Sultan Qaboos university

College of Engineering - Computer and Network Department

ECCE5232

Almaala Taal 135591

Abdulmunim Khamis 133225

Github copilot

Oct 16 2024

Introduction

In this lab, we wanted to compare the performance of several CPUs. A good measure to compare CPU performances is to look for their benchmarks. Using different benchmarks yields different results which makes it a challenge to compare the performance. From the mathematical point of view, we can get arithmetic mean to give an indication of the performance but it tends to neglect extreme values away from the mean, for example, the mean of 1 and 100 is 50.5 which is extremely misleading and does not accurately weigh into consideration both values, more specifically the “outliers” that are extreme. That's why using several means can eliminate these mathematical tendencies; this is the measure we will use in this lab. We will build a simple website based on HTML and JS to calculate the geometric and arithmetic means for each processor based on the speed metric to eventually rank all processors based on their benchmarks in a mathematically accurate manner. The website is hosted on GitHub and can be accessed through this link
https://oulla898.github.io/lab1_arch_135591_133225/cpu_performance_calculator.html

formulas

speed metric:

$$r_i = \frac{T_{ref_i}}{T_{sut_i}}$$

arithmetic mean of speed metrics:

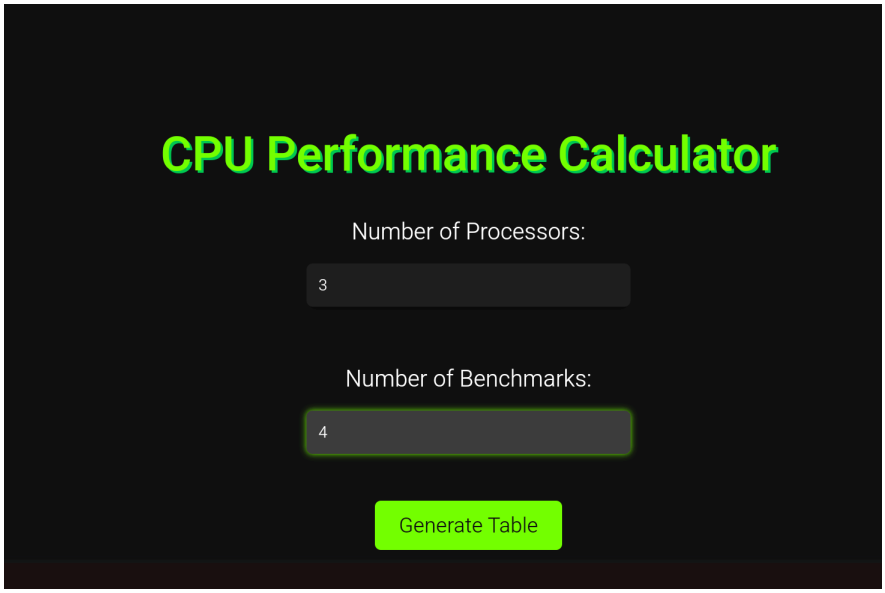
$$AM = \frac{1}{m}\sum_{i=1}^{m} r_i$$

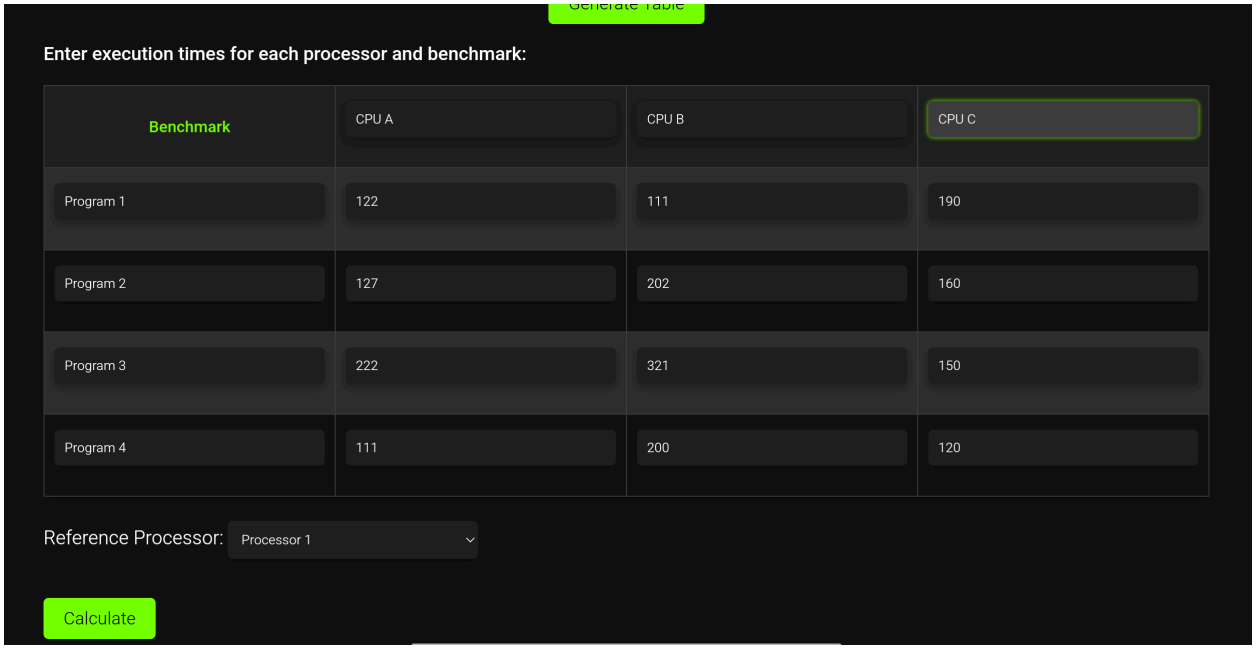
geometric mean of speed metrics:

$$GM = \left(\prod_{i=1}^{n} r_i\right)^{1/n}$$

Results

Three processors are compared using 4 benchmark programs. The website is very easy to use, users simply fill the table with execution time in seconds for every benchmark and then select a reference processor. Here is a sample output:





All calculations results and ranking are shown here:

Speed Metrics:

Benchmark	CPU A	CPU B	CPU C
Program 1	1.000	1.099	0.642
Program 2	1.000	0.629	0.794
Program 3	1.000	0.692	1.480
Program 4	1.000	0.555	0.925

Arithmetic Mean

Processor	Arithmetic Mean
CPU A	1.000
CPU B	0.744
CPU C	0.960

Geometric Mean

Processor	Geometric Mean
CPU A	1.000
CPU B	0.718
CPU C	0.914

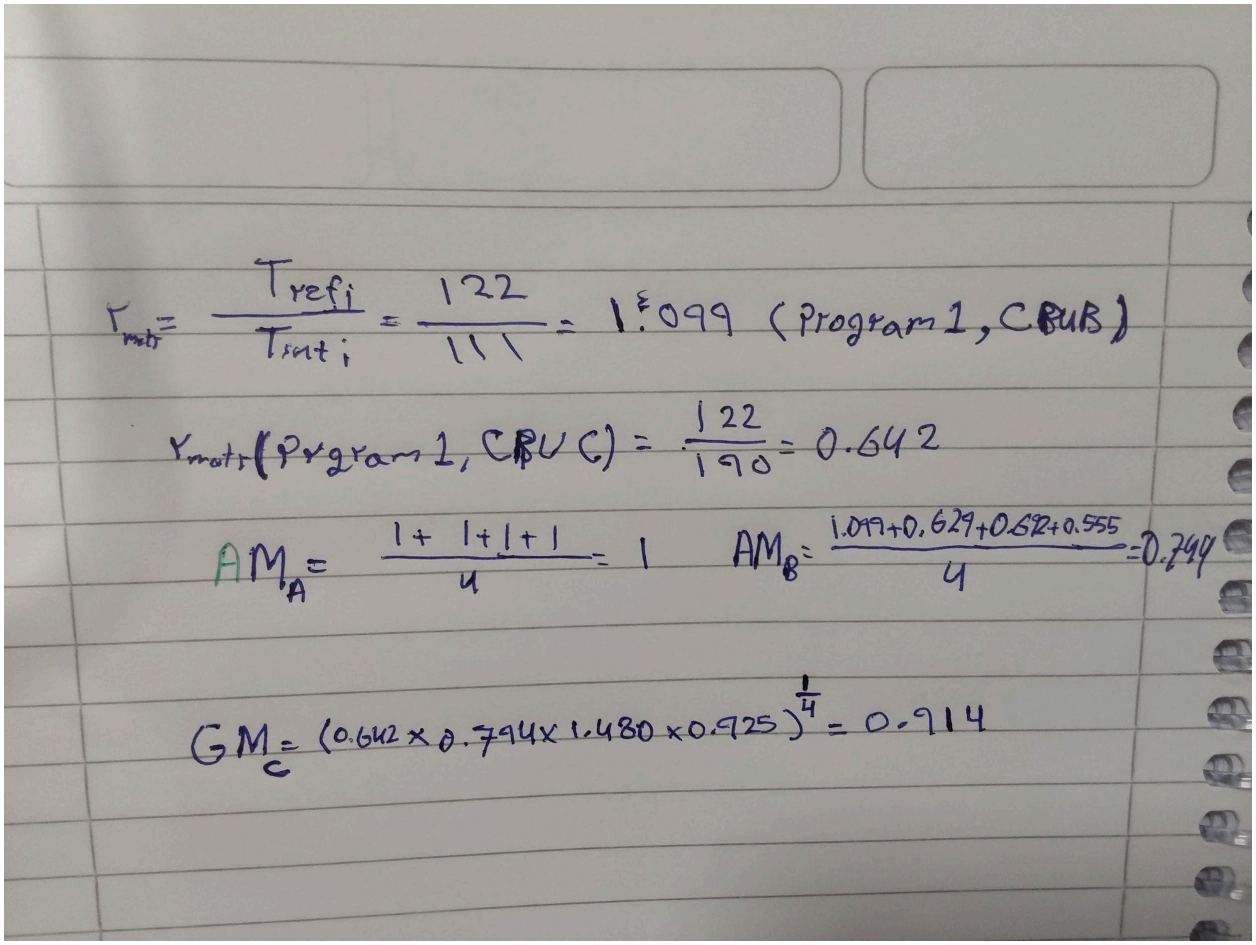
Ranking (Arithmetic Mean)

1. CPU A
2. CPU C
3. CPU B

Ranking (Geometric Mean)

1. CPU A
2. CPU C
3. CPU B

In order to check the results, some calculations are done manually and it is found that it is consistent with website calculations.



HTML/JS Code:

This is the HTML code defining the elements in the page.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>CPU Performance Calculator</title>
    <script src="cpu_performance_calculator.js"></script>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            background-color: #121212;
            color: #ffffff;
            margin: 0;
            padding: 0;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            animation: fadeIn 1s ease-in;
        }

        h1 {
            font-size: 2.5rem;
            margin-bottom: 20px;
            color: #76ff03;
            text-shadow: 2px 2px #00c853;
        }

        label {
            font-size: 1.2rem;
            margin: 10px 0;
        }

        input[type="number"], input[type="text"], select {
            padding: 10px;
            border: none;
            border-radius: 5px;
            margin: 5px 0 20px 0;
            width: 250px;
            background-color: #1e1e1e;
            color: #ffffff;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
            transition: all 0.3s ease;
        }
```

```html
        input[type="number"]:focus, input[type="text"]:focus, select:focus {
            outline: none;
            background-color: #3e3e3e;
            box-shadow: 0 0 5px #76ff03;
        }

        button {
            padding: 10px 20px;
            background-color: #76ff03;
            color: #121212;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            font-size: 1.1rem;
            transition: background-color 0.3s ease;
        }

        button:hover {
            background-color: #00c853;
            transform: scale(1.05);
        }

        table {
            border-collapse: collapse;
            width: 100%;
            margin-top: 20px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.3);
        }

        th, td {
            padding: 10px;
            text-align: center;
            border: 1px solid #3e3e3e;
        }

        th {
            background-color: #1e1e1e;
            color: #76ff03;
        }

        tr:nth-child(even) {
            background-color: #2e2e2e;
        }

        @keyframes fadeIn {
            from {
                opacity: 0;
            }
            to {
                opacity: 1;
            }
        }
    </style>
</head>
<body>
    <h1>CPU Performance Calculator</h1>
    <label for="numProcessors">Number of Processors:</label>
    <input type="number" id="numProcessors" name="numProcessors" min="1" required><br>

    <label for="numBenchmarks">Number of Benchmarks:</label>
    <input type="number" id="numBenchmarks" name="numBenchmarks" min="1" required><br>

    <button onclick="generateTable()">Generate Table</button>

    <div id="inputTable"></div>
    <div id="output"></div>
</body>
```

This is the JS code showing the math behind the calculations

```
function generateTable() {
    var numProcessors = parseInt(document.getElementById("numProcessors").value);
    var numBenchmarks = parseInt(document.getElementById("numBenchmarks").value);
    var table = "<h3>Enter execution times for each processor and benchmark:</h3><table border='1'><tr><th>Benchmark</th>";

    // Create the table header for processor names
    for (let i = 0; i < numProcessors; i++) {
        table += "<th><input type='text' id='processor" + i + "' placeholder='Processor " + (i + 1) + "'></th>";
    }
    table += "</tr>";

    // Create rows for benchmark names and execution times
    for (let j = 0; j < numBenchmarks; j++) {
        table += "<tr><td><input type='text' id='benchmark" + j + "' placeholder='Benchmark " + (j + 1) + "'></td>";
        for (let i = 0; i < numProcessors; i++) {
            table += "<td><input type='number' id='time" + i + "_" + j + "' placeholder='Time (sec)'></td>";
        }
        table += "</tr>";
    }

    table += "</table><br><label for='referenceProcessor'>Reference Processor:</label> <select id='referenceProcessor'>";
    for (let i = 0; i < numProcessors; i++) {
        table += "<option value='" + i + "'>Processor " + (i + 1) + "</option>";
    }
    table += "</select><br><br><button onclick='calculateMeans()'>Calculate</button>";

    document.getElementById("inputTable").innerHTML = table;
}

function calculateMeans() {
    var numProcessors = parseInt(document.getElementById("numProcessors").value);
    var numBenchmarks = parseInt(document.getElementById("numBenchmarks").value);
    var processors = [];
    var benchmarks = [];
    var times = [];
    var speedMetrics = [];
    var referenceProcessor = parseInt(document.getElementById("referenceProcessor").value);

    // Getting processor names
    for (let i = 0; i < numProcessors; i++) {
        let processorName = document.getElementById("processor" + i).value;
        processors.push(processorName);
    }
```

```
// Getting benchmark names
for (let j = 0; j < numBenchmarks; j++) {
    let benchmarkName = document.getElementById("benchmark" + j).value;
    benchmarks.push(benchmarkName);
}

// Getting execution times for each benchmark on each processor
for (let i = 0; i < numProcessors; i++) {
    times[i] = [];
    for (let j = 0; j < numBenchmarks; j++) {
        let time = parseFloat(document.getElementById("time" + i + "_" + j).value);
        times[i].push(time);
    }
}

// Calculating speed metrics for each benchmark
for (let j = 0; j < numBenchmarks; j++) {
    speedMetrics[j] = [];
    for (let i = 0; i < numProcessors; i++) {
        let speed = times[referenceProcessor][j] / times[i][j];
        speedMetrics[j].push(speed);
    }
}

// Calculating arithmetic and geometric means
var arithmeticMeans = [];
var geometricMeans = [];

for (let i = 0; i < numProcessors; i++) {
    let arithmeticSum = 0;
    let geometricProduct = 1;

    for (let j = 0; j < numBenchmarks; j++) {
        arithmeticSum += speedMetrics[j][i];
        geometricProduct *= speedMetrics[j][i];
    }

    arithmeticMeans.push(arithmeticSum / numBenchmarks);
    geometricMeans.push(Math.pow(geometricProduct, 1 / numBenchmarks));
}

// Displaying the results
var result = "<h3>Speed Metrics:</h3><table border='1'><tr><th>Benchmark</th>";

for (let i = 0; i < numProcessors; i++) {
    result += "<th>" + processors[i] + "</th>";
}
result += "</tr>";

for (let j = 0; j < numBenchmarks; j++) {
    result += "<tr><td>" + benchmarks[j] + "</td>";
    for (let i = 0; i < numProcessors; i++) {
        result += "<td>" + speedMetrics[j][i].toFixed(3) + "</td>";
    }
    result += "</tr>";
}

result += "</table><br><h3>Arithmetic Mean</h3><table border='1'><tr><th>Processor</th><th>Arithmetic Mean</th></tr>";

for (let i = 0; i < numProcessors; i++) {
    result += "<tr><td>" + processors[i] + "</td><td>" + arithmeticMeans[i].toFixed(3) + "</td></tr>";
}

result += "</table><br><h3>Geometric Mean</h3><table border='1'><tr><th>Processor</th><th>Geometric Mean</th></tr>";

for (let i = 0; i < numProcessors; i++) {
    result += "<tr><td>" + processors[i] + "</td><td>" + geometricMeans[i].toFixed(3) + "</td></tr>";
}

result += "</table><br><h3>Ranking (Arithmetic Mean)</h3><ol>";
var arithmeticRanking = [...processors].sort((a, b) => arithmeticMeans[processors.indexOf(b)] - arithmeticMeans[processors.indexOf(a)]);
for (let i = 0; i < numProcessors; i++) {
    result += "<li>" + arithmeticRanking[i] + "</li>";
}
result += "</ol><br><h3>Ranking (Geometric Mean)</h3><ol>";
var geometricRanking = [...processors].sort((a, b) => geometricMeans[processors.indexOf(b)] - geometricMeans[processors.indexOf(a)]);
for (let i = 0; i < numProcessors; i++) {
    result += "<li>" + geometricRanking[i] + "</li>";
}
result += "</ol>";

document.getElementById("output").innerHTML = result;
```

Conclusion

In conclusion, In this lab we created a simple website with a nice user interface in order to compare a number of processors speed by determining their arithmetic mean and geometric mean of execution time in a number of benchmarks. Program calculations and manual calculations are consistent. processor benchmarking is widely used in comparing performance of processors. There are websites where people who are interested in benchmarking can go and test their CPU, like UserBenchmark. This website also has all users benchmark data so anyone could check his processor speed and compare it with other processor models.