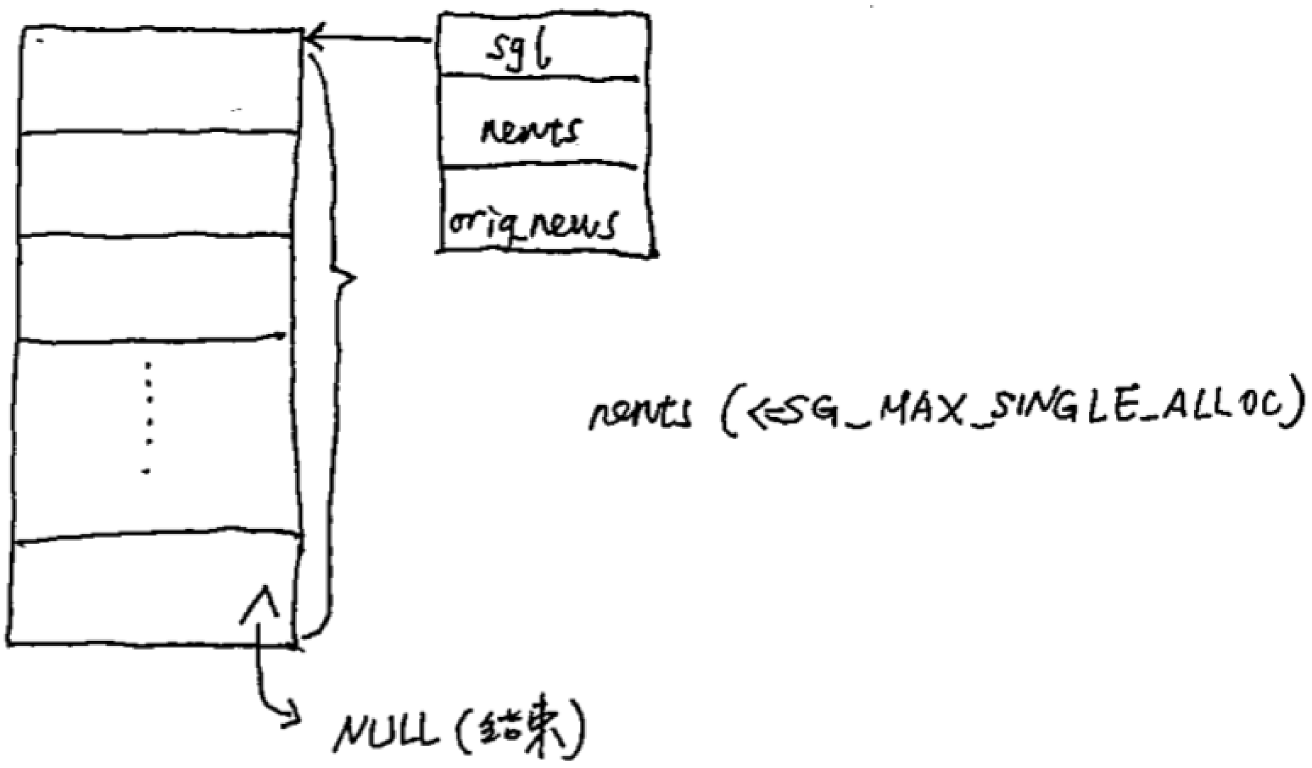
```c
struct sg_table {
        struct scatterlist *sgl;        /* the list */
        unsigned int nents;             /* number of mapped entries */
        unsigned int orig_nents;        /* original size of list */
};
```

nents v.s. orig_nents ?

在table的orig_nents <= SG_MAX_SINGLE_ALLOC，即整个sg_table就是单一的scatterlist array的情况下，nents = orig_nents。



在正常情况下，即使orig_nents > SG_MAX_SINGLE_ALLOC时，nents也是等于orig_nents的，但当__sg_alloc_table()中分配失败时，nents != orig_nents in __sg_alloc_table()

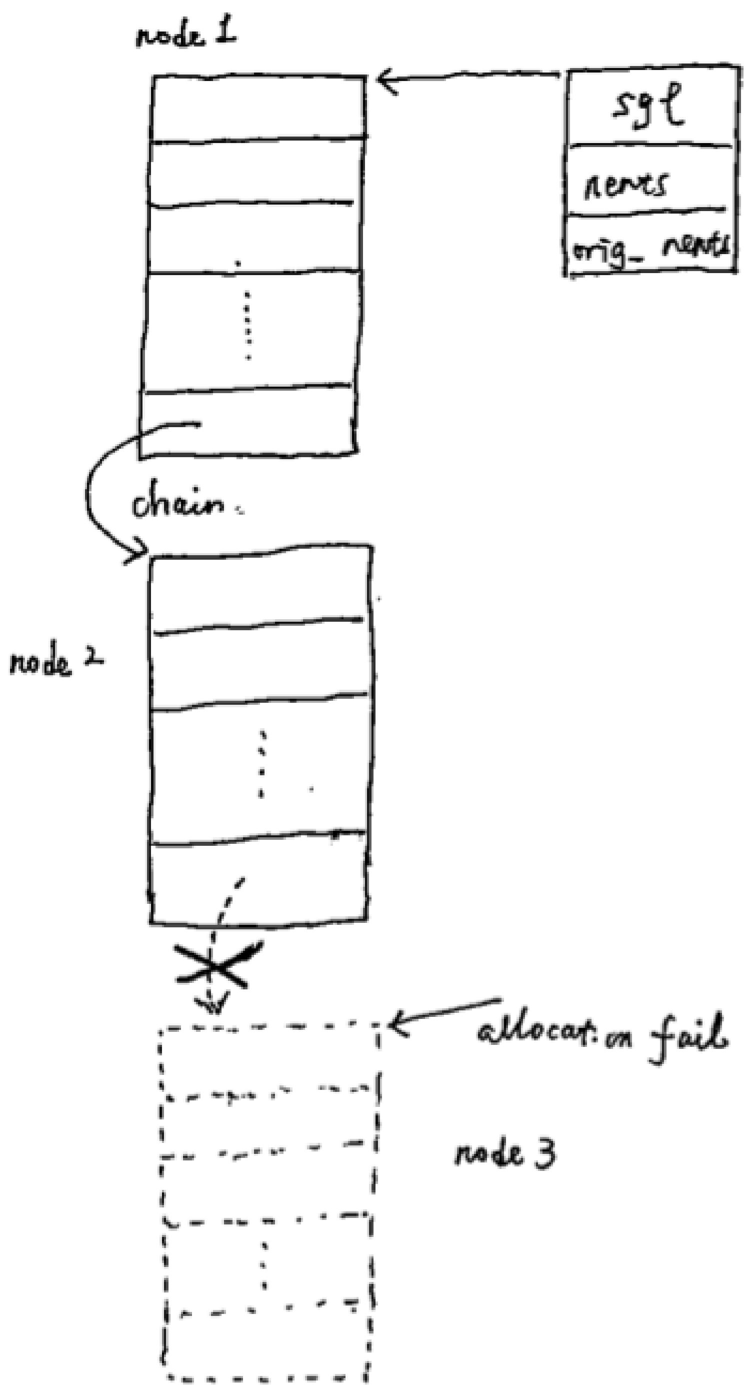
```c
int __sg_alloc_table(struct sg_table *table, unsigned int nents,
                     unsigned int max_ents, struct scatterlist *first_chunk,
                     gfp_t gfp_mask, sg_alloc_fn *alloc_fn)
{
        struct scatterlist *sg, *prv;
        unsigned int left;

        memset(table, 0, sizeof(*table));

        if (nents == 0)
                return -EINVAL;
#ifndef CONFIG_ARCH_HAS_SG_CHAIN
        if (WARN_ON_ONCE(nents > max_ents))
                return -EINVAL;
#endif

        left = nents;
        prv = NULL;
        do {
                unsigned int sg_size, alloc_size = left;

                if (alloc_size > max_ents) {
                        alloc_size = max_ents;
                        sg_size = alloc_size - 1;
                } else
                        sg_size = alloc_size;

                left -= sg_size;

                if (first_chunk) {
                        sg = first_chunk;
                        first_chunk = NULL;
                } else {
                        sg = alloc_fn(alloc_size, gfp_mask);
                }
                if (unlikely(!sg)) {
                        /*
                         * Adjust entry count to reflect that the last
                         * entry of the previous table won't be used for
                         * linkage.  Without this, sg_kfree() may get
                         * confused.
                         */
                        if (prv)
        ①
                                table->nents = ++table->orig_nents;     ②

                        return -ENOMEM;
                }

                sg_init_table(sg, alloc_size);
                table->nents = table->orig_nents += sg_size;    ③

                /*
```

```
                 * If this is the first mapping, assign the sg table header.
                 * If this is not the first mapping, chain previous part.
                 */
                if (prv)
                        sg_chain(prv, max_ents, sg);
                else
                        table->sgl = sg;

                /*
                 * If no more entries after this one, mark the end
                 */
                if (!left)
                        sg_mark_end(&sg[sg_size - 1]);

                prv = sg;
        } while (left);

        return 0;
}
```

①

allocation fail不是发生在第一个节点

②



当node 3 allocation fail,

table->orig_nents = 2 * (SG_MAX_SINGLE_ALLOC - 1) ,没变

table->nents = 2 * (SG_MAX_SINGLE_ALLOC - 1) + 1

即node 2的最后一个entry不再作为chain，而是作为scatterlist entry

③

table->nents = table->orig_nents += sg_size;

这里当allocate node 1时，table->nents = table->orig_nents = SG_MAX_SINGLE_ALLOC - 1

当allocate node 2时，table->nents = table->orig_nents = 2 * (SG_MAX_SINGLE_ALLOC - 1)