

## 第 35 回競技部門：TBD

### タイトル：データ巻き (仮)

### 学校名：神戸市立工業高等専門学校

#### 1. はじめに

今回の型抜きパズルは、抜き型の種類や使用する位置など、非常に分岐が多いものとなっている。そこで、私たちは後述する基本の動きを一般化し、そこから考えられる最適解を見つけるような戦略を取ることにした。

#### 2. アプローチ

今回作成した基本の動きでは、主に行と列を別々に揃えていくような戦略となっている。大まかな流れを以下に示す。

1. 最終盤面の各行の、0-3 の各要素の

#### 3. システム

全体を JavaScript で構築し、ソルバーと UI でそれぞれの実装を分割している。

##### 3.1 ソルバー

Promise を利用した非同期処理をベースに、アルゴリズム部を WebWorker に分割してメインスレッドから分断している。これによって、アルゴリズムを別の CPU コアで動作させることができ、CPU バウンドな処理でロギングをブロックしないようにしたり、並列化したりできるようになる。

具体的な処理形態は、サーバーとの通信、UI との通信、アルゴリズムの実行をそれぞれマイクロサービスとして分割している。プログラム全体をステートマシンとして構築して、マイクロサービス全体で状態を共有できる仕組みにしている。補助的なものとして、一方向に送受信可能なデータチャネルを用いて相互に通信を行うようにしている。

実行時は、ランタイムが解釈しやすいバイトコードに AOT コンパイルした上でランタイムそのものも同梱したバイナリを事前生成し、実行することで移植性とパフォーマンスの向上に努めている。

##### 3.2 UI

React を利用し、HTTP でサーバーからの SSE (Server Sent Event) をリッスンして、ソルバーの進捗をリアルタイムで表示するようにしている。ソルバーと完全に分離することで、時間のかかる処理が行われている間も UI はブロックされない。