

---

Subject: Re: [PATCH 2/4] Add a \_\_GFP\_SLABMEMCG flag  
Posted by [Christoph Lameter](#) on Mon, 11 Jun 2012 14:24:37 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Sat, 9 Jun 2012, James Bottomley wrote:

> On Fri, 2012-06-08 at 14:31 -0500, Christoph Lameter wrote:  
> > On Fri, 8 Jun 2012, Glauber Costa wrote:  
> >  
> > > \*/  
> > > #define \_\_GFP\_NOTRACK\_FALSE\_POSITIVE (\_\_GFP\_NOTRACK)  
> > >  
> > > -#define \_\_GFP\_BITS\_SHIFT 25 /\* Room for N \_\_GFP\_FOO bits \*/  
> > > +#define \_\_GFP\_BITS\_SHIFT 26 /\* Room for N \_\_GFP\_FOO bits \*/  
> > > #define \_\_GFP\_BITS\_MASK ((\_\_force gfp\_t)((1 << \_\_GFP\_BITS\_SHIFT) - 1))  
> >  
> > Please make this conditional on CONFIG\_MEMCG or so. The bit can be useful  
> > in particular on 32 bit architectures.  
>  
> I really don't think that's at all a good idea. It's asking for trouble  
> when we don't spot we have a flag overlap. It also means that we're  
> trusting the reuser to know that their use case can never clash with  
> CONFIG\_MEMCG and I can't think of any configuration where this is  
> possible currently.

Flag overlap can be avoided using the same method as we have done with the page flags (which uses an enum). There are other uses of N bits after GFP\_BITS\_SHIFT. On first look this looks like its 4 right now so we cannot go above 28 on 32 bit platforms. It would also be useful to have that limit in there somehow so that someone modifying the GFP\_BITS sees the danger.

> I think making the flag define of \_\_GFP\_SLABMEMCG conditional might be a  
> reasonable idea so we get a compile failure if anyone tries to use it  
> when !CONFIG\_MEMCG.

Ok that is another reason to do so.

---