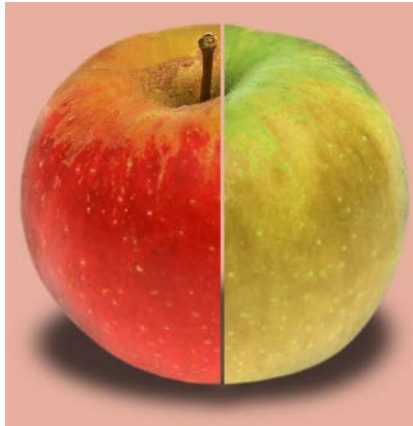




Konsistenter und kürzer

(C++-Tricks: Generischer Code mit Type Traits; iX 10/2020, S. 150)

Wenn man Boolean Type Traits definiert, ist es besser, von `std::true_type` oder `std::false_type` zu erben. Das Resultat: weniger Code und konsistent im Verhalten mit den Standard Traits. Wer Code sparen will, nutzt aber besser gleich `constexpr-bool`-Variablen-Templates:

```
template <typename T>
constexpr bool mytrait { false };

template <>
constexpr bool mytrait<peter> { false };
```

Leider sind diese nicht gleich als Concepts genommen worden. Dies hatte historische Gründe, der Concepts-TS basierte auf C++11 und hatte noch keine variablen Templates.

PETER SOMMERLAD, VIA E-MAIL

Vielen Dank für das aufmerksame Lesen und Ihren Hinweis. Das wären auch mögliche Implementierungen gewesen, sogar konsistenter und kürzer. Und das mit den Concepts ist echt schade.

(Detlef Wilkening)



Schludrig, da verführerisch einfach

(Netzvirtualisierung: VXLAN für standortübergreifende Layer-2-Vernetzung; iX 9/2020, S. 116)

Eine standortübergreifende L2-Vernetzung – egal über welches Transportmedium – kann tückisch sein. Ja, VMs ohne

langes Nachdenken hin und her zu schieben ist verführerisch einfach. Dass sich dies früher oder später rächt, sieht man an anderen Beispielen im schludrigen Umgang mit IT, zum Beispiel bei Sicherheitslücken und Standardpasswörtern.

Ich empfehle die Lektüre der entsprechenden Artikel von Ivan Pepelnjak (siehe blog.ipospace.net). Er hat langjährige Erfahrung mit diesem Thema und so manche Katastrophe miterlebt, die durch Split-Brain und andere Leckereien entstanden ist.

Vieles davon mag den Leser zur Annahme verleiten, dass das eigene Unternehmen so groß nicht sei und sich der Aufwand nicht lohnen würde. Sicher sind die aufgezeigten Szenarien nicht immer direkt umsetzbar, speziell für kleinere Umgebungen. Aber wenn man sich für eine Lösung

Der direkte Draht zu



Direktwahl zur Redaktion: 0511 5352-387

Redaktion iX | Postfach 61 04 07
30604 Hannover | Fax: 0511 5352-361
E-Mail: post@ix.de | Web: www.ix.de

www.facebook.com/ix.magazin
twitter.com/ixmagazin (News)
twitter.com/ix (Sonstiges)

Für E-Mail-Anfragen zu Artikeln, technischen Problemen, Produkten et cetera steht die Redaktion gern zur Verfügung.

post@ix.de	Redaktion allgemein
akl@ix.de	Alexandra Kleijn
ane@ix.de	Alexander Neumann
avr@ix.de	André von Raison
csc@ix.de	Carina Schipper
cle@ix.de	Carmen Lehmann
fo@ix.de	Moritz Förster
jd@ix.de	Jürgen Diercks
mde@ix.de	Madeleine Domogalla
mfe@ix.de	Markus Feilner
map@ix.de	Matthias Parbel
mm@ix.de	Michael Mentzel
nb@ix.de	Nicole Bechtel
odi@ix.de	Dr. Oliver Diedrich
rme@ix.de	Rainald Menge-Sonnentag
sih@ix.de	Silke Hahn
sun@ix.de	Susanne Nolte
un@ix.de	Bert Ungerer
urg@ix.de	Ute Roos

Listing-Service:

Sämtliche in iX seit 1990 veröffentlichten Listings sind über den iX-FTP-Server erhältlich: ftp.heise.de/pub/ix/

entscheidet, ist es hilfreich, das auf einer informierten Basis zu tun.

Wie schnell ist aus einer kleinen Umgebung eine große und kritische geworden, ohne dass man sich die Zeit zum Nachdenken genommen hat, ob das organisch Gewachsene mit den Geschäftsanforderungen noch Schritt halten kann. Und das nicht im Regelbetrieb, sondern wenns mal kracht. Oder um es mit den Worten von Ivan zu sagen: Es ist nicht die Frage, ob eine Umgebung mit Stretched-L2-VLANs auf die Bretter geht, sondern wann.

PATRIK SCHINDLER, VIA E-MAIL

Natürlich kann standortübergreifende L2-Vernetzung tückisch sein. Jeder Administrator, der sich bereits mit Layer-2-Schleifen beschäftigt hat, kann dies sicher bestätigen. Gerade bei reinem Spreizen der VLANs können sich Layer-2-Schleifen gravierend auswirken. VXLAN auf Basis eines grundlegenden Layer-3-Underlays stellt lediglich eine Abmilderung der Risiken dar. Jedoch nimmt hierdurch auch die Komplexität zu.

Es sollte grundlegend immer eine genaue Prüfung erfolgen, ob, wo und warum eine Vergrößerung der Layer-2-Domäne stattfinden soll. Sollten keine zwingenden Gründe bestehen, empfiehlt sich in den meisten Fällen eine reine Layer-3-Umgebung. Sollte es jedoch – aus welchem Grund auch immer – erforderlich sein, bietet VXLAN auf Basis von MP-BGP EVPN eine interessante Alternative zum klassischen Spreizen der VLANs, allerdings bei gleichzeitiger Steigerung der Komplexität. (Benjamin Pfister)



Fehler im C++-Listing

(C++-Tricks: auto als Non-Type-Templateparameter; iX 9/2020, S. 128)

Ich verfolge mit großem Interesse die C++-Serie und lerne mit jedem Artikel etwas Neues. Im letzten Artikel hat sich ein

kleiner Fehler eingeschlichen: Das `assert` im Listing 8 bewirkt, dass `DeleteFn` nur im Debug-Build aufgerufen, im Release aber herausgekürzt wird.

Möglich, dass das `fopen` scheitert, weil die Datei nicht erreichbar ist. In diesem Fall sollte `fclose` nicht aufgerufen werden. Es gibt Implementierungen der C `stdlib`, die bei `fclose(NULL)` auf den Terminate Handler zurückfallen. Mir ist unbekannt, ob der C-Standard das regelt.

Dieser Umstand kommt hier aber nicht zum Tragen, da `unique_ptr` den Deleter nur aufruft, wenn `get()!=nullptr` gilt. Damit konnte ich den Gedanken verwerfen, `fclose` von einer unnötigen Bedingung abhängig zu machen.

THOMAS HAENEL, VIA E-MAIL

Sie haben recht, `assert` wird normalerweise im Release-Build deaktiviert. Das Verhalten lässt sich durch Entfernen der Definition `NDEBUG` kontrollieren. Generell ging es mir weniger darum, `assert` als solches zu verwenden, sondern die Möglichkeit aufzuzeigen, hier eine Prüfung vorzunehmen. Wegen exakt der Situation, die Sie erkannt haben, nutze ich selbst ungern `assert`. Ich ziehe feingranularere eigene Funktionen vor. Einige von diesen `assert`-ähnlichen Funktionen werden dann im Release-Build deaktiviert, andere bleiben bestehen.

(Andreas Fertig)

Ergänzungen und Berichtigungen

SAP: Innige Verbindung: HANA auf Power-Systemen; iX 10/2020, S. 98

Zwischen Heftschluss und Erscheinen des Artikels veralteten leider die letzten beiden Absätze. IBM hat mittlerweile seinen Power10 vorgestellt und Intel musste aufgrund massiver Probleme in der Fertigung seine 7-nm-Prozessoren bis 2023 verschieben.

Embedded-Software: C++-Metaprogrammierung mit Templates für eingebettete Systeme; iX 10/2020, S. 144

Die aktuellste Version der Codebeispiele können Leser unter <https://github.com/Fraunhofer-IIS-ARC-WST/compile-time-interrupt-dispatching-with-templates> herunterladen. Die Entwickler freuen sich natürlich auch über Pull Requests.

Die iX-Redaktion behält sich Kürzungen und auszugsweise Wiedergabe der Leserbriefe vor. Die abgedruckten Zuschriften geben ausschließlich die Meinung des Einsenders wieder, nicht die der Redaktion.