

[patch 5/8] cpusets v3 – New bitmap lists format

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2004-06/6948.html>

From: Paul Jackson (pj_at_sgi.com)

Date: 06/29/04

Date: Tue, 29 Jun 2004 04:22:20 -0700 (PDT)

To: linux-kernel@vger.kernel.org

A bitmap print and parse format that provides lists of ranges of numbers, to be first used for by cpusets.

Index: 2.6.7-mm1/include/linux/bitmap.h

```
=====
--- 2.6.7-mm1.orig/include/linux/bitmap.h 2004-06-23 02:44:49.000000000 -0700
+++ 2.6.7-mm1/include/linux/bitmap.h 2004-06-23 02:44:56.000000000 -0700
@@ -41,7 +41,8 @@
 * bitmap_shift_right(dst, src, n, nbits) *dst = *src >> n
 * bitmap_shift_left(dst, src, n, nbits) *dst = *src << n
 * bitmap_scnprintf(buf, len, src, nbits) Print bitmap src to buf
- * bitmap_parse(ubuf, ulen, dst, nbits) Parse bitmap dst from buf
+ * bitmap_parse(ubuf, ulen, dst, nbits) Parse bitmap dst from user buf
+ * bitmap_parselist(buf, dst, nbits) Parse bitmap dst from list
 */

/*
@@ -98,6 +99,8 @@ extern int bitmap_scnprintf(char *buf, u
      const unsigned long *src, int nbits);
extern int bitmap_parse(const char __user *ubuf, unsigned int ulen,
      unsigned long *dst, int nbits);
+extern int bitmap_parselist(const char *buf, unsigned long *maskp,
+ int nmaskbits);

#define BITMAP_LAST_WORD_MASK(nbites) \
(\
```

Index: 2.6.7-mm1/include/linux/cpumask.h

```
=====
--- 2.6.7-mm1.orig/include/linux/cpumask.h 2004-06-23 02:44:49.000000000 -0700
+++ 2.6.7-mm1/include/linux/cpumask.h 2004-06-23 02:44:56.000000000 -0700
@@ -10,6 +10,8 @@
 *
 * For details of cpumask_scnprintf() and cpumask_parse(),
 * see bitmap_scnprintf() and bitmap_parse() in lib/bitmap.c.
+ * For details of cpulist_parse(), see bitmap_parselist(), also
+ * in lib/bitmap.c.
 *
```

Linux–Kernel: [patch 5/8] cpuset v3 – New bitmap lists format

```
* The available cpumask operations are:
*
@@ -46,6 +48,7 @@
*
* int cpumask_scnprintf(buf, len, mask) Format cpumask for printing
* int cpumask_parse(ubuf, ulen, mask) Parse ascii string as cpumask
+ * int cpulist_parse(buf, map) Parse ascii string as cpulist
*
* for_each_cpu_mask(cpu, mask) for–loop cpu over mask
*
@@ -262,14 +265,20 @@ static inline int __cpumask_scnprintf(ch
    return bitmap_scnprintf(buf, len, srcp->bits, nbits);
}

-#define cpumask_parse(ubuf, ulen, src) \
- __cpumask_parse((ubuf), (ulen), &(src), NR_CPUS)
+#define cpumask_parse(ubuf, ulen, dst) \
+ __cpumask_parse((ubuf), (ulen), &(dst), NR_CPUS)
static inline int __cpumask_parse(const char __user *buf, int len,
                                cpumask_t *dstp, int nbits)
{
    return bitmap_parse(buf, len, dstp->bits, nbits);
}

+#define cpulist_parse(buf, dst) __cpulist_parse((buf), &(dst), NR_CPUS)
+static inline int __cpulist_parse(const char *buf, cpumask_t *dstp, int nbits)
+{
+ return bitmap_parselist(buf, dstp->bits, nbits);
+}
+
+
+#if NR_CPUS > 1
+#define for_each_cpu_mask(cpu, mask) \
    for ((cpu) = first_cpu(mask); \
Index: 2.6.7-mm1/include/linux/nodemask.h
=====
--- 2.6.7-mm1.orig/include/linux/nodemask.h 2004-06-23 02:44:49.000000000 -0700
+++ 2.6.7-mm1/include/linux/nodemask.h 2004-06-23 02:44:56.000000000 -0700
@@ -10,7 +10,9 @@
*
* For details of nodemask_scnprintf() and nodemask_parse(),
* see bitmap_scnprintf() and bitmap_parse() in lib/bitmap.c.
- *
+ * For details of nodelist_parse(), see bitmap_parselist(), also
+ * in lib/bitmap.c.
+
* The available nodemask operations are:
*
* void node_set(node, mask) turn on bit 'node' in mask
@@ -46,6 +48,7 @@
*
* int nodemask_scnprintf(buf, len, mask) Format nodemask for printing
```

Linux–Kernel: [patch 5/8] cpusets v3 – New bitmap lists format

```

* int nodemask_parse(ubuf, ulen, mask) Parse ascii string as nodemask
+ * int nodelist_parse(buf, map) Parse ascii string as nodelist
*
* for_each_node_mask(node, mask) for–loop node over mask
*
@@ –275,14 +278,20 @@ static inline int __nodemask_scnprintf(c
    return bitmap_scnprintf(buf, len, srcp->bits, nbits);
}

–#define nodemask_parse(ubuf, ulen, src) \
– __nodemask_parse((ubuf), (ulen), &(src), MAX_NUMNODES)
+#define nodemask_parse(ubuf, ulen, dst) \
+ __nodemask_parse((ubuf), (ulen), &(dst), MAX_NUMNODES)
static inline int __nodemask_parse(const char __user *buf, int len,
    nodemask_t *dstp, int nbits)
{
    return bitmap_parse(buf, len, dstp->bits, nbits);
}

+#define nodelist_parse(buf, dst) __nodelist_parse((buf), &(dst), MAX_NUMNODES)
+static inline int __nodelist_parse(const char *buf, nodemask_t *dstp, int nbits)
+{
+ return bitmap_parselist(buf, dstp->bits, nbits);
+}
+
+#if MAX_NUMNODES > 1
+#define for_each_node_mask(node, mask) \
    for ((node) = first_node(mask); \
Index: 2.6.7–mm1/lib/bitmap.c
=====
--- 2.6.7–mm1.orig/lib/bitmap.c 2004–06–23 02:44:49.000000000 –0700
+++ 2.6.7–mm1/lib/bitmap.c 2004–06–23 02:44:56.000000000 –0700
@@ –291,6 +291,7 @@ EXPORT_SYMBOL(__bitmap_weight);
#define nbits_to_hold_value(val) fls(val)
#define roundup_power2(val, modulus) (((val) + (modulus) – 1) & ~((modulus) – 1))
#define unhex(c) (isdigit(c) ? (c – '0') : (toupper(c) – 'A' + 10))
+#define BASEDEC 10 /* fancier cpuset lists input in decimal */

/**
 * bitmap_scnprintf – convert bitmap to an ASCII hex string.
@@ –408,3 +409,86 @@ int bitmap_parse(const char __user *ubuf
    return 0;
}
EXPORT_SYMBOL(bitmap_parse);
+
+/**
+ * bitmap_parselist – parses a more flexible format for inputting bit masks
+ * @buf: read nul–terminated user string from this buffer
+ * @mask: write resulting mask here
+ * @nmaskbits: number of bits in mask to be written
+ *

```

Linux-Kernel: [patch 5/8] cpusets v3 – New bitmap lists format

```
+ * The input format supports a space separated list of one or more comma
+ * separated sequences of ascii decimal bit numbers and ranges. Each
+ * sequence may be preceded by one of the prefix characters '=',
+ * '-', '+', or '!', which have the following meanings:
+ * '=': rewrite the mask to have only the bits specified in this sequence
+ * '-': turn off the bits specified in this sequence
+ * '+': turn on the bits specified in this sequence
+ * '!': same as '-'.
+ *
+ * If no such initial character is specified, then the default prefix '='
+ * is presumed. The list is evaluated and applied in left to right order.
+ *
+ * Examples of input format:
+ * 0-4,9 # rewrites to 0,1,2,3,4,9
+ * -9 # removes 9
+ * +6-8 # adds 6,7,8
+ * 1-6 -0,2-4 +11-14,16-19 -14-16 # same as 1,5,6,11-13,17-19
+ * 1-6 -0,2-4 +11-14,16-19 =14-16 # same as just 14,15,16
+ *
+ * Possible errno's returned for invalid input strings are:
+ * -EINVAL: second number in range smaller than first
+ * -ERANGE: bit number specified too large for mask
+ * -EINVAL: invalid prefix char (not '=', '-', '+', or '!')
+ */
+
+int bitmap_parselist(const char *buf, unsigned long *maskp, int nmaskbits)
+{
+ char *p, *q;
+ int masklen = BITS_TO_LONGS(nmaskbits);
+
+ while ((p = strsep((char **)&buf, " ")) != NULL) { /* blows const XXX */
+ char op = isdigit(*p) ? '=' : *p++;
+ unsigned long m[masklen];
+ int maskbytes = sizeof(m);
+ int i;
+
+ if (op == '-')
+ continue;
+ memset(m, 0, maskbytes);
+
+ while ((q = strsep(&p, ",")) != NULL) {
+ unsigned a = simple_strtoul(q, 0, BASEDEC);
+ unsigned b = a;
+ char *cp = strchr(q, '-');
+ if (cp)
+ b = simple_strtoul(cp + 1, 0, BASEDEC);
+ if (!(a <= b))
+ return -EINVAL;
+ if (b >= nmaskbits)
+ return -ERANGE;
+ while (a <= b) {
```

Linux-Kernel: [patch 5/8] cpuset v3 – New bitmap lists format

```
+ set_bit(a, m);
+ a++;
+ }
+ }
+
+ switch (op) {
+ case '=':
+ memcpy(maskp, m, maskbytes);
+ break;
+ case '!':
+ case '-':
+ for (i = 0; i < masklen; i++)
+ maskp[i] &= ~m[i];
+ break;
+ case '+':
+ for (i = 0; i < masklen; i++)
+ maskp[i] |= m[i];
+ break;
+ default:
+ return -EINVAL;
+ }
+ }
+ return 0;
+}
+EXPORT_SYMBOL(bitmap_parselist);
```

--

```
I won't rest till it's the best ...
Programmer, Linux Scalability
Paul Jackson <pj@sgi.com> 1.650.933.1373
```

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>