kubernetes #72895

commit e3f4e1e37817c28c78500e071e8f8a6f77d78858

问题：当新的node在schedule和preemt两个阶段之间进入集群，scheduler的分配效率会下降

复现：无

原因：
pod在被scheduler分配到指定node之前，会进入一个pending队列。scheduler为队首pod选择node时（scheduler阶段），可能会因为资源紧张的因素无法选择一个合适的node，此时scheduler就会挑选node，选择其中合适的node并删除掉运行在它上面的一些优先级相对不高的pod（preemption阶段）。此时这个需要删除pod的阶段被标记成nominate，这个队首的pod会被标记成nominated然后放置到pending队列最后。

nominate node会删除一些pod来满足nominated pod，preemption阶段在开始删除之前，首先要选择出最时候删除的node称为nominate node，但是退出scheduler cycle之后，进入preemption选择nominate node之前，集群可能会因为加入了新的node或者突然资源充足起来，这时preemption就能够访问到更加适合的node（因为preemption阶段和scheduler阶段访问了不同的snapshot，即可能不同的集群node状态），这个新的node被选择成为nominate node，然后等待nominated pod重新排到队首，由scheduler重新检查后，进行分配。如果这是一个新加入的node（比如在k8s autoscaler的背景下），那么就会导致这个资源十分充足的node需要经过漫长的等待才能部署上除了nominated pod以外的其他pod（因为该node被标记成nominate了）。

In clusters with many pending pods with the same priority, if a pod is nominated, it goes back to the queue and behind all other pending pods with the same priority. This is important to avoid starvation of other pods, but it could also hold the nominated resources for a long time while the scheduler tries to schedule other pending pods in front of it. This scenario can happen and we cannot do much about it, but in the existing implementation of the scheduler, preemption updates its scheduler cache snapshot before starting the preemption work. If a node is added to the cluster or a pod is terminated, after a scheduling cycle and before the preemption logic starts its work, the preemption logic may find a feasible node without preempting any pods. The node becomes nominated in such cases and without preempting any pods. Now the nominated pod goes back to the queue and is placed behind other pods with similar priority, but none of those other pods can be scheduled on the node because there is a nominated pod for the node and the pod may take minutes before being retried if there are thousands of pending pods in the queue. To avoid such scenarios, we do not update the snapshot that the preemption logic uses and use the same one that its corresponding scheduling cycle has used.

修复：preemption和schedule阶段使用同样的snapshot，也就是防止preemption阶段scheduler访问到schedule阶段访问不到的node或者资源。

```
@@ -279,10 +286,6 @@ func (g *genericScheduler) Preempt(pod *v1.Pod,
nodeLister algorithm.NodeLister,
        if !ok || fitError == nil {
                return nil, nil, nil, nil
        }
-       err := g.snapshot()
-       if err != nil {
-               return nil, nil, nil, err
-       }
        if !podEligibleToPreemptOthers(pod, g.cachedNodeInfoMap) {
                klog.V(5).Infof("Pod %v/%v is not eligible for more
preemption.", pod.Namespace, pod.Name)
                return nil, nil, nil, nil
```