

ZFS – Ein Dateisystem mit Raid-integration

Im Folgenden werde ich mich von dem Raid, wie ich es auf den letzten Seiten beschrieben habe distanzieren. Es hat viele Probleme, die von einigen Dateisystemen gelöst wurden und ist veraltet. Das heißt nicht, dass die letzten neun Seiten keinen Zweck erfüllt hätten. Die Dateisysteme, die den Platz von Raid in den letzten Jahren übernommen haben nutzen die selben Techniken. Allerdings erweitern sie diese um ein Dateisystem mit vielen weiteren Funktionen.

Ich nutze ZFS hier als Beispiel. Das Dateisystem btrfs würde diese Funktionen zum Beispiel auch haben.c

Prüfsummenbildung

Das wohl wichtigste Merkmal von ZFS ist die Prüfsummenbildung, die es vor BitRot schützt. Wenn auf einem konventionellen Raid 1 eine Festplatte falsche Daten zurück gibt erkennt dies keiner. Es gibt keine Prüfung und der Raid-Controller kann nicht die Daten von der anderen Festplatte nehmen. Erstens merkt er nicht, dass verschiedene Daten vorliegen und zweitens könnte er sich im Zweifel für keine der Daten entscheiden.

ZFS berechnet Prüfsummen für jeden Stripe. Das heißt, dass beim lesen der ganze betroffene Stripe und die Prüfsumme(die meist immer im Ram gehalten wird) geladen werden muss. Das könnte theoretisch ein Performance-problem beim Lesen von vielen kleinen Daten ergeben, funktioniert in der Praxis aber sehr gut.

Der Vorteil ist, dass ZFS erkennen kann wenn eine Festplatte falsche Daten zurück gibt. Die betroffene Festplatte kann dann aus dem Verbund geworfen werden.

Copy-onWrite

Dies ist zwar kein Problem von Raid, denn Raid ist kein Dateisystem und muss sich darum nicht kümmern, aber Copy on Write ist trotzdem ein sehr wichtiges Feature für ZFS.

Es erlaubt ZFS mehrfach angelegt Dateien nicht mehrfach zu speichern. Es wird erst eine neue Kopie der Daten angelegt, wenn eine der Dateien verändert wurde.



Illustration 5: https://www.andybreuhan.de/wp-content/uploads/2009/03/zfs_feature_2.jpg