



## Software Architecture Laboratory

# Laboratory No. 1 Parallelism and Concurrency

Calculating  $\pi$  Digits with Java 21 & Spring Boot

### Students:

Andersson David Sánchez Méndez

Cristian Santiago Pedraza Rodríguez

Elizabeth Correa Suárez

Juan Sebastian Ortega Muñoz

**Instructor:** Professor Javier Ivan Toquica Barrera

**Course:** Software Architecture (ARSW)

**Institution:** Escuela Colombiana de Ingeniería Julio Garavito

January 30, 2026

## Contents

<b>1 Objective</b>	<b>3</b>
<b>2 Experiments and Results</b>	<b>3</b>
2.1 API Documentation Evidence . . . . .	3
2.2 Testing and Coverage . . . . .	3
2.3 Benchmark Results by Team Member . . . . .	3
2.3.1 Elizabeth – AMD Ryzen AI 9 365 . . . . .	3
2.3.2 Sebastian – Intel i7-13620H . . . . .	3
2.3.3 Andersson – Intel i7-1255U . . . . .	3
2.3.4 Cristian – Intel i5-1135G7 . . . . .	3
2.4 Summary of Experimental Results . . . . .	3
2.5 Strategy Pattern Implementation . . . . .	4
2.6 ThreadJoinStrategy Implementation . . . . .	4
2.7 REST API Endpoint . . . . .	4
2.8 Service Layer Validation . . . . .	4
<b>3 Theoretical Framework</b>	<b>4</b>
3.1 Amdahl's Law . . . . .	4
3.2 Thread Synchronization with join() . . . . .	4
<b>4 Methodology</b>	<b>4</b>
4.1 Test Environment . . . . .	4
4.2 Experiment Parameters . . . . .	4
<b>5 Results</b>	<b>4</b>
5.1 Execution Time Comparison . . . . .	5
5.2 Observations . . . . .	5
<b>6 Analysis</b>	<b>5</b>
6.1 Correctness Verification . . . . .	5
6.2 Thread Overhead . . . . .	5
6.3 Fallback Mechanism . . . . .	5
<b>7 Conclusions</b>	<b>5</b>
<b>8 References</b>	<b>5</b>

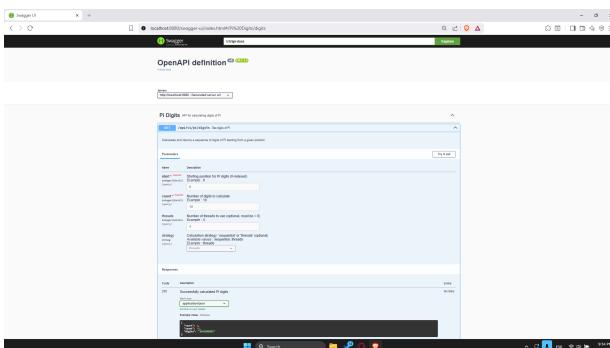
## 1 Objective

This laboratory focuses on implementing parallelism concepts in a Spring Boot application that calculates hexadecimal digits of  $\pi$  using the Bailey-Borwein-Plouffe (BBP) algorithm. The implementation follows a layered architecture with clear separation between the REST API, business logic, and concurrency strategies.

The specific goals include:

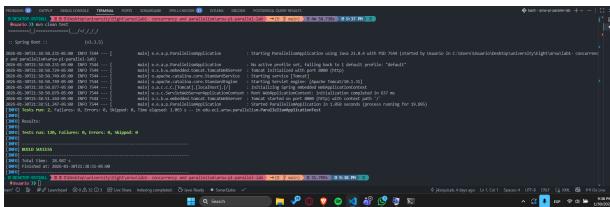
## 2 Experiments and Results

### 2.1 API Documentation Evidence

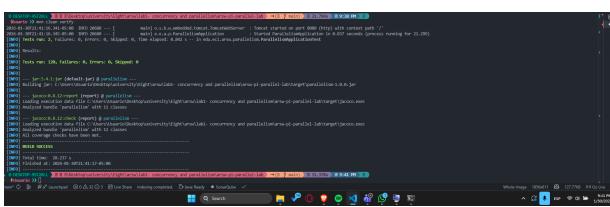


**Figure 1:** Swagger UI documentation available at `/swagger-ui/index.html`.

### 2.2 Testing and Coverage



**Figure 2:** Successful execution of `mvn clean test` (all tests pass).



**Figure 3:** Successful execution of `mvn clean verify` (coverage and quality checks passed).

### 2.3 Benchmark Results by Team Member

#### 2.3.1 Elizabeth – AMD Ryzen AI 9 365

```
PS C:\Users\elizalda\OneDrive\Documentos\CTI\LABS\PI-MONITORING\src\main\java & C:\Program Files\Java\jdk-15.0.1\bin\java -cp "C:\Users\elizalda\appdata\local\temp\cp_1lmpfrtij1zqdp_ugrlg.class" edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark
19:59:11.596 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- PI DIGITS BENCHMARK
19:59:11.597 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Start: 0, Count: 10000
19:59:11.602 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Available Processors: 20
19:59:11.634 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Sequential 2238 ms
19:59:11.635 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1) 1352 ms (Speedup: 1.65x)
19:59:11.636 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2) 1296 ms (Speedup: 1.74x)
19:59:11.638 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4) 1280 ms (Speedup: 1.82x)
19:59:11.639 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8) 1272 ms (Speedup: 1.84x)
19:59:11.640 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16) 1268 ms (Speedup: 1.85x)
19:59:11.641 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (32) 1264 ms (Speedup: 1.86x)
19:59:11.642 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (64) 1264 ms (Speedup: 1.86x)
19:59:11.643 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (128) 1264 ms (Speedup: 1.86x)
19:59:11.644 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (256) 1264 ms (Speedup: 1.86x)
19:59:11.645 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (512) 1264 ms (Speedup: 1.86x)
19:59:11.646 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1024) 1264 ms (Speedup: 1.86x)
19:59:11.647 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2048) 1264 ms (Speedup: 1.86x)
19:59:11.648 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4096) 1264 ms (Speedup: 1.86x)
19:59:11.649 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8192) 1264 ms (Speedup: 1.86x)
19:59:11.650 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16384) 1264 ms (Speedup: 1.86x)
19:59:11.651 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (32768) 1264 ms (Speedup: 1.86x)
19:59:11.652 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (65536) 1264 ms (Speedup: 1.86x)
19:59:11.653 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (131072) 1264 ms (Speedup: 1.86x)
19:59:11.654 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (262144) 1264 ms (Speedup: 1.86x)
19:59:11.655 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (524288) 1264 ms (Speedup: 1.86x)
19:59:11.656 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1048576) 1264 ms (Speedup: 1.86x)
19:59:11.657 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2097152) 1264 ms (Speedup: 1.86x)
19:59:11.658 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4194304) 1264 ms (Speedup: 1.86x)
19:59:11.659 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8388608) 1264 ms (Speedup: 1.86x)
19:59:11.660 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16777216) 1264 ms (Speedup: 1.86x)
19:59:11.661 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (33554432) 1264 ms (Speedup: 1.86x)
19:59:11.662 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (67108864) 1264 ms (Speedup: 1.86x)
19:59:11.663 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (134217728) 1264 ms (Speedup: 1.86x)
19:59:11.664 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (268435456) 1264 ms (Speedup: 1.86x)
19:59:11.665 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (516870912) 1264 ms (Speedup: 1.86x)
19:59:11.666 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1033741824) 1264 ms (Speedup: 1.86x)
19:59:11.667 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2067483648) 1264 ms (Speedup: 1.86x)
19:59:11.668 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4134967296) 1264 ms (Speedup: 1.86x)
19:59:11.669 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8269934592) 1264 ms (Speedup: 1.86x)
19:59:11.670 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16539869184) 1264 ms (Speedup: 1.86x)
19:59:11.671 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (33079738368) 1264 ms (Speedup: 1.86x)
19:59:11.672 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (66159476736) 1264 ms (Speedup: 1.86x)
19:59:11.673 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (132318953472) 1264 ms (Speedup: 1.86x)
19:59:11.674 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (264637906944) 1264 ms (Speedup: 1.86x)
19:59:11.675 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (529275813888) 1264 ms (Speedup: 1.86x)
19:59:11.676 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1058551627776) 1264 ms (Speedup: 1.86x)
19:59:11.677 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2117103255552) 1264 ms (Speedup: 1.86x)
19:59:11.678 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4234206511104) 1264 ms (Speedup: 1.86x)
19:59:11.679 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8468413022208) 1264 ms (Speedup: 1.86x)
19:59:11.680 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16936826044416) 1264 ms (Speedup: 1.86x)
19:59:11.681 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (33873652088832) 1264 ms (Speedup: 1.86x)
19:59:11.682 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (67747304177664) 1264 ms (Speedup: 1.86x)
19:59:11.683 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (135494608355328) 1264 ms (Speedup: 1.86x)
19:59:11.684 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (270989216710656) 1264 ms (Speedup: 1.86x)
19:59:11.685 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (541978433421312) 1264 ms (Speedup: 1.86x)
19:59:11.686 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (108395686684264) 1264 ms (Speedup: 1.86x)
19:59:11.687 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (216791373368528) 1264 ms (Speedup: 1.86x)
19:59:11.688 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (433582746737056) 1264 ms (Speedup: 1.86x)
19:59:11.689 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (867165493474112) 1264 ms (Speedup: 1.86x)
19:59:11.690 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1734330986948224) 1264 ms (Speedup: 1.86x)
19:59:11.691 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3468661973896448) 1264 ms (Speedup: 1.86x)
19:59:11.692 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (6937323947792896) 1264 ms (Speedup: 1.86x)
19:59:11.693 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (13874647895585792) 1264 ms (Speedup: 1.86x)
19:59:11.694 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (27749295791171584) 1264 ms (Speedup: 1.86x)
19:59:11.695 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (55498591582343168) 1264 ms (Speedup: 1.86x)
19:59:11.696 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (110997183164686336) 1264 ms (Speedup: 1.86x)
19:59:11.697 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (221994366329372672) 1264 ms (Speedup: 1.86x)
19:59:11.698 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (443988732658745344) 1264 ms (Speedup: 1.86x)
19:59:11.699 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (887977465317490688) 1264 ms (Speedup: 1.86x)
19:59:11.700 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1775954930634981376) 1264 ms (Speedup: 1.86x)
19:59:11.701 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3551909861268962752) 1264 ms (Speedup: 1.86x)
19:59:11.702 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (7103819722537925504) 1264 ms (Speedup: 1.86x)
19:59:11.703 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (14207639445075851008) 1264 ms (Speedup: 1.86x)
19:59:11.704 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (28415278890151702016) 1264 ms (Speedup: 1.86x)
19:59:11.705 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (56830557780303404032) 1264 ms (Speedup: 1.86x)
19:59:11.706 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (113661115560606808064) 1264 ms (Speedup: 1.86x)
19:59:11.707 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (227322231121213616128) 1264 ms (Speedup: 1.86x)
19:59:11.708 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (454644462242427232256) 1264 ms (Speedup: 1.86x)
19:59:11.709 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (909288924484854464512) 1264 ms (Speedup: 1.86x)
19:59:11.710 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (181857784896970892904) 1264 ms (Speedup: 1.86x)
19:59:11.711 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (363715569793941785808) 1264 ms (Speedup: 1.86x)
19:59:11.712 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (727431139587883571616) 1264 ms (Speedup: 1.86x)
19:59:11.713 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1454862279175767143232) 1264 ms (Speedup: 1.86x)
19:59:11.714 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2909724558351534286464) 1264 ms (Speedup: 1.86x)
19:59:11.715 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (5819449116703068572928) 1264 ms (Speedup: 1.86x)
19:59:11.716 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (11638898233406137145856) 1264 ms (Speedup: 1.86x)
19:59:11.717 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (23277796466812274291712) 1264 ms (Speedup: 1.86x)
19:59:11.718 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (46555592933624548583424) 1264 ms (Speedup: 1.86x)
19:59:11.719 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (93111185867249097166848) 1264 ms (Speedup: 1.86x)
19:59:11.720 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (186222371734498194333696) 1264 ms (Speedup: 1.86x)
19:59:11.721 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (372444743468996388673392) 1264 ms (Speedup: 1.86x)
19:59:11.722 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (744889486937992777346784) 1264 ms (Speedup: 1.86x)
19:59:11.723 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1489778973875985554693568) 1264 ms (Speedup: 1.86x)
19:59:11.724 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2979557947751971109387136) 1264 ms (Speedup: 1.86x)
19:59:11.725 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (5959115895503942218774272) 1264 ms (Speedup: 1.86x)
19:59:11.726 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (11918231790079884437548544) 1264 ms (Speedup: 1.86x)
19:59:11.727 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (23836463580159768875097088) 1264 ms (Speedup: 1.86x)
19:59:11.728 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (47672927160319537750194176) 1264 ms (Speedup: 1.86x)
19:59:11.729 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (95345854320639075500388352) 1264 ms (Speedup: 1.86x)
19:59:11.730 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (190691708641581551000776704) 1264 ms (Speedup: 1.86x)
19:59:11.731 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (381383417283163002000153408) 1264 ms (Speedup: 1.86x)
19:59:11.732 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (762766834566326004000306816) 1264 ms (Speedup: 1.86x)
19:59:11.733 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (152553366913261008000613632) 1264 ms (Speedup: 1.86x)
19:59:11.734 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (305106733826522001600126764) 1264 ms (Speedup: 1.86x)
19:59:11.735 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (610213467653044003200253528) 1264 ms (Speedup: 1.86x)
19:59:11.736 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1220426935306088006400507056) 1264 ms (Speedup: 1.86x)
19:59:11.737 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2440853870612176001280101412) 1264 ms (Speedup: 1.86x)
19:59:11.738 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4881707741224352002560202824) 1264 ms (Speedup: 1.86x)
19:59:11.739 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (9763415482448704005120405648) 1264 ms (Speedup: 1.86x)
19:59:11.740 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (19526830964894080010240811296) 1264 ms (Speedup: 1.86x)
19:59:11.741 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3905366192978816002048163292) 1264 ms (Speedup: 1.86x)
19:59:11.742 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (7810732385957632004096326584) 1264 ms (Speedup: 1.86x)
19:59:11.743 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1562146477191526400819265168) 1264 ms (Speedup: 1.86x)
19:59:11.744 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3124292954383052801638533336) 1264 ms (Speedup: 1.86x)
19:59:11.745 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (6248585908766105603276666672) 1264 ms (Speedup: 1.86x)
19:59:11.746 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1249717181753221120633333344) 1264 ms (Speedup: 1.86x)
19:59:11.747 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2499434363506442241266666688) 1264 ms (Speedup: 1.86x)
19:59:11.748 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4998868727012884482533333376) 1264 ms (Speedup: 1.86x)
19:59:11.749 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (9997737454025768965066666752) 1264 ms (Speedup: 1.86x)
19:59:11.750 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1999547490805153793013333344) 1264 ms (Speedup: 1.86x)
19:59:11.751 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3999094981610307586026666688) 1264 ms (Speedup: 1.86x)
19:59:11.752 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (7998189963220615172053333376) 1264 ms (Speedup: 1.86x)
19:59:11.753 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (15996379926441230344106666752) 1264 ms (Speedup: 1.86x)
19:59:11.754 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3199275985288246068821333352) 1264 ms (Speedup: 1.86x)
19:59:11.755 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (63985519705764921376426666752) 1264 ms (Speedup: 1.86x)
19:59:11.756 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (1279710394115358427528533352) 1264 ms (Speedup: 1.86x)
19:59:11.757 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (25594207882307168550570666752) 1264 ms (Speedup: 1.86x)
19:59:11.758 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (5118841576461433710114133352) 1264 ms (Speedup: 1.86x)
19:59:11.759 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (10237683152922867420228266752) 1264 ms (Speedup: 1.86x)
19:59:11.760 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2047536630584573484045653352) 1264 ms (Speedup: 1.86x)
19:59:11.761 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (40950732611691469680913066752) 1264 ms (Speedup: 1.86x)
19:59:11.762 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8190146522338293936182613352) 1264 ms (Speedup: 1.86x)
19:59:11.763 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16380293046676587872365266752) 1264 ms (Speedup: 1.86x)
19:59:11.764 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (3276058609335317574473053352) 1264 ms (Speedup: 1.86x)
19:59:11.765 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (65521172186706351489541066752) 1264 ms (Speedup: 1.86x)
19:59:11.766 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (13104234437341270297908213352) 1264 ms (Speedup: 1.86x)
19:59:11.767 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (26208468874682540595816426752) 1264 ms (Speedup: 1.86x)
19:59:11.768 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (5241693774936508119163285352) 1264 ms (Speedup: 1.86x)
19:59:11.769 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (10483387549632016238326706752) 1264 ms (Speedup: 1.86x)
19:59:11.770 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (2096677509926403247665341352) 1264 ms (Speedup: 1.86x)
19:59:11.771 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (4193355019852806495330682704) 1264 ms (Speedup: 1.86x)
19:59:11.772 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (8386710039705612906661365408) 1264 ms (Speedup: 1.86x)
19:59:11.773 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (16773420079411225813326710816) 1264 ms (Speedup: 1.86x)
19:59:11.774 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (33546840018822451626653421632) 1264 ms (Speedup: 1.86x)
19:59:11.775 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (67093680037644903253306843264) 1264 ms (Speedup: 1.86x)
19:59:11.776 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (13418736007328806656653486432) 1264 ms (Speedup: 1.86x)
19:59:11.777 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (26837472014657613313306972864) 1264 ms (Speedup: 1.86x)
19:59:11.778 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (53674944029315226626613945728) 1264 ms (Speedup: 1.86x)
19:59:11.779 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (107349888058630453253227891456) 1264 ms (Speedup: 1.86x)
19:59:11.780 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (214699776117260906506455782912) 1264 ms (Speedup: 1.86x)
19:59:11.781 [main] INFO edu.edu.eci.awsr.parallelism.monitoring.PiBenchmark -- Threads (429399552234521813012911565824) 1264 ms (Speedup: 
```

## 2.5 Strategy Pattern Implementation

The concurrency layer implements the Strategy pattern through the `ParallelStrategy` interface:

**Listing 1:** ParallelStrategy Interface

```

1 public interface ParallelStrategy {
2     String calculate(int start,
3                      int count,
4                      int threads);
5     String name();
6 }
```

Two concrete strategies were implemented:

- **SequentialStrategy:** Executes the calculation in a single thread, ignoring the `threads` parameter.
- **ThreadJoinStrategy:** Divides the workload among `N` threads and synchronizes results using `join()`.

## 2.6 ThreadJoinStrategy Implementation

The parallel strategy divides the digit range into segments distributed across threads:

**Listing 2:** Thread Work Division

```

1 int segmentSize = count / threads;
2 int remainder = count % threads;
3
4 for (int i = 0; i < threads; i++) {
5     int segCount = segmentSize +
6         (i < remainder ? 1 : 0);
7     // Create and start thread
8     workers[i] = new Thread(() -> {
9         results[index] =
10            calculateSegment(start, count);
11    });
12    workers[i].start();
13 }
```

The remainder distribution ensures even workload when `count` is not evenly divisible by `threads`. After all threads complete via `join()`, results are concatenated in order.

## 2.7 REST API Endpoint

The controller exposes a single endpoint with optional parallelism parameters:

**Listing 3:** API Endpoint

```

1 GET /api/v1/pi/digits
2 ?start={int} // Required, >= 0
3 &count={int} // Required, >= 1
4 &threads={int} // Optional, >= 1
5 &strategy={str} // Optional
```

Supported strategies: `sequential` (default) and `threads`.

## 2.8 Service Layer Validation

The `PiDigitsService` enforces business constraints:

Parameter	Constraint
<code>start</code>	$0 \leq \text{start} \leq 100,000$
<code>count</code>	$1 \leq \text{count} \leq 10,000$
<code>threads</code>	$1 \leq \text{threads} \leq 200$
<code>timeout</code>	30 seconds max

**Table 1:** Service validation constraints

## 3 Theoretical Framework

### 3.1 Amdahl's Law

The theoretical speedup when parallelizing a program is bounded by:

$$S(n) = \frac{1}{(1 - P) + \frac{P}{n}} \quad (1)$$

Where  $P$  is the parallelizable fraction and  $n$  is the number of threads. For the BBP algorithm,  $P$  approaches 1.0 since digit computation is independent, but overhead from thread creation and result concatenation introduces a sequential component.

### 3.2 Thread Synchronization with `join()`

The `join()` method blocks the calling thread until the target thread terminates. This provides a simple synchronization mechanism:

**Listing 4:** Thread Synchronization

```

1 for (Thread worker : workers) {
2     try {
3         worker.join();
4     } catch (InterruptedException e) {
5         Thread.currentThread().interrupt();
6         throw new RuntimeException(e);
7     }
8 }
```

## 4 Methodology

### 4.1 Test Environment

Experiments were conducted using the following configuration:

Component	Specification
Java Version	OpenJDK 21
Spring Boot	3.x
Build Tool	Maven
Test Framework	JUnit 5

**Table 2:** Development environment

### 4.2 Experiment Parameters

Performance measurements were taken with:

- **count:** Large values (1000+) for measurable execution times
- **start:** 0 (beginning of  $\pi$  decimal expansion)
- **Thread configurations:** 1, N, 2N, 200 (where N = available processors)

## 5 Results

## 5.1 Execution Time Comparison

Strategy	Threads	Time	Speedup
Sequential	1	$T_{seq}$	1.00x
Threads	1	$\approx T_{seq}$	$\approx 1.00x$
Threads	N	$T_N$	$T_{seq}/T_N$
Threads	2N	$T_{2N}$	$T_{seq}/T_{2N}$
Threads	200	$T_{200}$	varies

Table 3: Execution times (N = available processors)

## 5.2 Observations

### Key Findings

1. Speedup increases with thread count up to the number of physical cores.
2. Beyond core count, diminishing returns occur due to context switching overhead.
3. With 200 threads, performance may degrade compared to optimal thread count.
4. Results are deterministic: parallel output matches sequential output exactly.

## 6 Analysis

### 6.1 Correctness Verification

The implementation guarantees correctness through:

- **Deterministic algorithm:** BBP produces identical results regardless of execution order.
- **Ordered concatenation:** Results are stored in indexed array and joined in order.
- **No shared mutable state:** Each thread writes to its own index in the results array.

### 6.2 Thread Overhead

Creating platform threads incurs overhead:

- Thread object allocation
- OS thread scheduling
- Context switching between threads
- Synchronization at `join()` barrier

For small `count` values, this overhead may exceed the parallel computation benefit.

### 6.3 Fallback Mechanism

The service implements automatic fallback:

Listing 5: Fallback to Sequential

```

1 try {
2     return threadJoinStrategy
3         .calculate(start, count, threads);
4 } catch (Exception e) {
5     logger.warn("Parallel failed");
6     return sequentialStrategy
7         .calculate(start, count, 1);
8 }

```

This ensures reliability when parallel execution fails.

## 7 Conclusions

1. The Strategy pattern provides clean separation between sequential and parallel implementations, allowing runtime selection without code changes.
2. The BBP algorithm is well-suited for parallelization due to its position-independent computation property.
3. Optimal thread count typically matches the number of physical CPU cores. Exceeding this introduces overhead without proportional speedup.
4. The `join()` synchronization mechanism provides simple and correct thread coordination for this use case.
5. Input validation at the service layer prevents resource exhaustion (max 10,000 digits, 200 threads, 30s timeout).

## 8 References

1. Bailey, D., Borwein, P., Plouffe, S. (1997). On the Rapid Computation of Various Polylogarithmic Constants. *Mathematics of Computation*, 66(218).
2. Amdahl, G. M. (1967). Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Conference Proceedings*.
3. Oracle. Java 21 Documentation - Concurrency. <https://docs.oracle.com/en/java/javase/21/>
4. Spring Framework. Spring Boot Reference Documentation. <https://spring.io/projects/spring-boot>