設定したゴール
* 〜2018年1月31日: パフォーマンスが維持されるべきRubyコアのメソッドの一覧を作成し、ベンチマーク化
* 〜2018年2月11日: Railsとは異なるRubyの利用事例(Fluentdを検討中)におけるベンチマークを用意します
* 〜2018年3月21日: 追加したベンチマークセットがbenchmark_driver.gem で継続的に実行されるようにします

開発スケジュールに対しての進捗状況
各ゴールに対しての進捗状況を説明いたします。

中間報告の時点での課題
中間報告の際、「使用方法によっては実行に想定より長く時間がかかる問題が報告されているため、1月中にそちらの改善についても取り組もうと思っています。」と述べました。
この問題は、計測対象を何回実行すべきかを推定するための処理(warming up)の際に何度もRubyのプロセスを起動していたことや、その回数の推定の精度が悪かったことによるのですが、v0.4.0でwarming upの実装を改良し1回のRubyのプロセスの起動で処理が完了するようにし、同時に回数の推定の精度も向上したため、長く時間がかかっていたケースでの計測時間が大幅に短縮されました。
例えば、string_templateというgemでの使用例では、2分19秒かかっていたのが約12秒で完了するようになっています。
https://github.com/amatsuda/string_template/pull/3

また、複数のRubyのバージョン指定した時の比較時、全てのベンチマークの全てのバージョンでの実行結果が混ざった状態で表示されており見づらいという課題がありました。現在のbenchmark_driverでは、単一のRubyのバージョンが指定された時はベンチマーク間の比較を行ない、複数のRubyのバージョンが指定された時は各ベンチマークでの異なるRubyのバージョン間での比較を行なうようになっています。
おかげさまでこの出力は、このように私以外の方にも活用いただいてい

ます。
https://speakerdeck.com/284km/fast-code-for-ruby?slide=38

パフォーマンスが維持されるべきRubyコアのメソッドの一覧を作成し、ベンチマーク化
当初は私がメソッドの一覧を考え用意する予定でしが、 @Watson1978 さんが既にそのようなベンチマークセットの作成を行なっていたため、それを benchmark_driver.gem
から利用可能にすることで目的を達成しています。
完成したベンチマークセットはこちらです: https://github.com/benchmark-driver/ruby-method-benchmarks

また計測結果は https://benchmark-driver.github.io の"Ruby Method"のところで閲覧可能になっています。
非常多くのケースでベンチマークが用意されています。

Railsとは異なるRubyの利用事例(Fluentd)におけるベンチマークの用意
このマイルストーンの実現のため、内部実装の大幅な変更を行ない、"runner"と呼ばれるオプションに"command_stdout"を指定することで、rubyで実行可能な既存のベンチマークスクリプトを何でもbenchmark_driverに接続可能にしました。
これにより、弊社の中川が作成していた fluent/fluentd-benchmark というFluentdのベンチマークを私がrubyコマンドから簡単に実行可能なように改良した上で、benchmark_driverでFluentdのベンチマークを実行できるようになりました。
複数のRubyバージョン間でのパフォーマンスが比較しやすく、また計測結果も柔軟に集計可能になりました。

今回の拡張で、Optcarrotのようなベンチマークも実行可能になり、複数のRuby処理系でのパフォーマンス比較が容易になり
https://gist.github.com/k0kubun/abae81ffb024885fc5a70933e3cbd37a
、また簡単に https://benchmark-driver.github.io に計測結果が出力でき

るようになりました。

追加したベンチマークセットの benchmark_driver.gem による継続的実行
既に何度か書いていますが、今回 https://benchmark-driver.github.io を作成しました。
これは、以下のプロジェクトを組み合わせて動作しています。

* https://github.com/benchmark-driver/skybench
* https://github.com/benchmark-driver/sky2-infra
* https://github.com/benchmark-driver/ruby-builder
* https://github.com/benchmark-driver/benchmark-driver.github.io

類似のプロジェクトにRubyBenchがありますが、短かい期間でRubyBenchチームと連携してbenchmark_driver自体も手探りのままRubyBenchを書き換えていくのは困難だと判断したため、試験的に簡単な仕組みを用意し、作成したベンチマークセットをbenchmark_driverで実行可能にしました。
ベンチマークサーバーとして利用予定であったサーバーがインターネットからアクセスできない状態であったため、この仕組みでは静的サイトをGitHub
Pagesにアップロードするようになっている点がRubyBenchとは異なっています。
将来的には今回の成果をRubyBenchプロジェクトに還元し、全員が同じ仕組みを利用できるようにしたいと考えています。

ベンチマークの継続実行にbenchmark_driverを利用するメリットとして、プラグインサポートによって集計に利用するサイトとの連携の実装が簡単になるだけでなく、benchmark_driverの計測方法によってもたらされる計測精度の向上や、FluentdやOptcarrotのような特殊なベンチマークも統一的に扱えるようになっている点が挙げられます。
また、2.6で導入されるJITコンパイラが有効な状態での計測にも簡単に対応でき、現在実行中のベンチマーク結果の集計がおいつけばJITコン

パイラのパフォーマンスの日々の変化が監視できるようになる見込みです。