kubernetes #68984 issue #68899

commit 78f6484e14930cb827449637772198e8c7907f03

问题：在集群升级到1.13 release版本时，pod被错误分配到了master节点上，然后在该pod请求日志时被master节点拒绝。

原因：在版本1.13时，决定一个node是否能被分配pod (schedulable)是通过taint来实现的，而之前的版本是通过condition来（即node.Spec.Unschedulable）来实现的。

出现这个问题的root cause是，当kubelet在升级到1.13时，该node已经在1.11的版本中注册过了，因此node.Spec.Taints没有更新（注意此时的controller manager还是1.11版本，它不具备添加unschedule taint的能力），包括unschedulable的taint。而此时scheduler已经被更新到1.13了，TaintNodesByCondition是默认开启的，而NodeConditionCheck是默认关闭的，因此它忽略了node.Spec.Unschedulable。此时controller manager也升级到1.13了，它会根据node的configuration添加污点（node.Spec.Unschedulable）

因为node.Spec.Taints（包括node.Spec.Unschedulable）没有在kubelet更新时更新，因此必须要等到controller manager更新时才能更新，因此就可能晚于scheduler的更新，从而造成该node没有node.Spec.Unschedulable时被scheduler分配给了pod。

修复：保持向后兼容性，在1.13之前的版本既判断node.Spec.Unschedulable又判断pod.Spec.Tolerations是否容忍algorithm.TaintNodeUnschedulable。而在1.13版本开始去掉node.Spec.Unschedulable的判断。

```
@@ -1470,7 +1470,14 @@ func CheckNodeUnschedulablePredicate(pod *v1.Pod,
meta algorithm.PredicateMetada
                return false,
[]algorithm.PredicateFailureReason{ErrNodeUnknownCondition}, nil
        }

-       if nodeInfo.Node().Spec.Unschedulable {
+       // If pod tolerate unschedulable taint, it's also tolerate
`node.Spec.Unschedulable`.
+       podToleratesUnschedulable :=
v1helper.TolerationsTolerateTaint(pod.Spec.Tolerations, &v1.Taint{
+               Key:    algorithm.TaintNodeUnschedulable,
+               Effect: v1.TaintEffectNoSchedule,
+       })
+
+       // TODO (k82cn): deprecates `node.Spec.Unschedulable` in 1.13.
+       if nodeInfo.Node().Spec.Unschedulable &&
!podToleratesUnschedulable {
                return false,
[]algorithm.PredicateFailureReason{ErrNodeUnschedulable}, nil
```

```
}
```