

kubernetes [#56622](#) issue [#56261](#)

commit aac60b6cbb330bd6c9ef37bc52ec607fe8f6298

问题： scheduler在调度时重复输出错误信息

复现： 在schedule环节移除一个node

原因： 当scheduler没有在自己的cache中找到对应的node（比如此时node已经离开cluster），就会从进入到处理错误的函数，默认的处理错误的函数是MakeDefaultErrorFunc，它会在报错之后进入重试环节。如果该node一直没有，就会一直报错，以至于重复输出错误消息

修复： 在默认处理错误的函数中，判断是否是没有找到node的错误，如果出现两次，就从cache中移除该node

```
diff --git a/plugin/pkg/scheduler/factory/factory.go
b/plugin/pkg/scheduler/factory/factory.go
index 764f449ccc..a52b8d69fc 100644
--- a/plugin/pkg/scheduler/factory/factory.go
+++ b/plugin/pkg/scheduler/factory/factory.go
@@ -1109,6 +1109,21 @@ func (factory *configFactory)
MakeDefaultErrorFunc(backoff *util.PodBackoff, pod
    } else {
        if _, ok := err.(*core.FitError); ok {
            glog.V(4).Infof("Unable to schedule %v %v: no
fit: %v; waiting", pod.Namespace, pod.Name, err)
+        } else if errors.IsNotFound(err) {
+            if errStatus, ok := err.(errors.APIStatus); ok
&& errStatus.Status().Details.Kind == "node" {
+                nodeName :=
errStatus.Status().Details.Name
+                // when node is not found, We do not
remove the node right away. Trying again to get
+                // the node and if the node is still not
found, then remove it from the scheduler cache.
+                _, err :=
factory.client.CoreV1().Nodes().Get(nodeName, metav1.GetOptions{})
+                if err != nil && errors.IsNotFound(err)
{
+                    node := v1.Node{ObjectMeta:
metav1.ObjectMeta{Name: nodeName}}
+                    factory.schedulerCache.RemoveNode(&node)
+                    // invalidate cached predicate
for the node
+                    if
factory.enableEquivalenceClassCache {
+                        factory.equivalencePodCache.InvalidateAllCachedPredicateItemOfNode(nodeName)
+                    }
                }
```

```
+  
+  
    } else {  
        glog.Errorf("Error scheduling %v %v: %v;  
retrying", pod.Namespace, pod.Name, err)  
    }
```