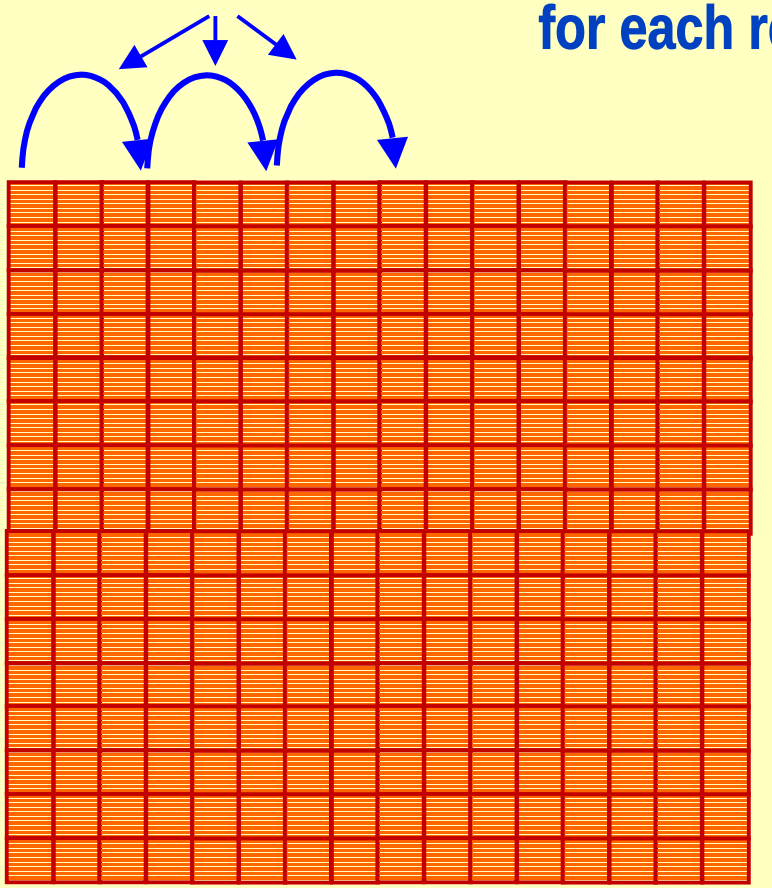
DAL



Example Program - Good Locality

DAL

GOOD Data Locality: 4 byte address jumps for each reference



Start Address
big_array(0,0,0)

65 MB Data Segment
16 Million 4-byte ints
(65 MB/pagesize) Pages

End Address
big_array(256,256,256)

Example Program - Poor Locality

DAL

```
/*
 * Allocate/touch large data space - poor locality
 */

#define SIZE 256
int big_array [SIZE] [SIZE] [SIZE];

main()
{
  int i,j,k;     /* array indices */

  for ( i=0; i<SIZE; i++ )
    for ( j=0; j<SIZE; j++ )
      for ( k=0; k<SIZE; k++ )
        big_array[k][j][i] = k;
}
```

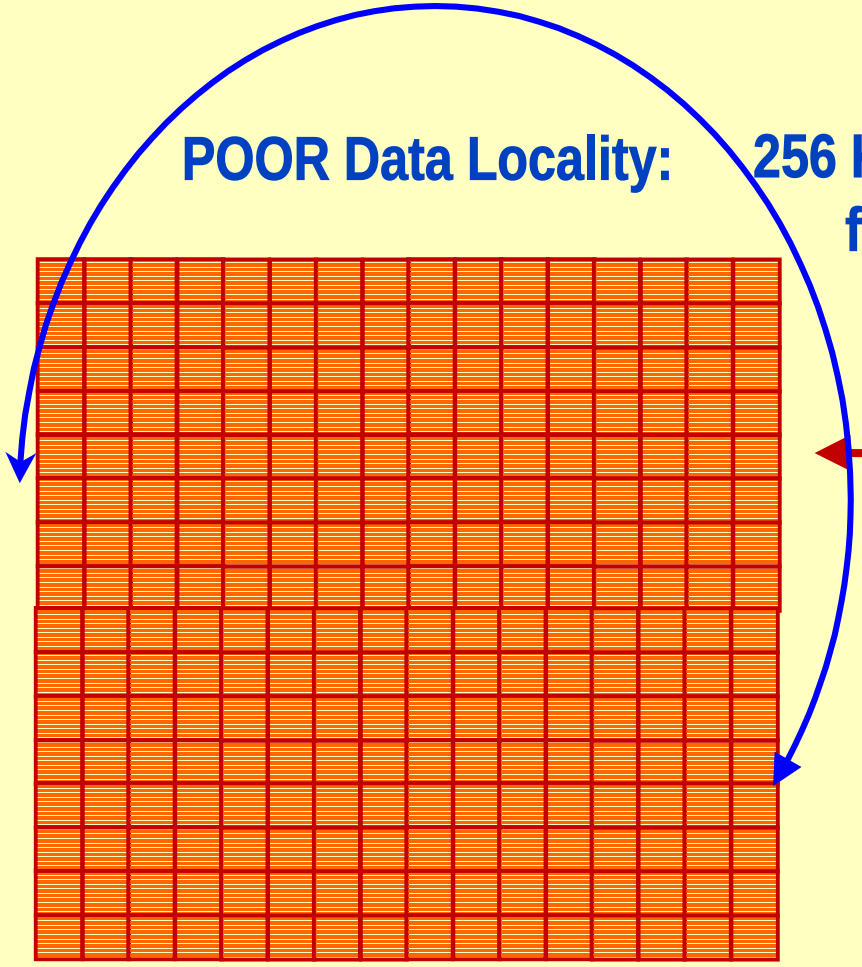
65 MB Data Segment
16 Million 4-byte ints

Poor Locality -
256K address jumps
for each reference

All we've done is reverse these indices!

Example Program - Poor Locality

DAL



Test Program
Default Compile and Run

DAL

- the default compile will set the page size to the kernel default page size:
  - chatr shows **"D"** for page size
- the default kernel tune is for 16K page size
- the Program Memory Map will consist of:
  - Text Segment: very small
  - Data Segment:

    4096 entries of 16 K byte pages = 65 M Bytes

Test Program Memory Map: vps_stats -p <process_id>

DAL

```
REGION INFO

TYPE     HINT   PHYSPAGES   Phys Page SIZE[Count]
----     ----   ---------   ---------------------

NULL     4K          1
TEXT     4K          3          4K[3]
DATA     16K         16387      4K[3]
                                16K[4096]
MMF      4K          0
MMF      4K          3          4K[3]
MMF      4K          1          4K[1]
MMF      4K          3          4K[3]
MMF      4K          4          4K[4]
MMF      4K          1          4K[1]
MMF      4K          2          4K[2]
MMF      4K          3          4K[3]
STACK    16K         3          4K[3]
MMF      4K          15         4K[15]
MMF      4K          206        4K[206]
MMF      4K          1          4K[1]
UAREA    4K          7          4K[7]
```

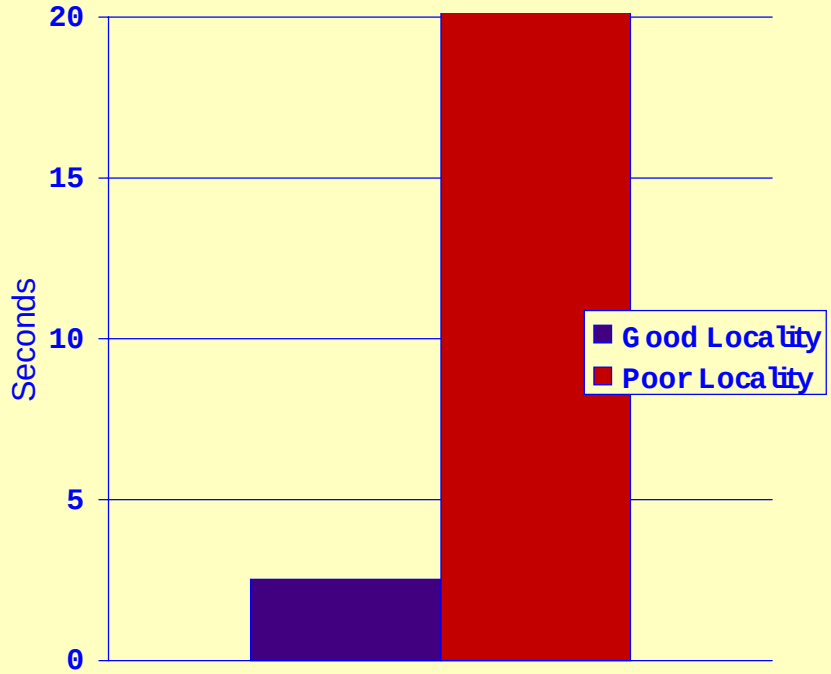
Data Segment:
16 KB pages
4096 entries
= 65 Mbytes space

Execution Times
Default (16 KB) pages

DAL

- Execution time: measure with "timex big_array"
- TLB misses: measure with "cyclemeter" (contributed tool)
  (PerfUX or GlanceUX also work)

- Good Locality program:
  Runtime 2.52 sec.
  Data TLB misses` 51,535
- Poor Locality program:
  Runtime 20.49 sec.
  Data TLB misses 17,062,436



Memory Map: chatr to 256K Data Pages

DAL

```
REGION INFO

TYPE     HINT    PHYSPAGES   Phys Page SIZE[Count]
----     ----    ---------   ---------------------

NULL     4K          1
TEXT     4K          3          4K[3]
DATA     256K        16387      4K[3]
                                16K[4]
                                64K[3]
                                256K[255]
MMF      256K        0
MMF      256K        3          4K[3]
MMF      4K          1          4K[1]
MMF      256K        3          4K[3]
MMF      4K          4          4K[4]
MMF      256K        1          4K[1]
MMF      256K        2          4K[2]
MMF      4K          3          4K[3]
STACK    256K        3          4K[3]
MMF      4K          15         4K[15]
MMF      4K          206        4K[206]
MMF      4K          1          4K[1]
UAREA    4K          7          4K[7]
```

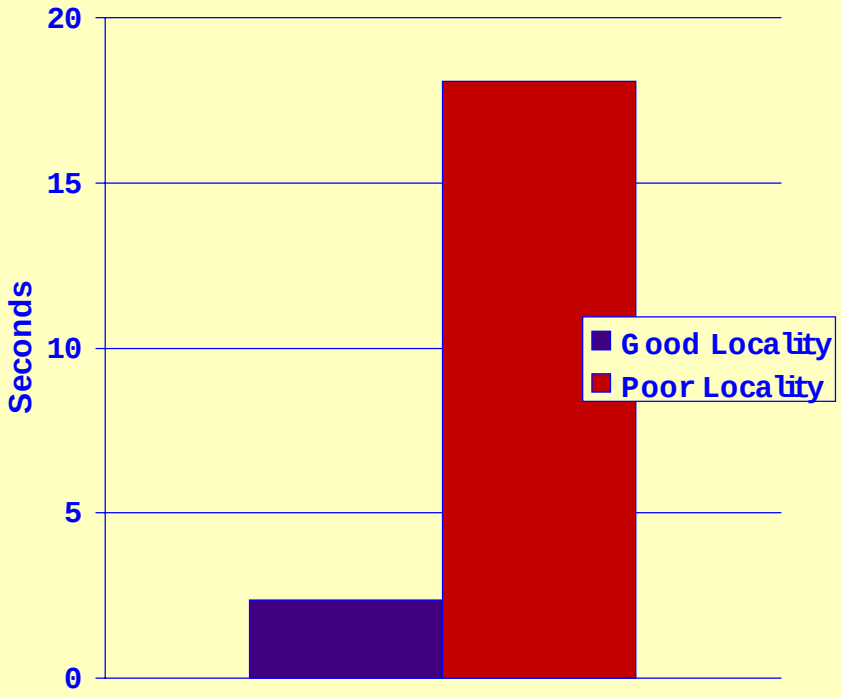
Data Segment:
256 KB pages
255 entries +
1 fragmented entry
= 65 Mbytes space

Execution Times chatr to 256 KByte pages

DAL

- Change Data Page size:
  "chatr +pd 256K big_array"

- Good Locality program :
  Runtime 2.35 sec.
  Data TLB misses` 34,562
- Poor Locality program :
  Runtime 18.06 sec.
  Data TLB misses 16,906,509



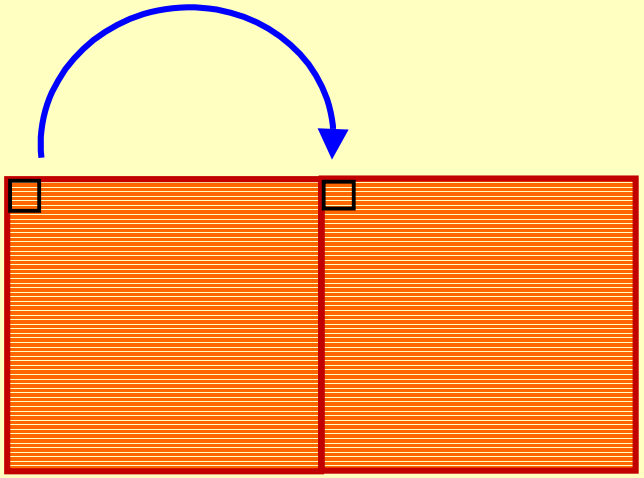
Change to 256 K byte Pages - why little performance change?

DAL

- Data Page Size is 256 K bytes
- "Poor Locality" C-program references 4-byte integers from address (A,B,C) to (A+65536,B,C),...
- CurrentAddress + (4 bytes * 65536) =

  CurrentAddress + 256K

  this is ***justover*** the next page boundary



Memory Map: chatr to 4M Data Pages

DAL

```
REGION INFO

TYPE     HINT    PHYSPAGES   Phys Page SIZE[Count]
----     ----    ---------   ---------------------

NULL     4K          1
TEXT     4K          3          4K[3]
DATA     4M          16387      4K[3]
                                16K[4]
                                64K[3]
                                256K[3]
                                1M[3]
                                4M[15]
MMF      4M          0
MMF      4M          3          4K[3]
MMF      4K          1          4K[1]
MMF      4M          3          4K[3]
MMF      4K          4          4K[4]
MMF      4M          1          4K[1]
MMF      4M          2          4K[2]
MMF      4K          3          4K[3]
STACK    4M          3          4K[3]
MMF      4K          15         4K[15]
MMF      4K          206        4K[206]
MMF      4K          1          4K[1]
UAREA    4K          7          4K[7]
```

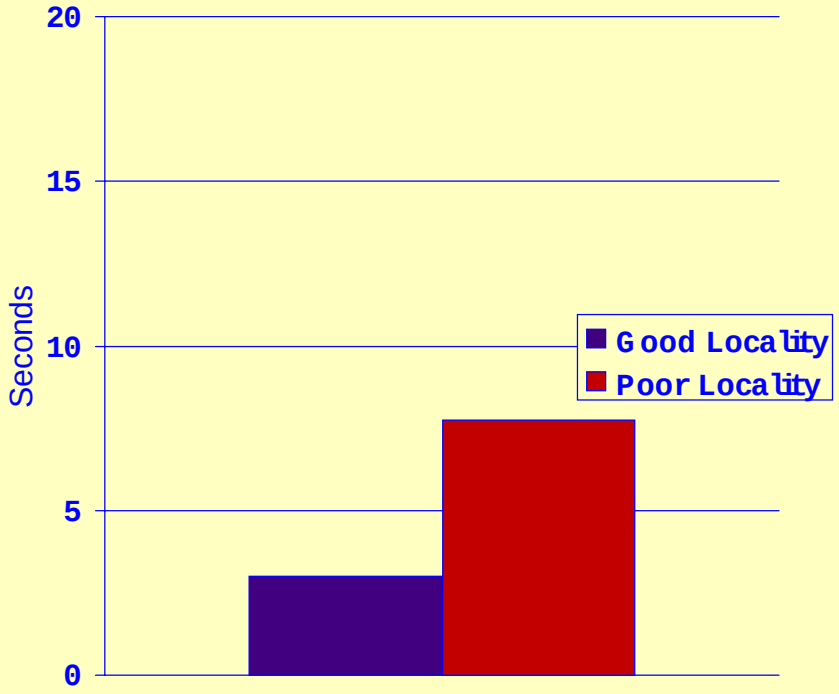
Data Segment:
4 MB pages
15 entries +
1 fragmented entry
= 65 Mbytes space

Execution Times chatr to 4 MByte pages

DAL

- Change Data Page size:
  "chatr +pd 4M big_array"

- Good Locality program :
  Runtime 2.99 sec.
  Data TLB misses` 33672
- Poor Locality program :
  Runtime 7.75 sec.
  Data TLB misses 30,227



Variable Pages - Other Considerations

DAL

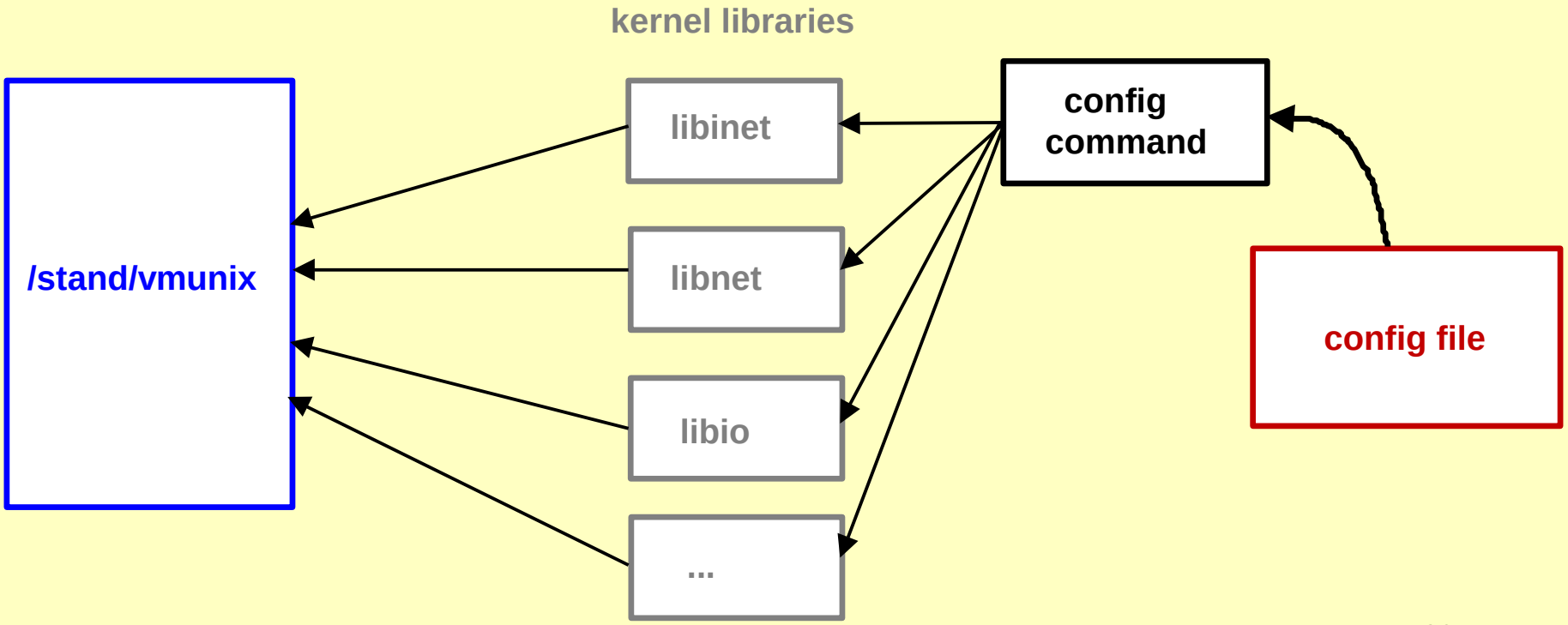
- Other factors affecting performance:
  - other ongoing system activity
  - memory page contention/thrashing/locking issues
  - cache sizes and cache fits

- Best performance is a combination of:
  - ✔ fast and efficient hardware (PA-8000)
  - ✔ correct system and kernel tunes
  - ✔ well-written software

Static Kernel Configuration

DAL

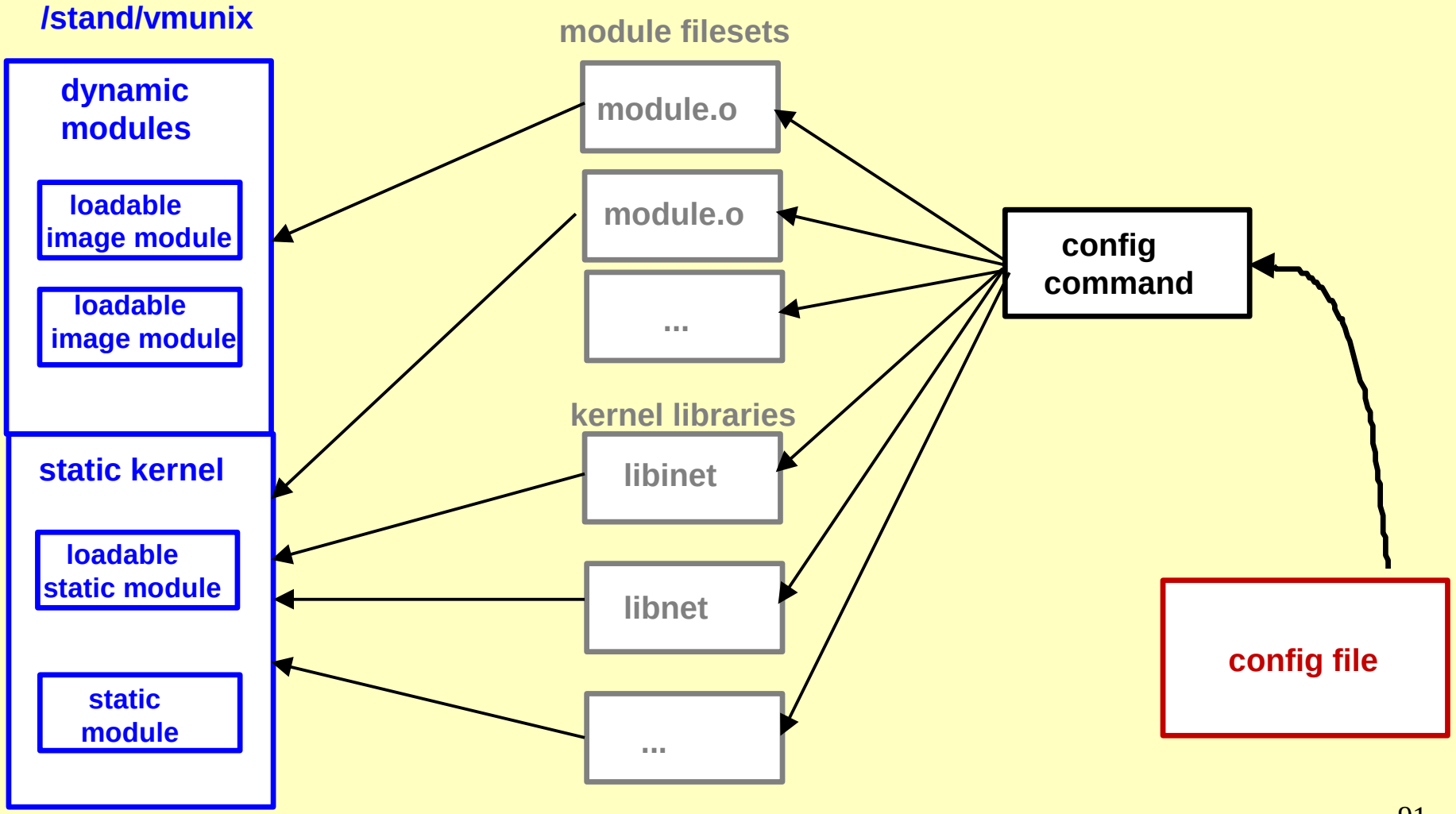
- Prior to 11.0 Release, HP-UX kernels have been statically configured and built
- config command to build kernel



Dynamic Kernel Configuration

DAL

Dynamically Loadable Kernel Modules allow one to load, unload, and configure kernel modules without rebooting



H

DLKM in 11.0

DAL

- DLKM Phase 1 delivered in 11.0:
  - DLKM infrastructure in kerneland commands
  - Major subsystems made DLKM-aware
  - no modules configured as dynamically loadable (yet)

Configuring Kernels

DAL

- Static kernel configuration remains exactly the same
  - config command and kernel config files
- To build a new kernel module:
  - config -M <module>
- To update the kernel with the new module:
  - config -u /stand/system -or-
  - kmupdate (new command)

New Commands for configuring Kernels

DAL

kmsystem	**set control flags in system files**
kmtune	**interface to set tunable parameters**
kmupdate	**update system with new kernel or loadable modules**
kminstall	**install/remove/update a module in a system**
kmmodreg	**register/unregister a module with the system**
kmadmin	**general administrative interface for DLKM**

DLKM - Future Capabilities

DAL

- On-Line Module Replacement
- Module support for more subsystems:
  - File systems
  - System calls
  - Other drivers

Fast Core Dump/Recovery

DAL

- Support of Large Memory (to 16 GBytes) means system image dump/recovery could take a long time
  - from several minutes to several hours
- Long dump/recovery times would not allow the High Uptime goals for HP-UX
  - one cannot afford to take several hours to dump/recover a system image
- Size of the dump image could be very large, and there may not be enough space to store the image

Fast Core Dump/Recovery

DAL

New/renamed commands:

crashconf	Add/remove/redefine dump devices at runtime. Devices can also be statically declared with a new field in /etc/fstab. This can override the dump device that has been configured in the kernel.
savecrash (savecore)	Save system image during the next reboot. Can select full/partial, or compress/nocompress
crashutil	Formats the crash dump image for further analysis

Fast Core Dump/Recovery

DAL

- Background mode operation of savecrash service at system startup
- There is a new mode flag to allow Background Operation available to startup scripts
  - Allows system startup to proceed while a lengthy process continues in background
  - Don't use this mode if other services depend on the immediate result of the startup of your service, or else the results may be confusing
- Background Mode startup will display "OK" in the startup screen if the service begins successfully

Fast password/uid/gid Lookup

DAL

- Extension to uid/gid/password lookup for local files.
  - applies partially to NIS systems
  - does not apply to Secure systems
- Large numbers of uids (to 4G.) means potentially large number of entries in /etc/passwd file.
- Existing APIs (getpwent(3c), getgrent(3c)) do a linear search through /etc/passwd - potentially long times required for simple login or lookup

Fast Passwords

DAL

- Solution to lengthy searches of password files
  - a helper process to speed the lookup
- New system process: **pwgrd** - a user space daemon
- All supported APIs such as getpwent(3C) and getgrent(3C) communicate to **pwgrd**
- **pwgrd** hashes and caches password entries to avoid potentially lengthy searches of */etc/passwd*

FastPasswords

DAL

- **pwgrd** fast lookup
  - hashing of all entries in */etc/passwd* - done at startup of the daemon or if the daemon is signaled
  - caching of recent lookups into */etc/passwd*
- **pwgrd** caches recent retrievals from NIS entries
- Documentation:
  - **pwgrd**(1M) man page on 11.0 system

Ignite-UX

DAL

- A new Cold Installation method
  - not an Update in Place method
- Available on 10.X (starting with DART 34) and 11.0

Ignite-UX Capabilities

DAL

	Cold Install	Ignite-UX
Install methods	From local media Pull from network server	From local media Pull from network server Push from network server Reinstall existing machines
Server capability	Single Release	Multiple Releases
User Interface	TUI on target	TUI on target GUI on server
Software Formats	Software Distributor	Software Distributor, cpio, tar can use multiple formats

Ignite-UX

DAL

- Install methods:
  - Boot from local media
  - Pull from network server
  - Push from network server

- uses rem sh to target system console
  - can easily add new clients

Ignite-UX
Creating Custom Install Packages

DAL

- Customize an install configuration and save the configuration for replication
  - "Save As" option to Ignite-UX

- Ignite-UX tools allow one to create a custom archive (Golden Image) and choose the install format

System Recovery

DAL

- Use to recover Root Filesystem with all your installation and customization

- Root recovery might ordinarily require these steps:
  - cold install
  - configure
  - reinstall patches
  - reinstall applications
  - reinstall user data and files

System Recovery - other uses

DAL

- Modify Root Filesystem size
- Modify primary swap size
- Convert Root Filesystem from HFS to VxFS
- Clone a system

System Recovery

DAL

- Supported on 11.0 and 10.X
- Installed with Ignite-UX
- Install from DART releases or HP Web site:
  - http://www.software.hp.com/
  - Link to: "Network and System Administration Tools"

System Recovery

DAL

make_recovery	make a system recovery tape ("make_recovery -A" - entire core VG)
save_config	create a configuration file that details the current system hardware and software configuration
check_recovery	compare current system configuration to last configuration file
print_manifest	print hardware configuration (CPUs, LVM and disks, I/O), OS configuration, installed products)

System Recovery

DAL

- Currently supports:
  - HFS/VxFS
  - SAM interface
  - LVM
  - DDS tape recovery image
- Not yet supported:
  - Striped or mirrored disks
  - Shared LVM

System Recovery - print_manifest utility

DAL

```
System Information
    Your Hewlett-Packard 9000 computer has software installed and configured as follows.

---------------------------------------------------------------
NOTE: You should retain this information for future reference.
---------------------------------------------------------------

System Hardware
    Model:              9000/777/C110
    Main Memory:        128 MB
    Processors:         1
    OS mode:            32 bit
    HW capability:      32 bit
    LAN hardware ID:    0x0060B001BA5B
    Software ID:        2011905808
    Keyboard Language:  PS2_DIN_US_English

    Storage devices                 HW Path      Interface
    SEAGATE ST15150W 4095 Mb        8/12.6.0     GSC built-in Fast/Wide SCSI Interface
    SEAGATE ST32430N 2048 Mb        8/16/5.4.0   Built-in SCSI
    HP C1533A                       8/16/5.3.0   Built-in SCSI

    I/O Interfaces
    Class          H/W Path       Driver         Description
    ext_bus        8/12           c720           GSC built-in Fast/Wide SCSI Interface
    ext_bus        8/16/0         CentIf         Built-in Parallel Interface
    audio          8/16/1         audio          Built-in Audio
    tty            8/16/4         asio0          Built-in RS-232C
    ext_bus        8/16/5         c720           Built-in SCSI
    lan            8/16/6         lan2           Built-in LAN
    ps2            8/16/7         ps2            Built-in Keyboard/Mouse
    pc             8/16/10        fdc            Built-in Floppy Drive
    hil            8/20/1         hil            Built-in HIL
    tty            8/20/2         asio0          Built-in RS-232C
    graphics       10/16          graph3         Graphics


Installed Software
    Your system was installed with HP-UX version B.11.00.

    Your system has the following software products installed and
    configured on the system disk drive(s).

    Product          Revision         Description
    B3782EA          B.10.20          HP-UX Media Kit (Reference Only. See Description)
    B3884EA_AGN      B.10.20          HP-UX 32-User License
    B3899BA          B.11.01.01       HP C/ANSI C Developer's Bundle for HP-UX 11.00 (S700)
    B3911DB          B.11.01.01       HP aC++ Compiler  (S700)
    B3919EA_AGS      B.11.00          HP-UX Unlimited-User License
    B4580AA          B.11.00.01       HP-UX 11.00 Software Transition Kit
    B5455CA          C.01.16.00       HP-UX Development Kit for Java*
    B5724AA_APZ      A.1.45           HP-UX Installation Utilities (Ignite-UX - S700 - 10.20)
    DCEProg          B.10.20          DCE Programming and Archive Libraries
    HPUXEngCR700     B.10.20          English HP-UX CDE Runtime Environment
    J2559C           D.06.15          Hewlett-Packard JetAdmin for Unix Utility
    UXCoreMedia-J    B.11.00          HP-UX Japanese Media Kit (Reference Only. See Description)
    XSWGR1100        B.11.00.39       HP-UX Extension Pack, June 1998
```

HARDWARE CONFIGURATION

INSTALLED SOFTWARE

System Recovery - print_manifest (continued)

DAL

```
LVM File System Configuration
    This system is configured with Logical Volume Manager (LVM) file systems.
    Refer to the File System layout section for information on the LVM layout.


Disk layout
    LVM disk                  Device file        HW Addr      size   vol. grp
    SEAGATE ST15150W          /dev/dsk/c0t6d0    8/12.6.0     4095   /dev/vg00
    non-LVM disk              Device file        HW Addr      size   swap
    SEAGATE ST32430N          /dev/dsk/c1t4d0    8/16/5.4.0   2003   0


File System layout
    LVM Device file           mount point    size  fs type
    /dev/vg00:
    /dev/vg00/lvol3           /              84    hfs
    /dev/vg00/lvol2           swap           256
    /dev/vg00/lvol1           /stand         48    hfs
    /dev/vg00/lvol7           /usr           700   hfs
    /dev/vg00/lvol5           /opt           1000  hfs
    /dev/vg00/lvol8           /var           160   hfs
    /dev/vg00/lvol6           /tmp           32    hfs
    /dev/vg00/lvol4           /home          1800  hfs
    /dev/vg00                 unallocated    12
    Device file                 mount point       size      fs type
    /dev/dsk/c1t4d0             /mnt/c1t4d0       2003      hfs


Swap configuration
    type       size  priority  device/location
    dev        256      1      /dev/vg00/lvol2


Kernel Configuration
    The following drivers or parameters are configured into your system's
    kernel.  After installing HP-UX, use the sam(1m) command to configure
    the following items into the kernel:
     STRMSGSZ                65535
     default_disk_ir             1
     maxdsiz              0X10000000
     maxssiz              0X10000000
     maxtsiz              0X10000000
     nstrpty                    60


System Information
    The following parameters were set on the configured target:
     hostname:            hp46t250
     IP address:          15.14.120.250
     subnet mask:         255.255.248.0
     gateway IP address:  15.14.120.250
     time zone:           PST8PDT
     DNS domain name:     cup.hp.com
     DNS IP address:      15.13.185.120
     DNS IP address:      15.13.192.134
```

DISK AND LVM LAYOUT

KERNEL CONFIGURATION

NETWORK CONFIGURATION

HP-UX 11.0 Networking

DAL

- Streams-based TCP/IP
- NFS PV3
- NIS+
- IP Aliasing/VirtualHosting
- Web Server Support
- Java Support

Streams-based TCP/IP

DAL

- At 10.30 the TCP/IP stack was changed to streams-based technology, replacing the BSD-UNIX base
- Very few changes to users or APIs
- Advantages in MP and supportability
- Visible to user as str* kernel daemons

NFS Protocol Version 3 Advantages

DAL

- Added Support for Large Files
- Improved Performance
- Enhanced File Access control
- New APIs

- Delivered in 10.30 and 11.0 Releases
- Workstation ACE2 Release for 10.20 OS

NFS 3 Support for Large Files

DAL

	NFS Version 2	NFS Version 3
File Size and Offsets	32 bits	64 bits
Maximum File Size Supported	2 GBytes	128 GBytes

NFS PV3 Performance Improvements

DAL

- Function calls now return attributes to reduce subsequent getattr() function calls
- Read/write blocks can be larger than the previous 8 Kbyte limit
- Performance of asynchronous write operations much improved: write request from client returns immediately, commit request (NEW) from client causes update to disk on server
- Weak cache consistency: if client modification time matches server modification time the client cache is assumed to be valid
- Remains fully interoperational with NFS V2

NIS+

DAL

- Improved performance and security over NIS
- Continues to support NIS and local file access
- Capability to add future services on line without rebooting or reconfiguring
- Delivered in the 10.30 and 11.0 Releases

NIS+ vs. NIS
Comparison of Features

DAL

	NIS	NIS+
Map updates (Master to Slaves)	Entire map updated	Incremental changes updated
Update propagation	Manual	Automatic
Data Access Restrictions	None	Access controlled on per-entry basis
Authentication	None	Secure RPC
Administration	Must be done on single Master Server	May be made within hierarchical namespace
Namespace	Flat	Hierarchical
Contacting Servers	UDP broadcast	coldstart config file and directory cache

IP Address Aliasing and VirtualHosting

DAL

- ifalias command on 10.20
  - allows multiple IP addresses on same interface (IP Address Aliasing)
  - (install appropriate kernel and commands patches)
- For 10.30 and 11.0, new options to the ifconfig command supply the same functionality
- ifalias is obsoleted at 10.30 and 11.0

HP-UX Internet Products

DAL

- Netscape FastTrack Server 3.01 bundled with HP-UX
- Netscape SuiteSpot available as separate product:
  - Enterprise Server 3.6 (web server)
  - Messaging Server 3.5 (mail)
  - Directory Server 4.0 (LDAP services)
  - Certificate Server
  - Calendar Server
  - Proxy Server
  - Admin Server 3.5
- Zeus Web Server 3.3
- New HP-UX/I Internet Bundle of products

Java Support on HP-UX

DAL

- Current releases: JDK 1.1.8 and 1.2
  - contains JIT compiler, JVM, class libraries
  - 10.20: uses Java internal (Green) threads
  - 11.0: uses HP-UX kernel threads or Java threads
- JVM 1.2 available in both HotSpot and Classic versions

Java Development on HP-UX

DAL

Java on HP-UX

DAL

- Java Web Page for HP-UX
  - http://www.hp.com/go/java
- Can download the Java Development Kit
- Additional documentation available
- Pointers to Java Reference Documents

Additional Networking Enhancements

DAL

- New FTP (June 1998 - Patch PHNE_14479)
  - improved logging, security, on the fly compression
- Kerberos v5 1.0
  - Provides encryption and authentication; simplified installation
- DNS 4.9.6
  - Eases load balancing through round robin
- Sendmail 8.8.6
  - Prevents system overload with anti-spamming feature
- BIND 4.9
  - Improves response times and enhances security
- Gateway Daemon 3.5.1 (gated)
  - Guaranteed connectivity

HP-UX 11.0 Extension Software

DAL

Extension Software/SupportPlus releases are patch bundles consisting of a set of recommended HP-UX OS patches:

Release	Date	OS's Refreshed	Part Number
SP47	December 99	10.20 11.00	B3782-10453
SP46	September 99	10.20 11.00	B3782-10438
XR45	June 99	10.10 10.20 11.00	B3782-10418
XR44	April 99	10.10 10.20 11.00	B3782-10386
XR43	February 99	10.10 10.20 11.00	B3782-10371
XR41	October 98	10.10 10.20 11.00	B3782-10356
XR40	August 98	10.10 10.20 11.00	B3782-10340
XR39	June 98	10.10 10.20 11.00	B3782-10346
XR38	April 98	10.10 10.20 11.00	B3782-10300
XR37	February 98	10.01 10.10 11.00	B3782-10293

HP-UX i11
Instant Ignition Bundle

DAL

Core Products:

- Netscape Fasttrack Server
- Web QoS Peak Service
- JAVA (JRE, JDK, currently instantly ignitable)
- CDSA Framework
- Network Drivers (all currently instantly ignitable)
- Ignite/UX
- EMS
- Kernel optimizations for web performance

HP-UX i11 ProductBundle

DAL

Optional Products:

- Manageability — OpenView, Service Control WebQoS Premium
- Web Servers — Netscape, Zeus, Apache
- Availability — MC/ServiceGuard
- Security — Praesidium
- App. Dev. — BEA
- Services — Pre and Post sales