Subject: [PATCH net-2.6.25 9/10][NETNS][FRAGS]: Make the LRU list per namespace.
Posted by Pavel Emelianov on Tue, 22 Jan 2008 14:08:54 GMT
View Forum Message <> Reply to Message

The inet_frags.lru_list is used for evicting only, so we have
to make it per-namespace, to evict only those fragments, who's
namespace exceeded its high threshold, but not the whole hash.
Besides, this helps to avoid long loops  in evictor.

The spinlock is not per-namespace because it protects the
hash table as well, which is global.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---
 include/net/inet_frag.h                |    2 +-
 net/ipv4/inet_fragment.c               |    8 ++++----
 net/ipv4/ip_fragment.c                 |    2 +-
 net/ipv6/netfilter/nf_conntrack_reasm.c |    2 +-
 net/ipv6/reassembly.c                  |    2 +-
 5 files changed, 8 insertions(+), 8 deletions(-)

diff --git a/include/net/inet_frag.h b/include/net/inet_frag.h
index 1917fbe..3695ff4 100644
--- a/include/net/inet_frag.h
+++ b/include/net/inet_frag.h
@@ -4,6 +4,7 @@
 struct netns_frags {
  int   nqueues;
  atomic_t  mem;
+ struct list_head lru_list;

  /* sysctls */
  int   timeout;
@@ -32,7 +33,6 @@ struct inet_frag_queue {
 #define INETFRAGS_HASHSZ  64

 struct inet_frags {
- struct list_head lru_list;
  struct hlist_head hash[INETFRAGS_HASHSZ];
  rwlock_t  lock;
  u32   rnd;
diff --git a/net/ipv4/inet_fragment.c b/net/ipv4/inet_fragment.c
index fcf5252..f1b95e1 100644
--- a/net/ipv4/inet_fragment.c
+++ b/net/ipv4/inet_fragment.c
@@ -57,7 +57,6 @@ void inet_frags_init(struct inet_frags *f)

```
  for (i = 0; i < INETFRAGS_HASHSZ; i++)
   INIT_HLIST_HEAD(&f->hash[i]);

- INIT_LIST_HEAD(&f->lru_list);
  rwlock_init(&f->lock);

  f->rnd = (u32) ((num_physpages ^ (num_physpages>>7)) ^
@@ -74,6 +73,7 @@ void inet_frags_init_net(struct netns_frags *nf)
 {
  nf->nqueues = 0;
  atomic_set(&nf->mem, 0);
+ INIT_LIST_HEAD(&nf->lru_list);
 }
 EXPORT_SYMBOL(inet_frags_init_net);

@@ -156,12 +156,12 @@ int inet_frag_evictor(struct netns_frags *nf, struct inet_frags *f)
  work = atomic_read(&nf->mem) - nf->low_thresh;
  while (work > 0) {
   read_lock(&f->lock);
-  if (list_empty(&f->lru_list)) {
+  if (list_empty(&nf->lru_list)) {
    read_unlock(&f->lock);
    break;
   }

-  q = list_first_entry(&f->lru_list,
+  q = list_first_entry(&nf->lru_list,
     struct inet_frag_queue, lru_list);
   atomic_inc(&q->refcnt);
   read_unlock(&f->lock);
@@ -211,7 +211,7 @@ static struct inet_frag_queue *inet_frag_intern(struct netns_frags *nf,

  atomic_inc(&qp->refcnt);
  hlist_add_head(&qp->list, &f->hash[hash]);
- list_add_tail(&qp->lru_list, &f->lru_list);
+ list_add_tail(&qp->lru_list, &nf->lru_list);
  nf->nqueues++;
  write_unlock(&f->lock);
  return qp;
diff --git a/net/ipv4/ip_fragment.c b/net/ipv4/ip_fragment.c
index 00646ed..29b4b09 100644
--- a/net/ipv4/ip_fragment.c
+++ b/net/ipv4/ip_fragment.c
@@ -441,7 +441,7 @@ static int ip_frag_queue(struct ipq *qp, struct sk_buff *skb)
   return ip_frag_reasm(qp, prev, dev);

  write_lock(&ip4_frags.lock);
- list_move_tail(&qp->q.lru_list, &ip4_frags.lru_list);
```

```
+ list_move_tail(&qp->q.lru_list, &qp->q.net->lru_list);
  write_unlock(&ip4_frags.lock);
  return -EINPROGRESS;

diff --git a/net/ipv6/netfilter/nf_conntrack_reasm.c b/net/ipv6/netfilter/nf_conntrack_reasm.c
index 6eed991..022da6c 100644
--- a/net/ipv6/netfilter/nf_conntrack_reasm.c
+++ b/net/ipv6/netfilter/nf_conntrack_reasm.c
@@ -385,7 +385,7 @@ static int nf_ct_frag6_queue(struct nf_ct_frag6_queue *fq, struct sk_buff
*skb,
    fq->q.last_in |= FIRST_IN;
  }
  write_lock(&nf_frags.lock);
- list_move_tail(&fq->q.lru_list, &nf_frags.lru_list);
+ list_move_tail(&fq->q.lru_list, &nf_init_frags.lru_list);
  write_unlock(&nf_frags.lock);
  return 0;

diff --git a/net/ipv6/reassembly.c b/net/ipv6/reassembly.c
index 8520700..0c4bc46 100644
--- a/net/ipv6/reassembly.c
+++ b/net/ipv6/reassembly.c
@@ -424,7 +424,7 @@ static int ip6_frag_queue(struct frag_queue *fq, struct sk_buff *skb,
    return ip6_frag_reasm(fq, prev, dev);

  write_lock(&ip6_frags.lock);
- list_move_tail(&fq->q.lru_list, &ip6_frags.lru_list);
+ list_move_tail(&fq->q.lru_list, &fq->q.net->lru_list);
  write_unlock(&ip6_frags.lock);
  return -1;

--
1.5.3.4
```