

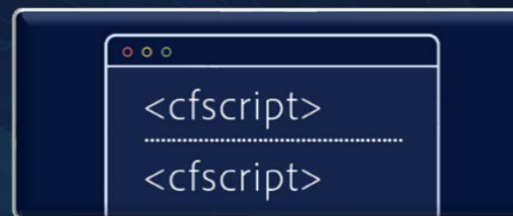


ColdFusion 2018 リリース セミナー

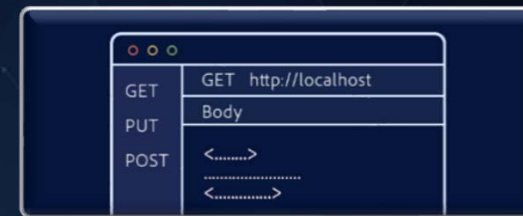
ColdFusion 2018 リリース トピック



パフォーマンスモニター
ツールセット

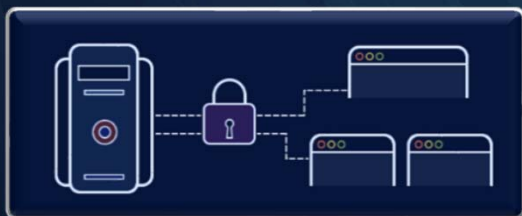
A screenshot of a code editor window showing CFML code. The code consists of two lines: `<cfscript>` followed by a dotted line, and another `<cfscript>` line below it. The editor has a dark background and a light-colored border.

CFML言語強化



Rest Playground

Cf



サーバー自動ロックダウン



その他

- Admin UI 改良
- CLI 機能追加
- キャッシュ強化
- メンバ関数強化

パフォーマンスモニターツ
ールセット (以下PMT)

PMTとは？

- サーバーモニターの後継
 - 専用インストーラーで提供
 - プログラム本体とデータストア (Redis)
- ColdFusionとは別のサーバーにインストール可能
 - モニターするサーバーを登録すると、常に計測する事になるので、負荷等を考慮すると別サーバーにインストールすることを推奨
- Standard版ユーザーも使用可能

サーバーモニタ (概要)

- コネクタやサーバーの状態、応答時間など



トポロジ (状態を可視化)

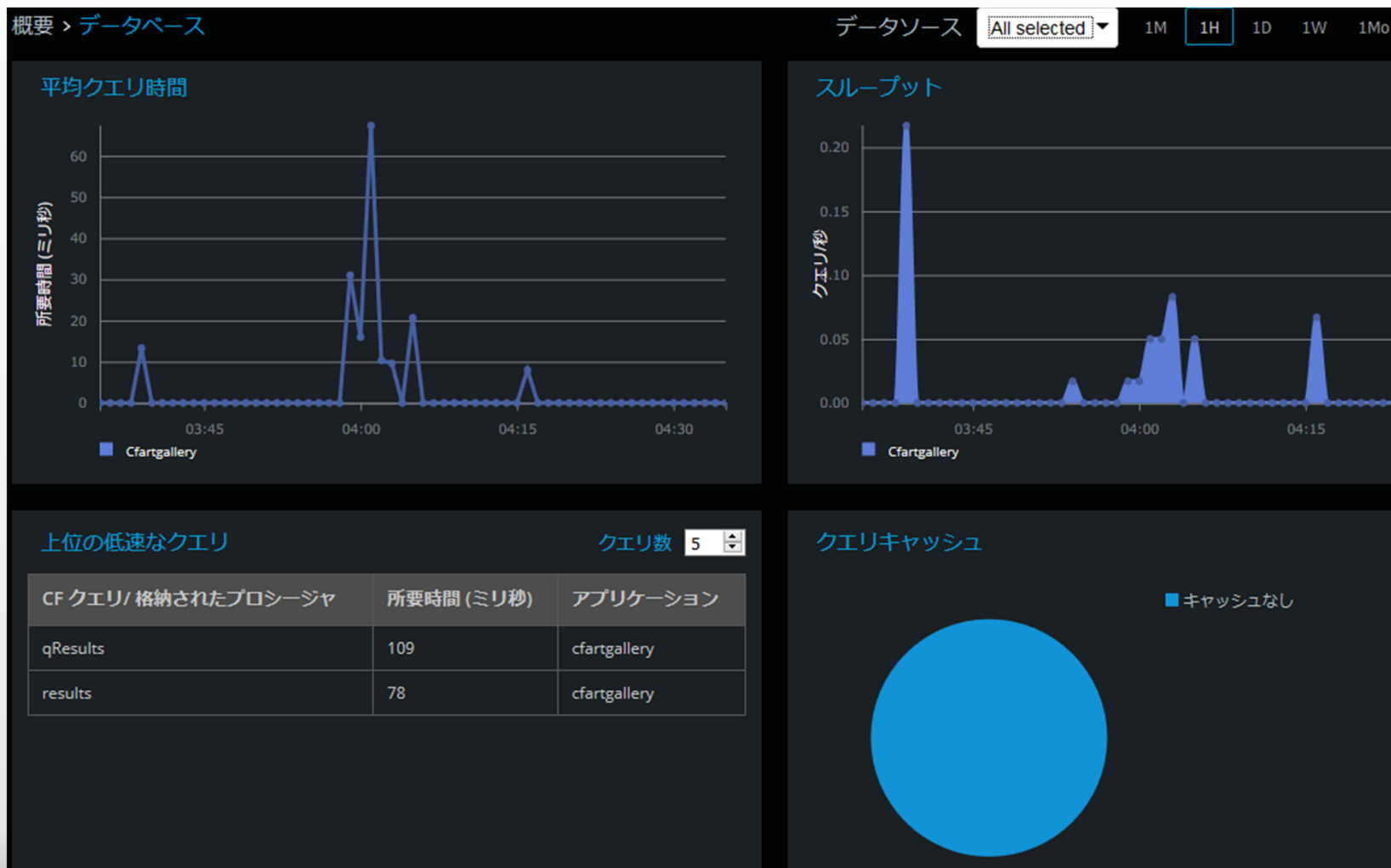
- Webサーバー ⇄ CF ⇄ DB、外部サービスなど



トポロジ (続き)



データベースモニタ

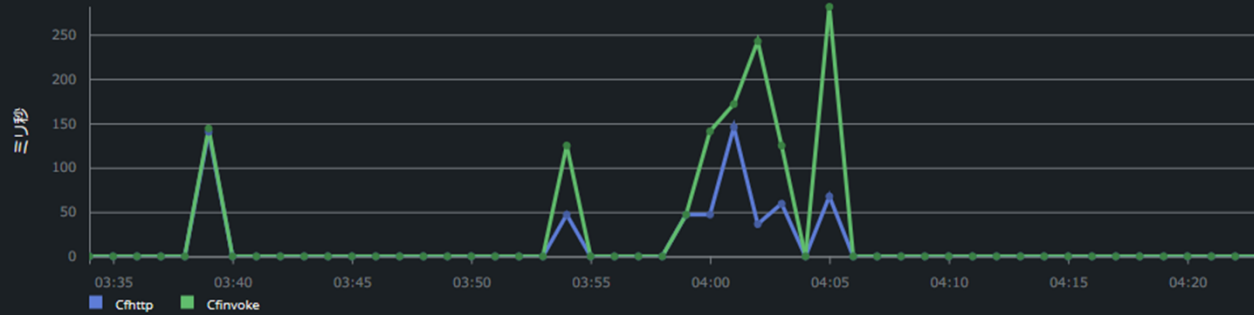


外部サービスモニタ

概要 > 外部サービス

概要 CFHTTP CFFTP CFFEED CFLDAP CFMAIL CFPOP CFFILE CFINVOKE CFSHAREPOINT

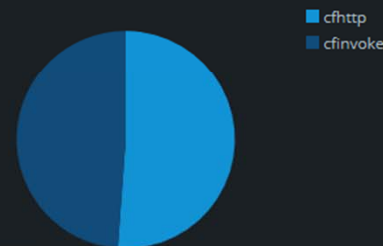
平均実行時間



上位の低速なサービス

サービス	最も遅い実行(ミリ秒)
cfinvoke	166.76
cfhttp	85.86

低速なサービスの分布



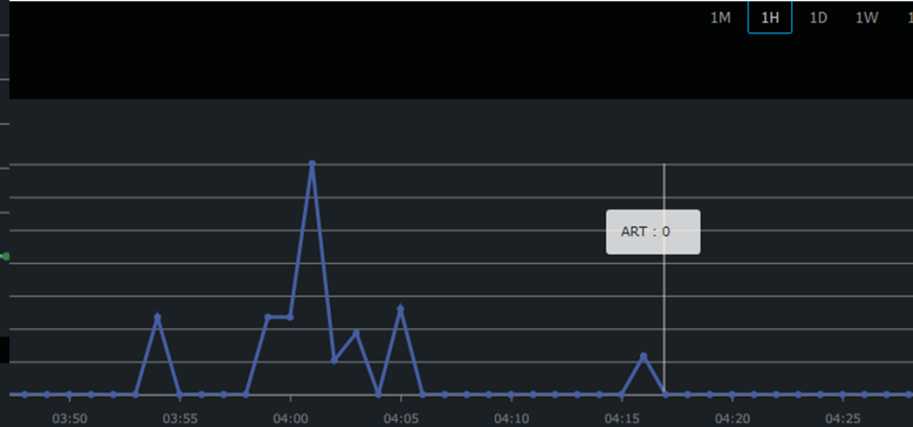
http://192.168.2.163/rest/cfartgallery/artService/1	234
http://192.168.2.163:8500/rest/cfartgallery/artService/1	125
http://192.168.2.163/rest/cfartgallery/artService/3	109
	46
	31

URL 数 5

コンテンツの分布



HTTP メソッド



コードプロファイラ

- トランザクションを監視し変数やフロー、応答時間、メモリを取得

The screenshot displays a code profiler interface. At the top, a bar shows the total execution time for the transaction as 32 ms. This bar is broken down into segments: 'データベース' (Database) in blue (0 ms), '外部サービス' (External Service) in green (0 ms), and 'サーバー' (Server) in orange (32 ms). Below this, the URL 'http://192.168.2.163:8500/rest/cfartgallery/artService/1' is shown with a toggle switch and an 'Add to Profile' button. A navigation menu includes '基本情報', 'コードフロー', 'CFサーバー', 'データベース', '外部サービス', 'タグ/関数', 'エラー/タイムアウト', and 'メモリ'. The '基本情報' (Basic Information) section is expanded, showing a table of request details.

HTTP メソッド	GET	ステータスコード	200
URI	http://192.168.2.163:8500/rest/cfartgallery/artService/1	応答時間	32 ms
クライアント IP	192.168.2.143	アプリケーション	cfartgallery
クライアントホスト	192.168.2.143	リクエストタイプ	WEB SERVICE REQUEST
テンプレートパス	C:\inetpub\sites\alias\cfdemo\cfartgallery\artService.cfc	Java スレッド ID	http-nio-8500-exec-10
リクエスト開始時刻	27 Jul 2018, 2:40:05.178 pm	インスタンス ID	cfwindows:cfusion:8500
リクエスト終了時刻	27 Jul 2018, 2:40:05.210 pm		

コードプロファイラ (続き)

タグ	平均所要時間(ミリ秒)
cfcontent	0
cfquery	0
cfqueryparam	0

関数	平均所要時間(ミリ秒)
onRequestStart	0
getArt	16
SerializeJSON	0

変数	タイプ	関数	スコープ	サイズ(バイト)
description	① SIMPLE	getArt	REQUEST	72
id	① SIMPLE	getArt	REQUEST	16
name	① SIMPLE	getArt	REQUEST	56
results	① QUERY		REQUEST	283
CFQUERY.EXECUTIONTIME	① SIMPLE		REQUEST	16
THIS	①	onRequestStart	REQUEST	168
__IMPLICITARRAYSTRUCTVAR0	①	getArt	REQUEST	208
ITEM	① STRUCT	getArt	REQUEST	208

- 一部タグ (cfif系、cftry系、cfset) は対象外
- プロファイラが可能な関数は一部に限定
- プロファイラを実行する際はサーバーの負荷に注意する
 - 一般的にメモリプロファイラなどはそれなりに負荷が掛かる

自動コネクタチューニング (IIS)

コネクタの自動調整

自動調整を有効にする

自動調整のしきい値 (パーセンテージ) 90

プールの増加率 20

最大プールサイズ 2000

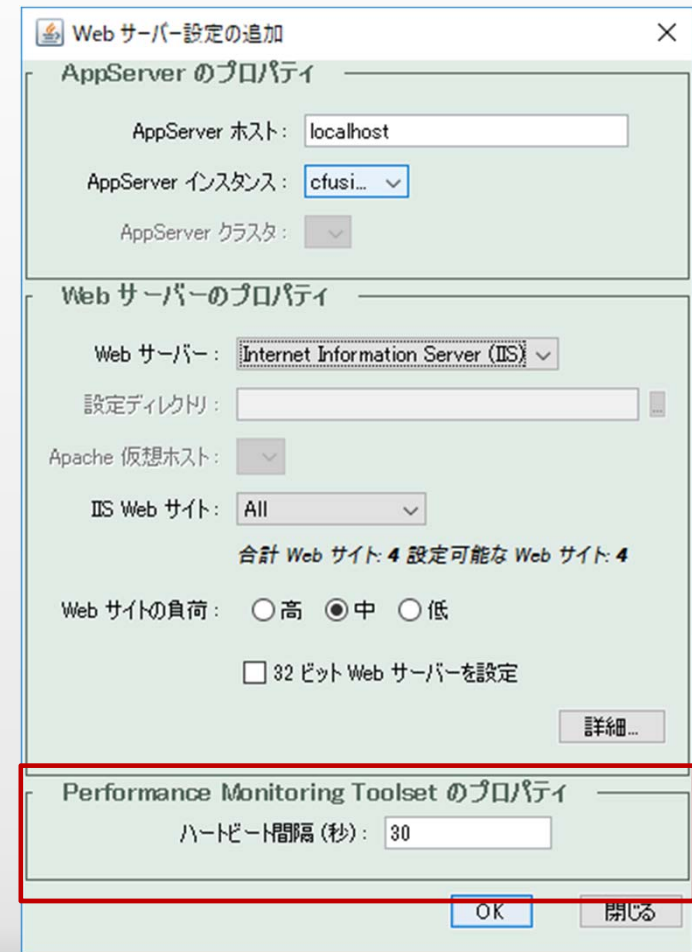
大量のトラフィックに対して自動でプールサイズの増加が可能

←プールサイズの90%がビジーになると、プールサイズが20%増加

自動チューニング中は、プールサイズは自動的に増加するためIISを再起動は必要ない

Webサーバーコネクタ

- PMT用のオプションが追加
 - ハートビート間隔
 - コネクタの負荷データをColdFusionに送信する間隔（秒）



その他

- ColdFusionノードの自動検出
- コネクタの監視
- 外部サービスの監視
- 外部サービスからCFへのアクセスの監視
- スレッドダンプ、ヒープダンプ、GCの実行
- コードプロファイルの結果を Builderで読み込み一覧とソース位置の表示
- アラートの設定

サーバー自動ロックダウン

自動サーバーロックダウン

- ロックダウンガイドの自動化
 - 専用のインストーラーを提供
 - エラーが発生しやすい手動操作からロックアウトを自動化
 - セキュリティを高めたサイト設定、ファイル権限、実行ユーザーなどの設定を行う
 - 外部向けWebアプリケーションサービスの公開にあたっての、推奨される設定
 - これまではガイドに沿って、手動によるロックダウン設定を行う必要があった

自動サーバーロックダウンを実行する前提

■ 必須要件:

- CF2018インストール時に「本番プロファイル」または「本番プロファイル+セキュアプロファイル」を選択
- 初期セットアップ済み
 - インストール後の初期セットアップ（移行）ウィザードを実行し、ColdFusion Administrator にアクセス済み
- Apache / IIS などのWeb サーバーをインストール済み

インストール画面の例

Adobe® ColdFusion® 2018 Server Auto-Lockdown

OS 管理者アカウントの詳細

OS 管理者ユーザーアカウントの詳細を入力します。このユーザーには、ColdFusion ディレクトリと Web サーバーの Web ルート / ドキュメントルートに対する完全なファイルシステム権限が付与されます。

システム管理者ユーザー
Administrator

システム管理者のパスワード
[]

ドメイン
WIN-6HK3UAQ2SAU

InstallAnywhere
キャンセル 戻る(P) 次へ(N)

Adobe® ColdFusion® 2018 Server Auto-Lockdown

ColdFusion ランタイムユーザー

ロックダウン後に ColdFusion を実行するユーザーアカウントを入力します。このユーザーアカウントには、ColdFusion ディレクトリと Web サーバーの Web ルート / ドキュメントルートに対するファイルシステム権限が付与されます。

CF サービスを実行するためのユーザーを作成しましたか？
 はい
 いいえ
新規ユーザーの詳細を入力します。

ユーザー名
cfuser

InstallAnywhere
キャンセル 戻る(P) 次へ(N)

インストーラーで行える設定

- 最新のUpdateの適用（可 / 非）
- Webサーバーのロックダウン設定
- ColdFusion実行ユーザーの設定
- 実行ユーザーに合わせたファイル/フォルダへのアクセス権限の設定
- シャットダウンポートの変更（オプション）

CFML言語強化

ColdFusion独自の記述・ルールを改良

- 他の言語を経験している人にも馴染みやすく、違和感なく触っていただけるように
 - nullへの対応
 - データ型の保持
 - クエリ・配列の操作の効率化
 - CFC強化
 - メンバ関数の強化
- スクリプト形式・タグ形式で同じ機能を提供
 - 従来は片方の形式のみ提供されている機能があった

言語強化：Nullサポート

null 値サポートの有効化

ColdFusion で null サポートを有効にするかどうかを指定します。
有効にした場合、null 値は空の文字列に変換されません。

または Application.cfc の

```
this.enableNULLSupport=true;
```

Nullを有効にすると …

- 変数名 = null が指定可能に
- nullの値の変数に対するIsDefined()の戻り値がtrueに
 - dumpの一覧に変数が表示されるように
- 配列内の未定義の要素がNULLに
 - 以前のバージョンでは[Undefined Array Element]

データ型の保持

- 自動で型を変換せず、設定時のまま維持する

```
q1 = queryNew("id,ctype","",  
[  
  {id=1, ctype=true},  
  {id=2, ctype=false},  
  {id=3, ctype="true"},  
  {id="4", ctype="false"}  
]);  
writeOutput (SerializeJSON(q1,"struct"));
```

CF2018

```
[{"ID":1,"CTYPE":true},  
 {"ID":2,"CTYPE":false},  
 {"ID":3,"CTYPE":"true"},  
 {"ID":"4","CTYPE":"false"}]
```

CF2016

```
[{"ID":"1","CTYPE":"true"},  
 {"ID":"2","CTYPE":"false"},  
 {"ID":"3","CTYPE":"true"},  
 {"ID":"4","CTYPE":"false"}]
```

- 元の形式に戻す

- -Dcoldfusion.literal.preservetype=false

クエリ・配列・構造体の機能強化

- QueryDeleteColumn (特定のカラムを削除)
 - QueryDeleteColumn(クエリオブジェクト,"カラム名");
- QueryDeleteRow (特定のレコードを削除)
 - QueryDeleteRow(クエリオブジェクト,行番号);
- ArrayNew["String"](1) (配列の値を文字列に限定)
- (配列の切り取り)
a=[1,2,3,4,5,6,7,8]
 - a[1:6] → 1,2,3,4,5,6 a[1:6:2] → 1,3,5 a[:3] → 1,2,3 a[::-6] → 1,2,3
- ArrayFirst, ArrayLast, StructIsOrdered の追加と一部関数の機能追加

オブジェクト指向プログラミング

- OOPのデザインパターンをもとに、他のOOP言語と同レベルにCFMLを強化
 - 抽象コンポーネント・メソッド
 - finalコンポーネント・メソッド・変数
 - インタフェースのデフォルト関数
 - 戻り値や引数のタイプの共変 (covariance)

非同期プログラミング

Futureを介した非同期プログラミングで エンドユーザーの応答時間を改善する

- CThreadよりも簡単に
- Futureサポート
- Thenのチェーンとerrorでワークフローをモデル化
- 実行の結果をresultに渡せる
- ほぼリアルタイム処理
- 独自の構成可能なスレッドプールを使用

Future API



- `get()`
- `get(timeout)`
- `isDone()`
- `cancel()`
- `isCancelled()`
- `then(UDFMethod)`
- `then(UDFMethod, timeout)`
- `error(UDFMethod)`
- `error(UDFMethod, timeout)`

Empty Future

- コードを実行しない空のFuture
 - Producer Consumer デザインパターンで活用
 - ProducerからConsumerにEmpty Futureが渡され、Consumerは処理が終わったら結果とともに complete して Producerへ

```
<cfscript>
```

```
p = runAsync(); // empty future
```

```
p.complete(10);
```

```
writelog(p.get()); // displays 10
```

```
</cfscript>
```

API

- isDone()
- cancel()
- isCancelled()
- get()
- complete(Object)

その他 言語強化

■ 関数

- 名前付き引数を指定可能（呼出順を変更できる）
- 関数の呼び出しのチェーン化

■ メンバ関数の強化

- 文字列にメンバ関数が可能

■ <cfscript>内の行末の ; を省略可

■ CFC, Javaなどのインスタンス化の記述方法

- `obj1 = new java("java.lang.String")`
- `m = new component("employee")`

Rest Playground

従来のRest機能の課題

- Rest URLの探索の難しさ

- CFCを誘導して誘導する

- `http://localhost/rest/servicemapping/shopping/itemcode`

- Rest呼び出しのキャッシュ

- コードを変更するたびにColdFusion Administratorでサーバーをリフレッシュまたは再起動が必要

- 呼び出しのデバッグ

- 想定された値が取れているかなどを、実際にプログラムで呼び出さないと分かりづらい

REST Playground

REST Playground

アプリケーション

検索 +

Cfartgallery ↻ 🗑️

/artService/{id}

GET

/artserialize

GET

GET

リクエスト

GET ▼ 📄

ヘッダー パラメーター 本文 説明

id	<input type="text" value="1"/>	! 🗑️
パラメーターを追加	値を追加	

送信

レスポンス

本文 ヘッダー リクエストの要約

```
1 {"image":"aiden01.jpg","description":"Pastels/Charcoal","id":1,"name":"charles1"}
```


その他の注目ポイント

パフォーマンスの改善

- ColdFusion 2016に引き続き、2018でも既存の処理に対するパフォーマンスの改良を実施
 - CF2016に比べて30%の速度向上
 - CF11に比べて45%の速度向上
- XML/JSON関数71%↑ List関数70% ↑ 日付時間関数60%↑ 他
 - 内部処理の改良のためプログラムの変更は発生しないが、潜在的な問題が潜んでいないか等、テスト環境等にて動作確認を推奨

https://www.images2.adobe.com/content/dam/acom/en/products/coldfusion/pdfs/CF2018_Performance.pdf

キャッシュエンジンのサポート追加

- 従来の Redis キャッシュから、企業レベルのアプリケーションでの実績が高い3つのエンジン（JCS、Memcached、Redis）を利用可。

The screenshot displays a configuration interface for cache engines. On the left, a dropdown menu titled "サーバー全体のキャッシュエンジン" (Cache engine for all servers) lists "EHCache", "EHCache", "JCS", "Redis", and "Memcached".

The main area is divided into three sections:

- Memcached キャッシュ設定** (Memcached Cache Settings): Includes a text field for "Memcached サーバー" (Memcached server) with the instruction "「localhost:11211,localhost:11212」のように、memcached サーバーの IP アドレスをカンマ区切りで指定します" (Specify memcached server IP addresses separated by commas, like 'localhost:11211,localhost:11212'). A "Verify Connection" button is present.
- Redis キャッシュ設定** (Redis Cache Settings): Includes a text field for "Redis サーバー" (Redis server) and a dropdown for "Redis サーバーのポート" (Redis server port). The instruction reads: "キャッシュ用に Redis サーバー設定を指定します。キャッシュしている場合、またはアプリケーションレベルのキャッシュエンジンがある場合、またはアプリケーションレベルのキャッシュエンジンがある場合、またはアプリケーションレベルのキャッシュエンジンがある場合" (Specify Redis server settings for cache. If there is a cache, or if there is a cache engine at the application level, or if there is a cache engine at the application level, or if there is a cache engine at the application level).
- JCS キャッシュ設定** (JCS Cache Settings): Includes a dropdown for "JCS JDBC 補助キャッシュによって使用されるテーブルを作成するデータソースを選択します" (Select data source for tables created by JCS JDBC auxiliary cache). A "Select data source" button is present. There are also two checkboxes: "テーブルが既に存在する場合はオーバーライド" (Override if table already exists) and "クラスタ" (Cluster).

ColdFusion Administrator

■ 画面レイアウトの変更

The screenshot displays the ColdFusion Administrator interface for the 2018 Release Developer edition. The page title is "サーバーの設定" (Server Settings). The navigation menu includes "設定" (Settings), "リクエストの調整" (Request Tuning), "キャッシュ機能" (Cache), "クライアント変数" (Client Variables), "メモリ変数" (Memory Variables), "マッピング" (Mapping), "メール" (Email), "スケジュールされたタスク" (Scheduled Tasks), "WebSocket", "チャート" (Charts), "フォント管理" (Font Management), "ドキュメント" (Documents), and "Java と JVM". The "設定" (Settings) menu is expanded, showing "設定要約" (Settings Summary) and "サーバーの設定" (Server Settings). The "サーバーの設定" page is active, showing a "変更の送信" (Save Changes) button. The settings are organized into a grid of six items:

- リクエストタイムアウト秒**

チェックボックスをオンにすると、指定の時間を越えたリクエストは終了されます。これにより、非常に長いリクエストがサーバーリソースを占有したり、他のリクエストのパフォーマンスに悪影響を与えないようにすることができます。
- アプリケーションごとの設定の有効化**

オンにすると、アプリケーションごとの設定がサーバー全体で有効になります。オフにすると、アプリケーションごとの設定がサーバー全体で無効になります。
- cftoken 用の UUID の使用**

クライアントとセッション変数 cftoken の値用に、乱数ではなく UUID を使用するよう設定します。UUID では、トークン用に固有の識別子が生成されます。
- HTTP ステータスコードの有効化**

ColdFusion エラーがブラウザに返された場合に、ColdFusion は HTTP エラーステータスコードを設定します。テンプレートが見つからない場合はエラーステータスコード 404 が設定され、サーバーエラーの場合はエラーステータスコード 500 が設定されます。
- 空白抑制の有効化**

ColdFusion が CFML ソースファイルによって出力するかもしれない多くの余分な空白、タブ、改行を取り除き、ColdFusion からブラウザに返されるページのサイズを縮小します。
- CFC 型チェックの無効化**

オンにすると、CFC 型の UDF 引数が検証されません。引数は "ANY" という型として処理されます。この設定は本番環境でのみ使用します。

ドライバのバージョンアップ

- EXT JS 4.1 ⇒ 4.2
- Hibernate(ColdFusion ORM) ⇒ 5.2
- POI 3.17
- Tomcat 9.0.5
- 同梱の JDBC Driver 5.3 ⇒ 5.4
- Java 8 ⇒ 10

廃止・非推奨の機能

<https://helpx.adobe.com/coldfusion/deprecated-features.html>

※日本語のページは更新されていないため、英語版推奨

- 項目に Retired since が追加
 - Frex Remoting, BlazeDS, Porlets, XMLForms, Flash Forms, cfapplet, LCDS Integration, CORBA Integration, AIR Integration, flashpaper
 - インストール時には含まれず、手動でファイルの変更やコピーなどの操作が必要（非推奨 非サポート）。
- CF2018で廃止された機能：
 - Server Monitor, Server Monitor API
- 非推奨（非サポート）となった機能：
 - COM/DCOM, cfscheduleのRequestTimeOut,一部Scriptの記述

サポート期間

<https://helpx.adobe.com/support/programs/eol-matrix.html>

- ColdFusion 2016より6年のサポート期間
 - コアサポート5年
 - 延長サポート1年

ColdFusion	2016		2/16/2016	2/17/2021	2/17/2022
ColdFusion	2018		7/12/2018	7/13/2023	7/13/2014

その他

- ColdFusion 2018 は 64ビット版のみリリース
- Docker image は既に2018で公開済み
 - <https://bintray.com/eaps/coldfusion>