BIND9.11について聞きたいこと(事前質問)

本日のMukund Sivaramanさんからの発表をよりよく理解できるように、事前にいくつか質問をしてみました。
以下、質問事項と回答(およびその超訳)です。
回答いただいたMukundさんおよびCathyさんに感謝します。

BIND 9.11 new features especially we'd like to know (use cases, configuration tips, background, etc)	BIND9.11の新機能で特に聞きたいこと (ユースケース、設定の小技、背景など)
Catalog zones	Catalog zones
The catalog zones draft describes the use-case for this feature: https://tools.ietf.org/html/draft-muks-dnsop-dns-catalog-zones-01 We spent a lot of time researching various approaches for zone and zone data provisioning and catalog zones is one of the features that resulted from it. Many other changes were made within BIND 9.11 out of this research too (rndc addzone/modzone, etc. changes, DynDB, NZF database backend changes, etc.). For configuration of catalog zones, please see chapter 4 (Advanced DNS features) in the BIND administrator reference manual.	Catalog zonesのユースケースは以下のインターネットドラフト(I-D)に記載されています。 https://tools.ietf.org/html/draft-muks-dnsop-dns-catalog-zones-01 ISCではゾーンおよびゾーンデータの提供について、多くの時間をかけてさまざまな角度から調査をしてきました。Catalog zoneはその結果に基づく機能の一つです。他にも、この調査の結果は多数BIND9.11に反映されています(rndc addzone/modzoneなどの変更、DynDB、NZFデータベースバックエンドの変更、など)。 Catalog zoneの設定はBIND ARMの第4章(高度なDNS機能)を参照してください。また、以下も参照してください。 http://jpmens.net/2016/05/24/catalog-zones-are-coming-to-bind-9-11/
Minimal response to 'any' queries	Minimal response to 'any' queries
My colleague Cathy wrote a reply to this question: It's an implementation of https://tools.ietf.org/html/draft-ietf-dnsop-refuse-any-03 minimal-any is the related config option, that also describes behavior. I don't know if it implements the synthesised HINFO for CNAME hack (also detailed in the draft) - this perhaps needs mention somewhere in the ARM - although we're still at draft stage with this one - no agreed and published RFC yet.	同僚のCathyさんが答えてくれました。 これは以下のI-Dの実装です。 https://tools.ietf.org/html/draft-ietf-dnsop-refuse-any-03 minimal-anyは関連オプションで、挙動を設定します。 CNAMEハック用のHINFO合成(上記I-Dに詳細が記載されています)については、まだ議論がまとまっていないので、実装されていません(そのはず)。

SERVFAIL caching

This feature is useful to avoid overloading an authoritative server that is already SERVFAIL'ing. servfail-ttl is the config option used to set the duration that a SERVFAIL response is cached. The default was initially considered as 10 seconds but was finalized to 1 second. A 0 setting turns off the SERVFAIL cache.

SERVFAIL caching

これは既にSERVFAILを返した権威サーバに繰り返し負荷をかけないための有用な機能です。servfail-ttlはSERVFAIL応答をキャッシュする期間を設定するオプションで、デフォルトは1秒です(最初は10秒を考えていましたが)。A 0を設定するとSERVFAIL応答キャッシュを無効にします。

Questions to BIND 9.11 new features

Why fetches-per-server and fetches-per-zone are enabled by default?

My colleague Cathy wrote a reply to this question:

They're not enabled by default – the default value is zero, which disables both of these. If the question is "why aren't they enabled by default?" then it's because they are an optional feature and unless there is a compelling reason to change BIND's default behaviour, we generally don't – particularly when this might result in some client queries being dropped or SERVFAILed.

BIND9.11新機能への質問

どうしてfetches-per-serverとfetches-per-zoneはデフォルトで有効になったのですか？

同僚のCathyさんが答えてくれました。

それらはデフォルトでは有効になっていません。デフォルト値は、それらを無効にする値である0になっています。質問の意図が「なぜデフォルトで有効になっていないのですか？」ということであれば、それはオプション機能だからです。BINDのデフォルトの挙動を変更するには相応の理由が必要で、通常ISCは変更しません。特に、クライアントからのクエリが落とされたりSERVFAILになるような場合は。

What is the expected use case of DynDB?	DynDBはどのようなユースケースを想定していますか？
There is an "object-oriented" interface (in C) called dns_db in libdns library. It is a database abstraction layer which is used to store and query the data corresponding to a zone and also the data corresponding to a cache. BIND uses something called an RBTDB (Red Black Tree DataBase – which actually is a Red Black Forest) to store zones and cache in memory which is a dns_db interface implementation. So far, the only dns_db implementations supported were statically linked ones such as RBTDB, but we got requests to make this interface reusable so that other backends could be written for this dns_db interface for storing zone data. DynDB stands for dynamic DB. With this feature, named is able to load dns_db implementations from .so files at runtime, write zone data into backends using them and query zone data from these backends. We do not currently ship any dns_db implementations (called DynDB drivers) with BIND, but we hope that others will contribute drivers for various databases, etc. We know of 3rd parties that use DynDB to serve zone data from an LDAP backend (which is why this feature was added). You can find some more information about this feature here: https://kb.isc.org/article/AA-01420/0/What-is-dyndb-and-how-is-it-better-than-DLZ.html	これはdns_dbと呼ばれるオブジェクト指向のインターフェース(C言語)で、libdnsライブラリに含まれます。データベースの抽象化層で、ゾーンやキャッシュに関連するデータを格納したり参照したりするのに使われます。 BINDはゾーンやキャッシュをメモリへ格納するためにRBTDB(Red Black Tree DataBase; 赤黒木データベース)と呼ばれるものを使っており、dns_dbはそのインターフェースの実装です。 これまで、dns_db実装のみがRBTDBに静的にリンクされていましたが、このインターフェースを再利用可能にして欲しいという要望がISCに寄せられました。dns_dbインターフェースを通じて別のバックエンドにゾーンデータを格納できるようにして欲しいということです。 DynDBは動的DBを意味します。この機能では、namedが実行時に.soファイルからdns_db実装を読み込めるようになり、ゾーンデータを別のバックエンドに書き込み、そのバックエンドからゾーンデータを参照できるようになります。 現状では、BINDにはDynDBドライバと呼ばれるdns_dbの実装は1つも付属していませんが、第三者がさまざまなデータベース用のドライバを提供してくれることを期待しています。 ISCは、DynDBでLDAPバックエンドを使う第三者を知っています(そしてそれがこの機能追加の理由です)。 この機能については、さらに以下を参照してください。 https://kb.isc.org/article/AA-01420/0/What-is-dyndb-and-how-is-it-better-than-DLZ.html
Will nsec/nsec3 aggressive use be implemented in BIND9.11?	nsec/nsec3 aggressive useはBIND9.11に実装されますか？
For now, it is planned to be implemented as part of the 9.12 development cycle. As this is a very important feature that users are looking to deploy quickly, we may backport it to the 9.11 and 9.10 branches when it is ready.	今のところそれは9.12の開発サイクルの中で実装することを計画しています。しかし、ユーザが早く利用したいと望んでいるとても重要な機能であるため、9.12で準備ができ次第、9.11および9.10にも取り込むつもりです。

Other questions regarding to BIND9

Why the default number of UDP listeners are differ from version to version?

A couple of years ago, the default number of UDP listeners used to match the number of detected processors. We got a bug report that on Red Hat Enterprise Linux, BIND performance was lower due to this setting and that a lower count of listeners improved performance. So a BIND engineer lowered it to half the number of processors detected at that time.

In the past year, we discovered that this performance problem was actually due to a Red Hat Enterprise Linux kernel bug which was long fixed, and that the lower UDP listeners count was actually harming BIND performance for everyone quite significantly. So we changed it back to correspond roughly to number of detected processors − 1, which improved performance.

You can find some information about this setting here:
https://kb.isc.org/article/AA-01249/0/UDP-Listeners-choosing-the-right-value-for-U-when-starting-named.html

Which version will be next ESV?

We still have not decided, but we will need a new ESV next winter and 9.11 is possibly going to become the candidate branch for that, as 9.12 will not be ready by then. Please don't take this as a guarantee − we still have not decided. :)

BIND9に関するその他の質問

UDP listenersのデフォルト値がバージョンによって異なるのはなぜですか？

数年前は、UDP listenersの値は検知したプロセッサ数に適するようにしていました。RHELではこの設定によってBINDのパフォーマンスが低下し、値を低くすると改善したというバグレポートを受け、BINDエンジニアはそれ以後、検知したプロセッサ数の1/2に適するように値を変更(低く)しました。

昨年、ISCはこのパフォーマンス問題はかなり前に修正されたRHELのカーネルバグに起因していること、およびUDP listenersの値を低くすることはBINDパフォーマンスに重大な悪影響を与えていることに気づきました。そのため、その値を検知したプロセッサ数-1に適するように変更し、パフォーマンスを改善しました。

この設定に関する情報は以下を参照してください。
https://kb.isc.org/article/AA-01249/0/UDP-Listeners-choosing-the-right-value-for-U-when-starting-named.html

次のESVバージョンはどれになりますか？

まだ決まっていませんが、来年の冬までには次のESVが必要なため、9.11が候補になるでしょう(9.12の準備が間に合いそうにないので)。ただ、この回答をもって保証としないでください、本当にまだ決めていないので :)