Subject: Re: [RFC][-mm] [1/2] Simple stats for cpu resource controller
Posted by Peter Zijlstra on Thu, 10 Apr 2008 16:25:11 GMT
View Forum Message <> Reply to Message

On Thu, 2008-04-10 at 21:39 +0530, Balaji Rao wrote:
> On Monday 07 April 2008 06:54:53 pm Peter Zijlstra wrote:
> > On Sun, 2008-04-06 at 02:01 +0530, Balaji Rao wrote:
> >
> > > > > > > +static s64 cpu_cgroup_read_stat(struct cpu_cgroup_stat *stat,
> > > > > > > +  enum cpu_cgroup_stat_index idx)
> > > > > > > +{
> > > > > > > + int cpu;
> > > > > > > + s64 ret = 0;
> > > > > > > + unsigned long flags;
> > > > >
> > > > > > > +
> > > > > > > + local_irq_save(flags);
> > > > >
> > > > I am just wondering. Is local_irq_save() enough?
> > > >
> > > Hmmm.. You are right.This does not prevent concurrent updates on other
> CPUs
> > > from crossing a 32bit boundary. Am not sure how to do this in a safe way.
> I
> > > can only think of using atomic64_t now..
> > >
> > > > > > > + for_each_possible_cpu(cpu)
> > > > > > > +  ret += stat->cpustat[cpu].count[idx];
> > > > > > > + local_irq_restore(flags);
> > > > > > > +
> > > > > > > + return ret;
> > > > > > > +}
> > > > > > > +
> >
> > So many stats to steal code from,.. but you didn't :-(
> >
> > Look at mm/vmstat.c, that is a rather complete example.
> >
> > The trick to solving the above is to use per cpu deltas instead, the
> > deltas can be machine word size and are thus always read in an atomic
> > manner (provided they are also naturally aligned).
> >
> >
> Hi Peter,
>
> This wont work for time based statistics. At nsec granularity, a word can hold
> a time value of up to ~4s.

4 seconds is plenty for a delta, most increments are in the ms range.

> I propose to solve this problem by using a lock to protect the statistics, but
> only on 32bit architectures.
>
> I'm not sure how good a solution this is, but that's the best I can think of
> ATM.

Not needed, keep per cpu word deltas and fold into a global u64 counter
while holding a lock.