
Subject: [PATCH 1/2] Uninline the sk_stream_alloc_pskb
Posted by [Pavel Emelianov](#) on Mon, 26 Nov 2007 17:14:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

This function seems too big for inlining. Indeed, it saves half-a-kilo when uninline:

```
add/remove: 1/0 grow/shrink: 0/7 up/down: 195/-719 (-524)
function      old    new  delta
sk_stream_alloc_pskb      -    195   +195
ip_rt_init      529    525    -4
__inet_lookup_listener      284    274   -10
tcp_sendmsg      2583   2486   -97
tcp_sendpage      1449   1305  -144
tso_fragment      417    267  -150
tcp_fragment      1149    992  -157
__tcp_push_pending_frames      1998   1841  -157
```

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

```
diff --git a/include/net/sock.h b/include/net/sock.h
index 67e35c7..492dc4a 100644
--- a/include/net/sock.h
+++ b/include/net/sock.h
@@ -1230,33 +1230,8 @@ static inline void sk_stream_moderate_sndbuf(struct sock *sk)
 }
 }

-static inline struct sk_buff *sk_stream_alloc_pskb(struct sock *sk,
-    int size, int mem,
-    gfp_t gfp)
-{
-    struct sk_buff *skb;
-
-    /* The TCP header must be at least 32-bit aligned. */
-    size = ALIGN(size, 4);
-
-    skb = alloc_skb_fclone(size + sk->sk_prot->max_header, gfp);
-    if (skb) {
-        skb->truesize += mem;
-        if (sk_stream_wmem_schedule(sk, skb->truesize)) {
-            /*
-             * Make sure that we have exactly size bytes
-             * available to the caller, no more, no less.
-             */
-            skb_reserve(skb, skb_tailroom(skb) - size);
-        }
-    }
-}
```

```

- return skb;
- }
- __kfree_skb(skb);
- } else {
- sk->sk_prot->enter_memory_pressure();
- sk_stream_moderate_sndbuf(sk);
- }
- return NULL;
-}
+struct sk_buff *sk_stream_alloc_pskb(struct sock *sk,
+ int size, int mem, gfp_t gfp);

static inline struct sk_buff *sk_stream_alloc_skb(struct sock *sk,
int size,
diff --git a/net/ipv4/tcp.c b/net/ipv4/tcp.c
index 8e65182..0dfda20 100644
--- a/net/ipv4/tcp.c
+++ b/net/ipv4/tcp.c
@@ -501,6 +501,33 @@ static inline void tcp_push(struct sock *sk, int flags, int mss_now,
}
}

+struct sk_buff *sk_stream_alloc_pskb(struct sock *sk,
+ int size, int mem, gfp_t gfp)
+{
+ struct sk_buff *skb;
+
+ /* The TCP header must be at least 32-bit aligned. */
+ size = ALIGN(size, 4);
+
+ skb = alloc_skb_fclone(size + sk->sk_prot->max_header, gfp);
+ if (skb) {
+ skb->truesize += mem;
+ if (sk_stream_wmem_schedule(sk, skb->truesize)) {
+ /*
+ * Make sure that we have exactly size bytes
+ * available to the caller, no more, no less.
+ */
+ skb_reserve(skb, skb_tailroom(skb) - size);
+ return skb;
+ }
+ __kfree_skb(skb);
+ } else {
+ sk->sk_prot->enter_memory_pressure();
+ sk_stream_moderate_sndbuf(sk);
+ }
+ return NULL;
+}

```

```
+
static ssize_t do_tcp_sendpages(struct sock *sk, struct page **pages, int poffset,
    size_t psize, int flags)
{
--
```

1.5.3.4

Subject: Re: [PATCH 1/2] Uninline the sk_stream_alloc_pskb

Posted by [Herbert Xu](#) on Thu, 29 Nov 2007 07:45:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Mon, Nov 26, 2007 at 08:14:12PM +0300, Pavel Emelyanov wrote:

> This function seems too big for inlining. Indeed, it saves

> half-a-kilo when uninline:

>

> add/remove: 1/0 grow/shrink: 0/7 up/down: 195/-719 (-524)

> function old new delta

> sk_stream_alloc_pskb - 195 +195

> ip_rt_init 529 525 -4

> __inet_lookup_listener 284 274 -10

> tcp_sendmsg 2583 2486 -97

> tcp_sendpage 1449 1305 -144

> tso_fragment 417 267 -150

> tcp_fragment 1149 992 -157

> __tcp_push_pending_frames 1998 1841 -157

>

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

Applied to net-2.6.25. Thanks Pavel!

--

Visit Openswan at <http://www.openswan.org/>

Email: Herbert Xu ~{PmV>Hl~} <herbert@gondor.apana.org.au>

Home Page: <http://gondor.apana.org.au/~herbert/>

PGP Key: <http://gondor.apana.org.au/~herbert/pubkey.txt>
