

1. Overview

1.1 What is SR-IOV

Single Root I/O Virtualization (SR-IOV) is a PCI Express Extended capability which makes one physical device appear as multiple virtual devices. The physical device is referred to as Physical Function (PF) while the virtual devices are referred to as Virtual Functions (VF). Allocation of the VF can be dynamically controlled by the PF via registers encapsulated in the capability. By default, this feature is not enabled and the PF behaves as traditional PCIe device. Once it's turned on, each VF's PCI configuration space can be accessed by its own Bus, Device and Function Number (Routing ID). And each VF also has PCI Memory Space, which is used to map its register set. VF device driver operates on the register set so it can be functional and appear as a real existing PCI device.

2. User Guide

2.1 How can I enable SR-IOV capability

The device driver (PF driver) will control the enabling and disabling of the capability via API provided by SR-IOV core. If the hardware has SR-IOV capability, loading its PF driver would enable it and all VFs associated with the PF.

2.2 How can I use the Virtual Functions

The VF is treated as hot-plugged PCI devices in the kernel, so they should be able to work in the same way as real PCI devices. The VF requires device driver that is same as a normal PCI device's.

3. Developer Guide

3.1 SR-IOV API

To enable SR-IOV capability:

```
int pci_enable_sriov(struct pci_dev *dev, int nr_virtfn);
```

'nr_virtfn' is number of VFs to be enabled.

To disable SR-IOV capability:

```
void pci_disable_sriov(struct pci_dev *dev);
```

To notify SR-IOV core of Virtual Function Migration:

```
irqreturn_t pci_sriov_migration(struct pci_dev *dev);
```

3.2 Usage example

Following piece of code illustrates the usage of the SR-IOV API.

```
static int __devinit dev_probe(struct pci_dev *dev, const struct pci_device_id
```

```

*id)
{
    pci_enable_sriov(dev, NR_VIRTFN);

    ...

    return 0;
}

static void __devexit dev_remove(struct pci_dev *dev)
{
    pci_disable_sriov(dev);

    ...
}

static int dev_suspend(struct pci_dev *dev, pm_message_t state)
{
    ...

    return 0;
}

static int dev_resume(struct pci_dev *dev)
{
    ...

    return 0;
}

static void dev_shutdown(struct pci_dev *dev)
{
    ...
}

static struct pci_driver dev_driver = {
    .name = "SR-IOV Physical Function driver",
    .id_table = dev_id_table,
    .probe = dev_probe,
    .remove = __devexit_p(dev_remove),
    .suspend = dev_suspend,
    .resume = dev_resume,
    .shutdown = dev_shutdown,
};

```