Soul List Optimisation: Implementation Notes

This PR was created because of the soul list getting too long. The primary problem relating to the free lists was two fold.

a) the free list was getting too long
b) we took too long to traverse a long list.

Earlier changes addressed (b). And now we can handle a long free list much better.

Revision 1:

The current changes address the way the soul allocations work.

In the existing code base, the way the soul allocations work, we allocate a certain number (minimum) up front on the first pass through the object allocator for a given soul, and then upon demand grow the souls by one item at a time. We kept track of a high water, recent used counter etc, to allow us to release souls that are just "sitting around". We also had some code to potentially allow us to "garbage collect" from the idle thread (called object_compact).

The main issue was not because we allocated the various soul objects (which are each fixed in size, but are all small items differing from other souls) by going to the system free list (ker or proc) by calling scalloc.

This meant that after the initial growth of the souls, we would go scalloc with sizes that are "1" soul at a time. For pulses and sync objects for example, this would be the 24 bytes and 16 bytes etc. This meant that remainders would get retained on the free list, creating various holes, which wont necessary be filled unless things aligned up well in terms of the objects being freed.

These new changes alter the way we do the soul allocations. We go to scalloc with a request size of (BSIZ = 4096 bytes). This is then carved up into as many pieces as each soul sized object would allow us to fit into this block.

The soul_entry in public/kernel/objects.h has been altered as follows.

The original soul_entry was as follows

```
struct soul_entry {
  void      *next;
  int32_t      type;    // Type of soul
  uint32_t     total;   // Total allocated(free or in use).
  uint32_t     used;    // Number in use.
  uint32_t     size;    // Size of object in the list.
  uint32_t     newmin;  // Current min we need.
  uint32_t     min;     // Lowest min allowed.
  uint32_t     highwater;
  uint16_t     counter;
  uint8_t      align;   // Alignment requirement on soul
  uint8_t      flags;   // Soul flags
};
```

We removed the next, newmin, highwater, counter fields. Instead we added some fields, so that new soul_entry look like this.

```
struct soul_entry {
  int32_t      type;    // Type of soul
  uint32_t     total;   // Total allocated (free or in use).
  uint32_t     used;    // Number in use.
  uint32_t     size;    // Size of object in the list.
  uint32_t     min;     // Lowest min allowed.
  uint8_t      align;   // Alignment requirement on soul
  uint8_t      flags;   // Soul flags
  uint16_t     incr_size; // number of soul entries per block allocation
  void         *clist_a;  // available block allocation list
  void         *clist_d;  // depleted  block allocation list
};
```

On first pass through the object_allocator, we calculate incr_size to be the number of items that will fit into the chunk size (BSIZ=4096). (in object_grow)

Earlier when object_alloc was called, we would look at soulp->next to find the next available block. Now we look in soulp->clist_a and if this is empty we call scalloc to create a new "block", and if it is not empty, we look at soulp->clist_a->alist. If this is empty, we clip off the next available soul (using the macro NEXT_AVAILABLE_SOUL_BLK in nano_object.c)

On object_free, we release the block to the appropriate chunk, and when the chunk is empty, we release it back to the system free list using sfree (as a 4096 byte chunk).

To allow us to service critical allocations earlier we used to call scalloc with alloc_critical if SOUL_CRITICAL is set. Now we try and allocated a chunk without alloc_critical, and if this fails, but SOUL_CRITICAL is set, we retry the allocation by looking for a single soul sized piece by calling scalloc with alloc_critical set, (this is in object_alloc). When soul pieces from critical memory are released, we free them immediately back to the critical pool.

Revision 2:

Based on reviews in the core OS forum, the following changes were implemented.

We've inlined what were originally macros, and I've made the code go directly to mmap for 4k sized pieces, and these are released back to the system as soon as we empty a "chunk". This way memory is reclaimed back as soon as its possible, based on the configured minimums for the specific soul.

One additional thing that was modified was to convert the ocb allocations in devmem.c to being ocb_soul allocations instead of calling into scalloc each time (the ocb is 16 bytes in length, so I've created a ocb_soul in devmem, and we go through the proc_object_alloc interface).

Looking at the results that are attached, as expected we are down to nearly a flat performance curve (the test before and after the "fragmentation" runs at the same pace. (this is expected, since the fragmentation now really doesnt fragment anymore).

Before and after the test, the amount of memory used is compared, and we actually use less memory than in stock code today (again this is relevant to illustrate that we are reclaiming memory from the heap). We run faster based on simple "time" measurements. tests like ker_pr10963 and wh1_SignalKill reclaim _all_ memory used in the tests, rather than the earlier versions which would still have memory left on the heap or in the various souls themselves. ker_pr10963 for example is able to complete _many_ more iterations in its run that earlier.

The SOUL_NOFREE and SOUL_PROC flags have been removed, and eventually we can remove the SOUL_CRITICAL also, and maybe even the flag field altogether.

Revision 3:

With the advent of the soul changes we started using the soul allocation code for more and more things. We started mapping, unmapping directly, and in cases where we were low on memory, we would call into scalloc, sfree.

This means that all proc allocations obtained the proc_object_lock, in proc/support.c, which then would funnel into object_alloc, and then if we called into scalloc, it would obtain the scalloc version of the proc_alloc_mutex, (now holding two locks).. it then could call mmap, which it would do releasing the proc_alloc_mutex and so on.

object_alloc obtains the proc_object_mutex in the object_alloc code if it is a proc thread. When we call _out_ from object_alloc code to mmap, munmap, scalloc, smalloc, sfree etc, we release the proc_object_mutex, and re-acquire upon return if necessary. This means that we never hold more than one mutex at a time. The more places we call into the object_alloc code the more likely there could be problems. With the soul changes we are calling into the allcoator code more often than before (more places, and we call unmap too)

Testing (Full testing results in Appendix)

wh1_freelist

there are 3 different conditions

a) stock head 6.4.2
b) these changes on top of 6.4.2
c) these changes with also converting the memref's in mm_memref.c to be allocated from a soul instead of going directly to scalloc themselves.

We ran the changes on a qemu based system, and also a real x86 4 way (2x2 hyper-threaded smp box).

As one can see with just these changes, the ker list fragmentation is almost non-existent, and with the memref changes also, the proc list fragmentation has reduced by an order of magnitude.

We also ran wh1_SignalKill and ker_pr10963 as described in this PR (cannot use the instrumentation described, because the free list code has changed :)). ker_pr10963, runs significantly faster on both qemu and on real hardware.

Appendix

==========
Stock 6.4.2 (head) on QEMU
==========
wh1_freelist

pidin in
CPU:X86 Release:6.4.2 FreeMem:124Mb/127Mb BootTime:Oct 22 14:57:08 UTC 2009
Processes: 4, Threads: 15
Processor1: 686 Pentium II Stepping 3 1827MHz FPU
uname -a
QNX localhost 6.4.2 2009/10/22-10:50:31UTC x86pc x86
time wh1_freelist -p -vv
START: wh1_freelist, pid 16388
NOTE: file regress/ker/wh1_freelist.c, line 1177 initial run of tests to stabilize system...
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
P_list 7, entries 797, min 0, max 9480
P_mmap ---, free 4723056

P_slot	count	memory	maxrun	avgrun	first
P_0	8	64	2	1	777
P_16	2	32	2	2	778
P_32	11	376	2	1	105
P_64	6	472	5	3	21
P_128	0	0	0	0	4294967295
P_256	81	43160	1	1	5
P_1024	689	4678952	689	689	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
P_list 7, entries 13606, min 0, max 9480
P_mmap ---, free 514824

P_slot	count	memory	maxrun	avgrun	first
P_0	108	864	2	1	15
P_16	12476	298720	70	14	2
P_32	478	22040	11	2	0
P_64	318	23928	262	6	14
P_128	7	1448	2	1	10490
P_256	203	65672	201	67	10505
P_1024	16	102152	16	16	13569

NOTE: file regress/ker/wh1_freelist.c, line 1042 fragmenting KER free list.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
K_list 6, entries 88, min 0, max 326944
K_mmap ---, free 969816

K_slot	count	memory	maxrun	avgrun	first
K_0	4	32	2	1	79
K_16	57	1360	28	14	3
K_32	8	304	3	1	0
K_64	3	376	2	1	45
K_256	2	640	1	1	2
K_1024	14	967104	14	14	1

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
K_list 6, entries 6810, min 0, max 147024
K_mmap ---, free 808128

K_slot	count	memory	maxrun	avgrun	first
K_0	4	32	2	1	6801
K_16	6356	152536	48	15	1
K_32	419	13728	411	69	16
K_64	17	1408	15	5	0
K_256	1	320	1	1	6802
K_1024	13	640104	13	13	6765

NOTE: file regress/ker/wh1_freelist.c, line 1196 running 'after' benchmarks...
NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
P_list 7, entries 13606, min 0, max 9480
P_mmap ---, free 514824

P_slot	count	memory	maxrun	avgrun	first
P_0	108	864	2	1	15
P_16	12476	298720	70	14	2
P_32	478	22040	11	2	0
P_64	318	23928	262	6	14
P_128	7	1448	2	1	10490
P_256	203	65672	201	67	10505
P_1024	16	102152	16	16	13569

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
P_list 7, entries 797, min 0, max 9480
P_mmap ---, free 4723024

P_slot	count	memory	maxrun	avgrun	first
P_0	8	64	2	1	777
P_16	2	32	2	2	778
P_32	11	376	2	1	105
P_64	6	472	5	3	21
P_128	0	0	0	0	4294967295
P_256	81	43160	1	1	5
P_1024	689	4678920	689	689	0

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
K_list 6, entries 6811, min 0, max 147024
K_mmap ---, free 808104

K_slot	count	memory	maxrun	avgrun	first
K_0	4	32	2	1	6802
K_16	6358	152584	48	15	0
K_32	419	13728	411	69	17
K_64	16	1336	15	8	6780
K_256	1	320	1	1	6803
K_1024	13	640104	13	13	6766

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
K_list 6, entries 92, min 0, max 319528
K_mmap ---, free 969816

K_slot	count	memory	maxrun	avgrun	first
K_0	4	32	2	1	83
K_16	31	736	17	6	0
K_32	10	400	3	1	17
K_64	29	2272	15	9	33
K_256	3	1456	1	1	30
K_1024	15	964920	15	15	29

NOTE: file regress/ker/wh1_freelist.c, line 1205 comparing the results...
POINT: file regress/ker/wh1_freelist.c, line 657 SHM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SHM - 838.871ms -> 1067.836ms: change (after/before) = 1.27
POINT: file regress/ker/wh1_freelist.c, line 657 SEM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SEM - 427.934ms -> 609.906ms: change (after/before) = 1.43
POINT: file regress/ker/wh1_freelist.c, line 657 CHANNEL - threshold 8.000000
PASS: file regress/ker/wh1_freelist.c, line 672 CHANNEL - 605.907ms -> 1525.766ms: change (after/before) = 2.52
POINT: file regress/ker/wh1_freelist.c, line 657 TIMER - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 TIMER - 1033.841ms -> 2247.656ms: change (after/before) = 2.17
POINT: file regress/ker/wh1_freelist.c, line 1209 Total testcase performance
PASS: file regress/ker/wh1_freelist.c, line 1213 passed.
STOP: wh1_freelist, pid 16388 point=5,pass=5,fail=0,xpass=0,xfail=0,unres=0
81.05s real 22.20s user 14.83s system
#

==========
Modified 6.4.2 with soul changes on QEMU
==========

****
Without Memmrefs as SOULS
****

pidin in
CPU:X86 Release:6.4.2 FreeMem:124Mb/127Mb BootTime:Oct 22 14:51:52 UTC 2009
Processes: 4, Threads: 14
Processor1: 686 Pentium II Stepping 3 1823MHz FPU
uname -a
QNX localhost 6.4.2 2009/10/22-10:45:08EDT x86pc x86
time wh1_freelist -p -vv
START: wh1_freelist, pid 16388
NOTE: file regress/ker/wh1_freelist.c, line 1102 PR62953 -- fragmented kernel/proc free lists.
P_list 7, entries 800, min 0, max 9584

P_mmap ---, free 4724936

P_slot	count	memory	maxrun	avgrun	first
P_0	3	24	2	1	790
P_16	5	112	3	2	777
P_32	6	240	2	1	130
P_64	11	800	2	1	214
P_128	1	136	1	1	417
P_256	81	41416	2	1	11
P_1024	693	4682208	693	693	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
P_list 7, entries 13532, min 0, max 8480
P_mmap ---, free 516704

P_slot	count	memory	maxrun	avgrun	first
P_0	86	688	2	1	16
P_16	12386	296592	66	13	2
P_32	499	22624	13	2	0
P_64	324	24136	253	4	31
P_128	9	1704	2	1	10371
P_256	209	67112	206	69	10386
P_1024	19	103848	19	19	13492

NOTE: file regress/ker/wh1_freelist.c, line 1042 fragmenting KER free list.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
K_list 6, entries 29, min 0, max 294656
K_mmap ---, free 974816

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	16
K_16	1	16	1	1	25
K_32	6	208	3	1	13
K_64	2	288	2	2	14
K_256	0	0	0	0	4294967295
K_1024	18	974288	18	18	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
K_list 6, entries 32, min 0, max 155216
K_mmap ---, free 659424

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	19
K_16	1	16	1	1	28
K_32	6	208	3	1	16
K_64	2	288	2	2	17
K_256	0	0	0	0	4294967295
K_1024	21	658896	21	21	0

NOTE: file regress/ker/wh1_freelist.c, line 1196 running 'after' benchmarks...
NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
P_list 7, entries 13532, min 0, max 8480
P_mmap ---, free 516704

P_slot	count	memory	maxrun	avgrun	first
P_0	86	688	2	1	16
P_16	12386	296592	66	13	2
P_32	499	22624	13	2	0
P_64	324	24136	253	4	31
P_128	9	1704	2	1	10371
P_256	209	67112	206	69	10386
P_1024	19	103848	19	19	13492

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
P_list 7, entries 800, min 0, max 9584
P_mmap ---, free 4724904

P_slot	Count	memory	maxrun	avgrun	first
P_0	3	24	2	1	790
P_16	5	112	3	2	777
P_32	6	240	2	1	130
P_64	11	800	2	1	214
P_128	1	136	1	1	417
P_256	81	41416	2	1	11
P_1024	693	4682176	693	693	0

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
K_list 6, entries 33, min 0, max 155216
K_mmap ---, free 667616

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	20
K_16	1	16	1	1	29
K_32	6	208	3	1	17
K_64	2	288	2	2	18
K_256	0	0	0	0	4294967295
K_1024	22	667088	22	22	0

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
K_list 6, entries 30, min 0, max 294656
K_mmap ---, free 983008

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	17
K_16	1	16	1	1	26
K_32	6	208	3	1	14
K_64	2	288	2	2	15
K_256	0	0	0	0	4294967295
K_1024	19	982480	19	19	0

NOTE: file regress/ker/wh1_freelist.c, line 1205 comparing the results...
POINT: file regress/ker/wh1_freelist.c, line 657 SHM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SHM - 832.872ms -> 1040.840ms: change (after/before) = 1.25
POINT: file regress/ker/wh1_freelist.c, line 657 SEM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SEM - 432.933ms -> 599.908ms: change (after/before) = 1.39
POINT: file regress/ker/wh1_freelist.c, line 657 CHANNEL - threshold 8.000000
PASS: file regress/ker/wh1_freelist.c, line 672 CHANNEL - 589.909ms -> 590.909ms: change (after/before) = 1.00
POINT: file regress/ker/wh1_freelist.c, line 657 TIMER - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 TIMER - 528.919ms -> 526.919ms: change (after/before) = 1.00
POINT: file regress/ker/wh1_freelist.c, line 1209 Total testcase performance
PASS: file regress/ker/wh1_freelist.c, line 1213 passed.
STOP: wh1_freelist, pid 16388 point=5,pass=5,fail=0,xpass=0,xfail=0,unres=0
75.59s real 22.74s user 8.27s system
pidin in
CPU:X86 Release:6.4.2 FreeMem:118Mb/127Mb BootTime:Oct 22 14:51:52 UTC 2009
Processes: 4, Threads: 14
Processor1: 686 Pentium II Stepping 3 1823MHz FPU
#

****
With Memmrefs as SOULS
****

pidin in
CPU:X86 Release:6.4.2 FreeMem:124Mb/127Mb BootTime:Oct 22 15:02:12 UTC 2009
Processes: 4, Threads: 14
Processor1: 686 Pentium II Stepping 3 1824MHz FPU
uname -a
QNX localhost 6.4.2 2009/10/22-11:01:29EDT x86pc x86
time wh1_freelist -p -vv
START: wh1_freelist, pid 32772
NOTE: file regress/ker/wh1_freelist.c, line 1102 PR62953 -- fragmented kernel/proc free lists.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
P_list 7, entries 796, min 0, max 17408
P_mmap ---, free 4671904

P_slot	count	memory	maxrun	avgrun	first
P_0	4	32	2	1	777
P_16	4	88	1	1	489
P_32	9	392	2	1	379
P_64	6	464	2	1	564
P_128	1	184	1	1	779
P_256	132	111216	2	1	0
P_1024	640	4559528	640	640	1

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
P_list 7, entries 1649, min 0, max 12248
P_mmap ---, free 463672

P_slot	Count	memory	maxrun	avgrun	first
P_0	203	1624	2	1	1
P_16	413	8904	3	1	3
P_32	396	19096	88	30	0
P_64	32	2880	9	1	80
P_128	21	3912	2	1	672
P_256	524	147952	15	11	673
P_1024	60	279304	60	60	695

NOTE: file regress/ker/wh1_freelist.c, line 1042 fragmenting KER free list.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
K_list 6, entries 29, min 0, max 294656
K_mmap ---, free 974560

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	16
K_16	1	16	1	1	25
K_32	6	208	3	1	13
K_64	2	288	2	2	14
K_256	0	0	0	0	4294967295
K_1024	18	974032	18	18	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
K_list 6, entries 32, min 0, max 155216
K_mmap ---, free 659168

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	19
K_16	1	16	1	1	28
K_32	6	208	3	1	16
K_64	2	288	2	2	17
K_256	0	0	0	0	4294967295
K_1024	21	658640	21	21	0

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
P_list 7, entries 1649, min 0, max 12248

P_mmap ---, free 463672

P_slot	count	memory	maxrun	avgrun	first
P_0	203	1624	2	1	1
P_16	413	8904	3	1	3
P_32	396	19096	88	30	0
P_64	32	2880	9	1	80
P_128	21	3912	2	1	672
P_256	524	147952	15	11	673
P_1024	60	279304	60	60	695

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
P_list 7, entries 798, min 0, max 17408
P_mmap ---, free 4671840

P_slot	count	memory	maxrun	avgrun	first
P_0	6	48	2	1	5
P_16	4	88	1	1	491
P_32	8	360	2	1	381
P_64	7	576	2	1	566
P_128	2	392	1	1	0
P_256	131	110896	2	1	15
P_1024	640	4559480	640	640	1

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
K_list 6, entries 33, min 0, max 155216
K_mmap ---, free 667360

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	20
K_16	1	16	1	1	29
K_32	6	208	3	1	17
K_64	2	288	2	2	18
K_256	0	0	0	0	4294967295
K_1024	22	666832	22	22	0

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
K_list 6, entries 30, min 0, max 294656
K_mmap ---, free 982752

K_slot	count	memory	maxrun	avgrun	first
K_0	2	16	1	1	17
K_16	1	16	1	1	26
K_32	6	208	3	1	14
K_64	2	288	2	2	15
K_256	0	0	0	0	4294967295
K_1024	19	982224	19	19	0

NOTE: file regress/ker/wh1_freelist.c, line 1205 comparing the results...

POINT: file regress/ker/wh1_freelist.c, line 657 SHM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SHM - 834.872ms -> 926.858ms: change (after/before) = 1.11
POINT: file regress/ker/wh1_freelist.c, line 657 SEM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SEM - 420.935ms -> 460.929ms: change (after/before) = 1.10
POINT: file regress/ker/wh1_freelist.c, line 657 CHANNEL - threshold 8.000000
PASS: file regress/ker/wh1_freelist.c, line 672 CHANNEL - 590.909ms -> 593.909ms: change (after/before) = 1.01
POINT: file regress/ker/wh1_freelist.c, line 657 TIMER - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 TIMER - 518.920ms -> 522.919ms: change (after/before) = 1.01
POINT: file regress/ker/wh1_freelist.c, line 1209 Total testcase performance
PASS: file regress/ker/wh1_freelist.c, line 1213 passed.
STOP: wh1_freelist, pid 32772 point=5,pass=5,fail=0,xpass=0,xfail=0,unres=0
70.49s real 21.81s user 8.08s system
#

==========
stock 6.4.2 (head) UNI on a Intel 4 way (2x2 hyperthreaded)
==========

$ pidin in
CPU:X86 Release:6.4.2 FreeMem:1707Mb/2047Mb BootTime:Oct 22 11:15:21 UTC 2009
Processes: 31, Threads: 80
Processor1: 686 Intel 686 F6M14S8 1667MHz FPU
$ uname -a
QNX shivintelc4 6.4.2 2009/10/22-10:50:31UTC x86pc x86
$ time ./wh1_freelist -p -vv
START: ./wh1_freelist, pid 327711
NOTE: file regress/ker/wh1_freelist.c, line 1102 PR62953 -- fragmented kernel/proc free lists.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
P_list 7, entries 883, min 0, max 10072
P_mmap ---, free 4725536

P_slot	count	memory	maxrun	avgrun	first
P_0	41	328	6	2	792
P_16	42	760	15	5	776
P_32	10	408	3	1	430
P_64	11	880	3	1	144
P_128	4	680	3	2	781
P_256	86	46968	2	1	10
P_1024	689	4675512	689	689	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
P_list 7, entries 13644, min 0, max 8520
P_mmap ---, free 517304

P_slot	count	memory	maxrun	avgrun	first
P_0	117	936	6	1	16
P_16	12478	298376	64	13	2
P_32	476	22032	10	1	0
P_64	338	25672	277	6	61
P_128	13	2320	3	1	10499
P_256	205	67208	201	51	10484
P_1024	17	100760	17	17	13520

NOTE: file regress/ker/wh1_freelist.c, line 1042 fragmenting KER free list.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
K_list 6, entries 236, min 0, max 519760
K_mmap ---, free 975328

K_slot	count	memory	maxrun	avgrun	first
K_0	37	296	3	1	49
K_16	76	1704	24	4	5
K_32	51	2016	9	2	3
K_64	54	6272	14	6	37
K_256	9	4768	7	3	0
K_1024	9	960272	9	9	1

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
K_list 6, entries 6960, min 0, max 519760
K_mmap ---, free 813552

K_slot	Count	memory	maxrun	avgrun	first
K_0	39	312	3	1	6750
K_16	6377	152928	32	15	2
K_32	463	15464	414	18	0
K_64	66	7184	26	8	6772
K_256	8	4016	7	4	6768
K_1024	7	633648	7	7	6769

NOTE: file regress/ker/wh1_freelist.c, line 1196 running 'after' benchmarks...
NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:

P_list 7, entries 13644, min 0, max 8520
P_mmap ---, free 517304

P_slot	count	memory	maxrun	avgrun	first
P_0	117	936	6	1	16
P_16	12478	298376	64	13	2
P_32	476	22032	10	1	0
P_64	338	25672	277	6	61
P_128	13	2320	3	1	10499
P_256	205	67208	201	51	10484
P_1024	17	100760	17	17	13520

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
P_list 7, entries 883, min 0, max 10072
P_mmap ---, free 4725504

P_slot	count	memory	maxrun	avgrun	first
P_0	41	328	6	2	792
P_16	42	760	15	5	776
P_32	10	408	3	1	430
P_64	11	880	3	1	144
P_128	4	680	3	2	781
P_256	86	46968	2	1	10
P_1024	689	4675480	689	689	0

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
K_list 6, entries 6961, min 0, max 519760
K_mmap ---, free 813552

K_slot	count	memory	maxrun	avgrun	first
K_0	39	312	3	1	6750
K_16	6376	152904	32	15	2
K_32	463	15464	406	17	0
K_64	67	7272	26	7	114
K_256	8	4016	7	4	6768
K_1024	8	633584	8	8	6769

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
K_list 6, entries 239, min 0, max 519760
K_mmap ---, free 975264

K_slot	count	memory	maxrun	avgrun	first
K_0	37	296	3	1	63
K_16	51	1104	15	2	2
K_32	53	2120	9	2	0
K_64	80	8208	26	8	32
K_256	8	4144	7	4	46
K_1024	10	959392	10	10	29

```
NOTE:  file regress/ker/wh1_freelist.c, line 1205 comparing the results...
POINT: file regress/ker/wh1_freelist.c, line 657       SHM - threshold 5.000000
PASS:  file regress/ker/wh1_freelist.c, line 672         SHM -    86.986ms ->  102.984ms: change
(after/before) =  1.18
POINT: file regress/ker/wh1_freelist.c, line 657       SEM - threshold 5.000000
PASS:  file regress/ker/wh1_freelist.c, line 672         SEM -    31.995ms ->   42.993ms: change
(after/before) =  1.34
POINT: file regress/ker/wh1_freelist.c, line 657    CHANNEL - threshold 8.000000
PASS:  file regress/ker/wh1_freelist.c, line 672    CHANNEL -    49.992ms ->  169.973ms: change
(after/before) =  3.40
POINT: file regress/ker/wh1_freelist.c, line 657      TIMER - threshold 5.000000
PASS:  file regress/ker/wh1_freelist.c, line 672      TIMER -    75.988ms ->  156.975ms: change
(after/before) =  2.07
POINT: file regress/ker/wh1_freelist.c, line 1209 Total testcase performance
PASS:  file regress/ker/wh1_freelist.c, line 1213 passed.
STOP:  ./wh1_freelist, pid 327711 point=5,pass=5,fail=0,xpass=0,xfail=0,unres=0
   6.83s real     1.38s user     1.14s system
$ pidin in
CPU:X86 Release:6.4.2  FreeMem:1700Mb/2047Mb BootTime:Oct 22 11:15:21 UTC 2009
Processes: 31, Threads: 80
Processor1: 686 Intel 686 F6M14S8 1667MHz FPU

================================================
modified 6.4.2  (head) UNI on a Intel 4 way (2x2 hyperthreaded)
================================================

************************
Without Memmrefs as SOULS
************************

$ pidin in
CPU:X86 Release:6.4.2  FreeMem:1707Mb/2047Mb BootTime:Oct 22 11:19:24 UTC 2009
Processes: 31, Threads: 80
Processor1: 686 Intel 686 F6M14S8 1665MHz FPU
$ uname -a
QNX shivintelc4 6.4.2 2009/10/22-11:10:48UTC x86pc x86
$ time ./wh1_freelist -p -vv
START: ./wh1_freelist, pid 327711
NOTE:  file regress/ker/wh1_freelist.c, line 1102 PR62953 -- fragmented kernel/proc free lists.
NOTE:  file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
P_list 7, entries 882, min 0, max 10376
P_mmap ---, free 4725640
```

P_slot	count	memory	maxrun	avgrun	first
P_0	35	280	4	1	788
P_16	34	568	9	3	778
P_32	14	592	5	1	210
P_64	18	1440	4	1	25
P_128	5	904	2	1	779
P_256	82	40264	4	1	9
P_1024	694	4681592	694	694	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
P_list 7, entries 13622, min 0, max 8520
P_mmap ---, free 517408

P_slot	count	memory	maxrun	avgrun	first
P_0	126	1008	4	1	16
P_16	12448	297856	64	13	2
P_32	470	21624	11	2	0
P_64	332	25168	258	5	31
P_128	16	3040	2	1	10368
P_256	212	69024	205	42	10382
P_1024	18	99688	18	18	13499

NOTE: file regress/ker/wh1_freelist.c, line 1042 fragmenting KER free list.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
K_list 6, entries 115, min 0, max 532048
K_mmap ---, free 1040960

K_slot	count	memory	maxrun	avgrun	first
K_0	6	48	2	1	23
K_16	8	160	2	1	8
K_32	31	1320	4	1	4
K_64	38	5008	7	2	5
K_256	8	3552	3	1	12
K_1024	24	1030872	24	24	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
K_list 6, entries 118, min 0, max 532048
K_mmap ---, free 725568

K_slot	count	memory	maxrun	avgrun	first
K_0	6	48	2	1	26
K_16	8	160	2	1	11
K_32	31	1320	4	1	7
K_64	38	5008	7	2	8
K_256	8	3552	3	1	15
K_1024	27	715480	27	27	0

NOTE: file regress/ker/wh1_freelist.c, line 1196 running 'after' benchmarks...

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
P_list 7, entries 13622, min 0, max 8520
P_mmap ---, free 517408

P_slot	count	memory	maxrun	avgrun	first
P_0	126	1008	4	1	16
P_16	12448	297856	64	13	2
P_32	470	21624	11	2	0
P_64	332	25168	258	5	31
P_128	16	3040	2	1	10368
P_256	212	69024	205	42	10382
P_1024	18	99688	18	18	13499

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
P_list 7, entries 882, min 0, max 10376
P_mmap ---, free 4725608

P_slot	count	memory	maxrun	avgrun	first
P_0	35	280	4	1	788
P_16	34	568	9	3	778
P_32	14	592	5	1	210
P_64	18	1440	4	1	25
P_128	5	904	2	1	779
P_256	82	40264	4	1	9
P_1024	694	4681560	694	694	0

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
K_list 6, entries 118, min 0, max 532048
K_mmap ---, free 725568

K_slot	count	memory	maxrun	avgrun	first
K_0	6	48	2	1	26
K_16	8	160	2	1	11
K_32	31	1320	4	1	7
K_64	38	5008	7	2	8
K_256	8	3552	3	1	15
K_1024	27	715480	27	27	0

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
K_list 6, entries 115, min 0, max 532048
K_mmap ---, free 1040960

K_slot	count	memory	maxrun	avgrun	first
K_0	6	48	2	1	23
K_16	8	160	2	1	8
K_32	31	1320	4	1	4
K_64	38	5008	7	2	5
K_256	8	3552	3	1	12
K_1024	24	1030872	24	24	0

NOTE: file regress/ker/wh1_freelist.c, line 1205 comparing the results...
POINT: file regress/ker/wh1_freelist.c, line 657 SHM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SHM - 87.986ms -> 102.984ms: change (after/before) = 1.17
POINT: file regress/ker/wh1_freelist.c, line 657 SEM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SEM - 32.994ms -> 43.993ms: change (after/before) = 1.33
POINT: file regress/ker/wh1_freelist.c, line 657 CHANNEL - threshold 8.000000
PASS: file regress/ker/wh1_freelist.c, line 672 CHANNEL - 48.992ms -> 47.992ms: change (after/before) = 0.98
POINT: file regress/ker/wh1_freelist.c, line 657 TIMER - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 TIMER - 51.992ms -> 51.992ms: change (after/before) = 1.00
POINT: file regress/ker/wh1_freelist.c, line 1209 Total testcase performance
PASS: file regress/ker/wh1_freelist.c, line 1213 passed.
STOP: ./wh1_freelist, pid 327711 point=5,pass=5,fail=0,xpass=0,xfail=0,unres=0
6.32s real 1.39s user 0.56s system
$

****
With Memmrefs as SOULS
****

$ pidin in
CPU:X86 Release:6.4.2 FreeMem:1707Mb/2047Mb BootTime:Oct 22 11:22:38 UTC 2009
Processes: 31, Threads: 81
Processor1: 686 Intel 686 F6M14S8 1667MHz FPU
$ uname -a
QNX shivintelc4 6.4.2 2009/10/22-11:13:48UTC x86pc x86
$ time ./wh1_freelist -p -vv
START: ./wh1_freelist, pid 286751
NOTE: file regress/ker/wh1_freelist.c, line 1102 PR62953 -- fragmented kernel/proc free lists.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
P_list 7, entries 849, min 0, max 17376
P_mmap ---, free 4675208

P_slot	count	memory	maxrun	avgrun	first
P_0	25	200	7	1	782
P_16	26	440	8	2	774
P_32	8	400	1	1	34
P_64	18	1568	4	2	302
P_128	2	360	1	1	12
P_256	132	110736	3	1	5
P_1024	638	4561504	638	638	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
P_list 7, entries 1713, min 0, max 9664
P_mmap ---, free 466976

P_slot	count	memory	maxrun	avgrun	first
P_0	213	1704	7	1	8
P_16	459	9704	8	1	2
P_32	402	19456	186	44	0
P_64	30	2736	5	1	198
P_128	11	2040	1	1	770
P_256	536	152312	15	11	678
P_1024	62	279024	62	62	689

NOTE: file regress/ker/wh1_freelist.c, line 1042 fragmenting KER free list.
NOTE: file regress/ker/wh1_freelist.c, line 1044 Before fragmentation:
K_list 6, entries 110, min 0, max 532048
K_mmap ---, free 1035072

K_slot	count	memory	maxrun	avgrun	first
K_0	5	40	2	1	23
K_16	7	144	1	1	8
K_32	32	1320	4	1	4
K_64	36	4656	7	2	5
K_256	8	3288	3	2	43
K_1024	22	1025624	22	22	0

NOTE: file regress/ker/wh1_freelist.c, line 1050 After fragmentation:
K_list 6, entries 112, min 0, max 532048
K_mmap ---, free 719680

K_slot	count	memory	maxrun	avgrun	first
K_0	5	40	2	1	25
K_16	7	144	1	1	10
K_32	32	1320	4	1	6
K_64	36	4656	7	2	7
K_256	8	3288	3	2	45
K_1024	24	710232	24	24	0

NOTE: file regress/ker/wh1_freelist.c, line 1196 running 'after' benchmarks...
NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
P_list 7, entries 1713, min 0, max 9664
P_mmap ---, free 466976

P_slot	count	memory	maxrun	avgrun	first
P_0	213	1704	7	1	8
P_16	459	9704	8	1	2
P_32	402	19456	186	44	0
P_64	30	2736	5	1	198
P_128	11	2040	1	1	770
P_256	536	152312	15	11	678
P_1024	62	279024	62	62	689

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
P_list 7, entries 851, min 0, max 17376
P_mmap ---, free 4675176

P_slot	count	memory	maxrun	avgrun	first
P_0	27	216	7	1	3
P_16	26	440	8	2	776
P_32	8	400	1	1	36
P_64	18	1568	4	2	304
P_128	2	360	1	1	14
P_256	132	110736	3	1	7
P_1024	638	4561456	638	638	0

NOTE: file regress/ker/wh1_freelist.c, line 1072 Before cleanup:
K_list 6, entries 112, min 0, max 532048
K_mmap ---, free 719680

K_slot	count	memory	maxrun	avgrun	first
K_0	5	40	2	1	25
K_16	7	144	1	1	10
K_32	32	1320	4	1	6
K_64	36	4656	7	2	7
K_256	8	3288	3	2	45
K_1024	24	710232	24	24	0

NOTE: file regress/ker/wh1_freelist.c, line 1077 After cleanup:
K_list 6, entries 110, min 0, max 532048
K_mmap ---, free 1035072

K_slot	count	memory	maxrun	avgrun	first
K_0	5	40	2	1	25
K_16	7	144	1	1	10
K_32	32	1320	4	1	6
K_64	36	4656	7	2	7
K_256	8	3288	3	2	45
K_1024	24	710232	24	24	0

NOTE: file regress/ker/wh1_freelist.c, line 1205 comparing the results...
POINT: file regress/ker/wh1_freelist.c, line 657 SHM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SHM - 87.986ms -> 92.985ms: change (after/before) = 1.06
POINT: file regress/ker/wh1_freelist.c, line 657 SEM - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 SEM - 32.994ms -> 33.994ms: change (after/before) = 1.03
POINT: file regress/ker/wh1_freelist.c, line 657 CHANNEL - threshold 8.000000
PASS: file regress/ker/wh1_freelist.c, line 672 CHANNEL - 47.992ms -> 48.992ms: change (after/before) = 1.02
POINT: file regress/ker/wh1_freelist.c, line 657 TIMER - threshold 5.000000
PASS: file regress/ker/wh1_freelist.c, line 672 TIMER - 51.992ms -> 51.992ms: change (after/before) = 1.00
POINT: file regress/ker/wh1_freelist.c, line 1209 Total testcase performance
PASS: file regress/ker/wh1_freelist.c, line 1213 passed.
STOP: ./wh1_freelist, pid 286751 point=5,pass=5,fail=0,xpass=0,xfail=0,unres=0
5.96s real 1.45s user 0.60s system
$ pidin in
CPU:X86 Release:6.4.2 FreeMem:1701Mb/2047Mb BootTime:Oct 22 11:22:38 UTC 2009
Processes: 31, Threads: 80
Processor1: 686 Intel 686 F6M14S8 1667MHz FPU

Revision 2 Testing:

QEMU

==========
New Code + Memref
==========

wh1_freelist

Mem before: 130662400

Test Time: 70.69s real 21.81s user 8.12s system

Fragmentation Results
SHM - 824.873ms -> 1064.837ms: change (after/before) = 1.29
SEM - 419.935ms -> 530.918ms: change (after/before) = 1.26
CHANNEL - 587.910ms -> 591.909ms: change (after/before) = 1.01
TIMER - 526.919ms -> 527.919ms: change (after/before) = 1.00

Mem After: 125325312

Mem used: 5337088 Bytes

==========
New Code + Memref + Ocb
==========

wh1_freelist

Mem before: 130662400

Test Time: 69.76s real 21.98s user 8.02s system

Fragmentation Results
SHM - 812.875ms -> 880.865ms: change (after/before) = 1.08
SEM - 413.936ms -> 446.931ms: change (after/before) = 1.08
CHANNEL - 590.909ms -> 586.910ms: change (after/before) = 0.99
TIMER - 543.916ms -> 537.917ms: change (after/before) = 0.99

Mem After: 125321216

Mem used: 5341184 Bytes

=====
Stock

=====

Mem before: 130760704

Test Time: 80.32s real 22.32s user 14.90s system

Fragmentation Results
SHM   - 837.871ms -> 1073.835ms: change (after/before) =   1.28
SEM   - 421.935ms -> 608.906ms: change (after/before) =   1.44
CHANNEL - 604.907ms -> 1519.767ms: change (after/before) =   2.51
TIMER   - 1066.836ms -> 2277.651ms: change (after/before) =   2.13

Mem After: 124297216

Mem used: 6463488 Bytes

===
641
===

Mem before: 130785280

Test Time: 151.14s real 22.72s user 40.25s system

Fragmentation Results
SHM   - 753.884ms -> 4850.257ms: change (after/before) =   6.43
SEM   - 387.940ms -> 3716.431ms: change (after/before) =   9.58
CHANNEL - 591.909ms -> 1162.822ms: change (after/before) =   1.96
TIMER   - 657.899ms -> 19005.091ms: change (after/before) = 28.89

Mem After: 124317696

Mem used: 6467584 Bytes

****
Real Hardware: Intel Dual Core running Uni
****

===
641
===

Mem before: 1789890560

Test Time: 44.16s real 1.54s user 5.92s system

Fragmentation Results

SHM - 80.987ms -> 3212.508ms: change (after/before) = 39.67
SEM - 29.995ms -> 3045.533ms: change (after/before) = 101.53
CHANNEL - 47.992ms -> 154.976ms: change (after/before) = 3.23
TIMER - 59.990ms -> 3239.504ms: change (after/before) = 54.00

Mem After: 1783414784

Mem used: 6475776 Bytes

=====
Stock
=====

Mem before: 1789874176

Test Time: 6.75s real 1.40s user 1.11s system

Fragmentation Results
SHM - 86.986ms -> 102.984ms: change (after/before) = 1.18
SEM - 32.994ms -> 43.993ms: change (after/before) = 1.33
CHANNEL - 48.992ms -> 169.973ms: change (after/before) = 3.47
TIMER - 76.988ms -> 141.978ms: change (after/before) = 1.84

Mem After: 1783410688

Mem used: 6463488 Bytes

======================
New Code + Memref
======================

wh1_freelist

Mem before: 1790169088

Test Time: 5.95s real 1.38s user 0.57s system

Fragmentation Results
SHM - 86.986ms -> 88.986ms: change (after/before) = 1.02
SEM - 31.995ms -> 32.994ms: change (after/before) = 1.03
CHANNEL - 47.992ms -> 48.992ms: change (after/before) = 1.02
TIMER - 51.992ms -> 51.992ms: change (after/before) = 1.00

Mem After: 1784823808

Mem used: 5345280 Bytes

==========
New Code + Memref + Ocb
==========

wh1_freelist

Mem before: 1790205952

Test Time: 5.93s real 1.46s user 0.56s system

Fragmentation Results
SHM - 85.986ms -> 91.985ms: change (after/before) = 1.07
SEM - 33.994ms -> 34.994ms: change (after/before) = 1.03
CHANNEL - 47.992ms -> 47.992ms: change (after/before) = 1.00
TIMER - 51.992ms -> 51.992ms: change (after/before) = 1.00

Mem After: 1784860672

Mem used: 5345280 Bytes

==========
Other Tests All on QEMU
==========
****
ker_pr10963
****

New Code: (Memref + OCB)

Mem Before: 130662400
Mem After : 130662400
Mem Used : 0

iterations completed: 160157 (pass 4)
iterations completed: 153021 (pass 5)

Time: 789.43s real 87.11s user 625.62s system

****
wh1_SignalKill
****

MemBefore : 130662400
MemAfter : 130662400
Mem Used : 0
count : 32057

Time: 6.07s real 0.06s user 0.15s system

****
wh1_spawn
****

Mem Before: 130646016
Mem After : 128856064
Mem Used : 1789952

****
wh1_mutex
****

Mem Before: 130617344
Mem After : 130572288
Mem Used : 45056

Time: 0.50s real 0.13s user 0.24s system

****
wh2_mutex
****

Mem Before: 130617344
Mem After : 130617344
Mem Used : 0

Time: 3.19s real 0.04s user 0.01s system

Stock Code:

****
ker_pr10963
****

Mem Before: 130719744
Mem After : 108257280
Mem Used : 22462464

iterations completed:
iterations completed: 3978 (pass 3) (Out of Memory)
iterations completed: 6901 (pass 4)
iterations completed: 6845 (pass 5)

Time: 949.87s real 4.27s user 865.92s system

****
wh1_SignalKill
****

MemBefore : 130719744
MemAfter : 129953792
Mem Used :   765952
count   : 31932

Time: 6.07s real   0.07s user   0.33s system

****
wh1_spawn
****

Mem Before: 130703360
Mem After : 126091264
Mem Used :   4612096

****
wh1_mutex
****

Mem Before: 130674688
Mem After : 129581056
Mem Used :   1093632

Time: 0.90s real   0.14s user   0.64s system

****
wh2_mutex
****

Mem Before: 130674688
Mem After : 130662400
Mem Used :   12288

Time: 3.19s real   0.05s user   0.01s system