

باسمه تعالی

پروژه پنجم برنامه نویسی موازی

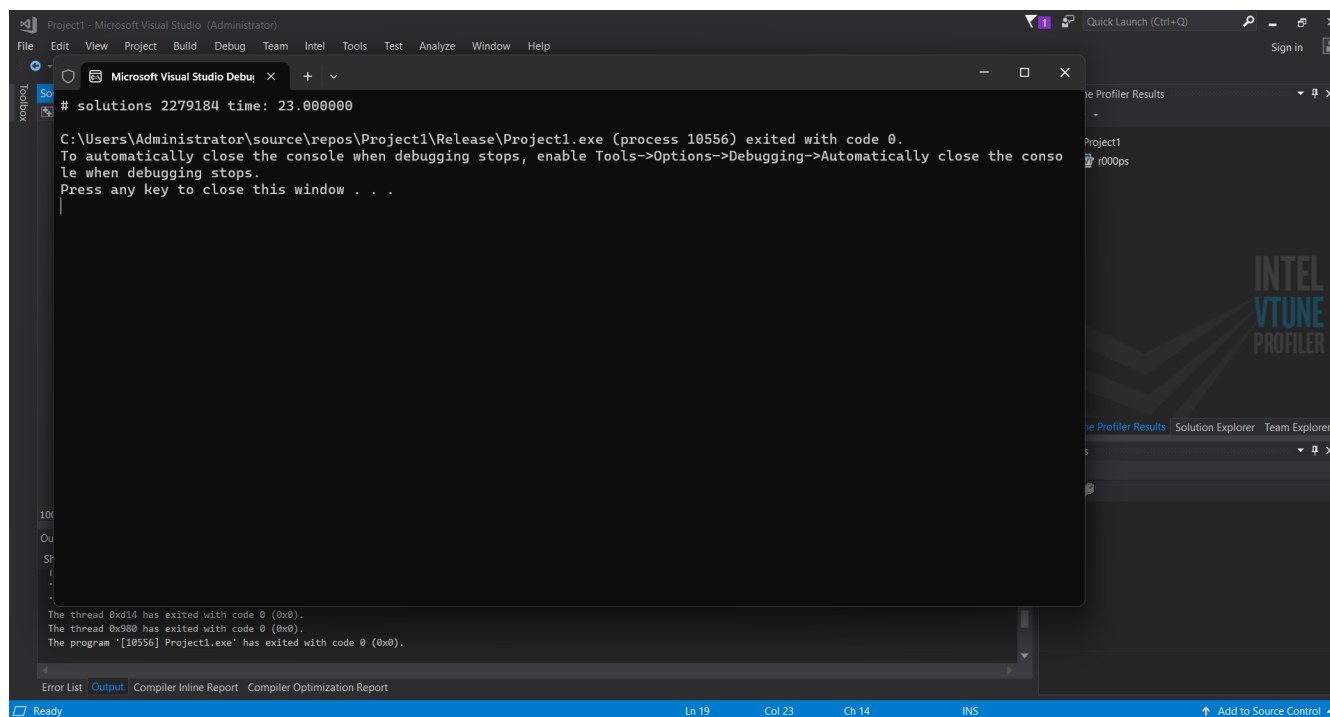
استاد صفری

دانشجویان:

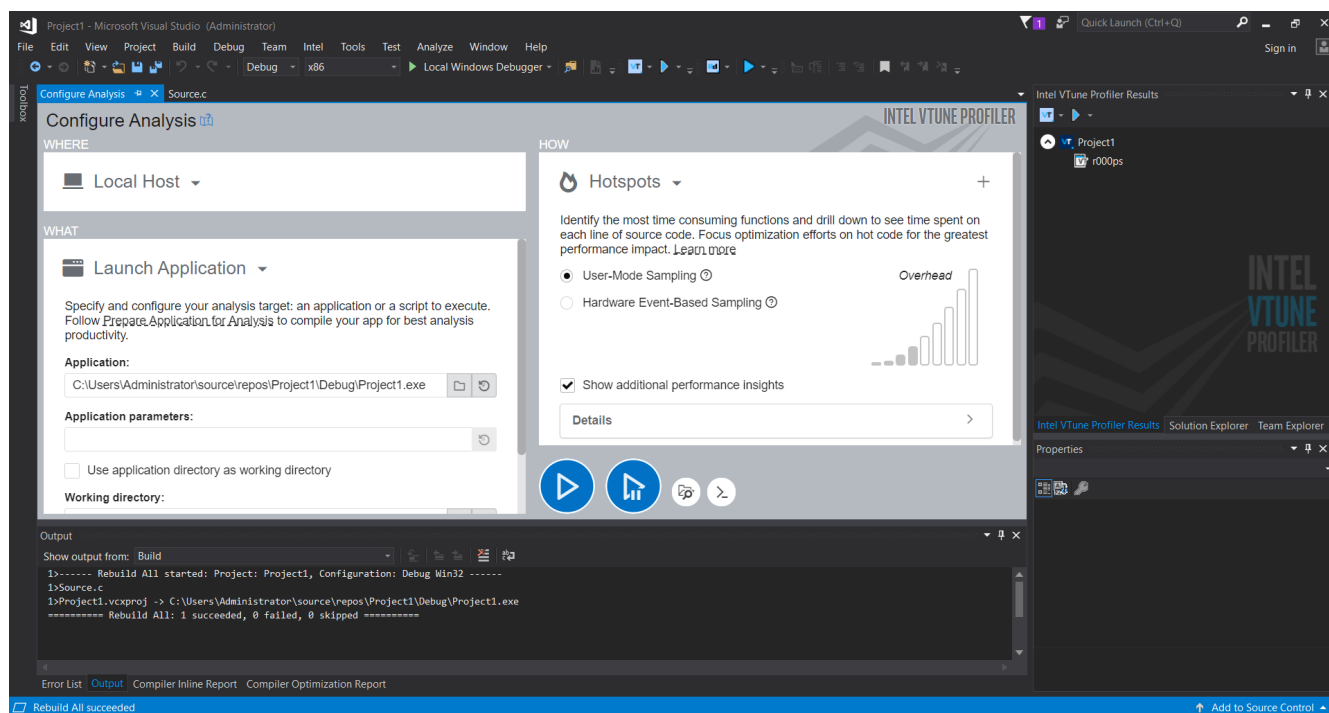
علی قنبری: ۸۱۰۱۹۹۴۷۳

بهراد علمی: ۸۱۰۱۹۹۵۵۷

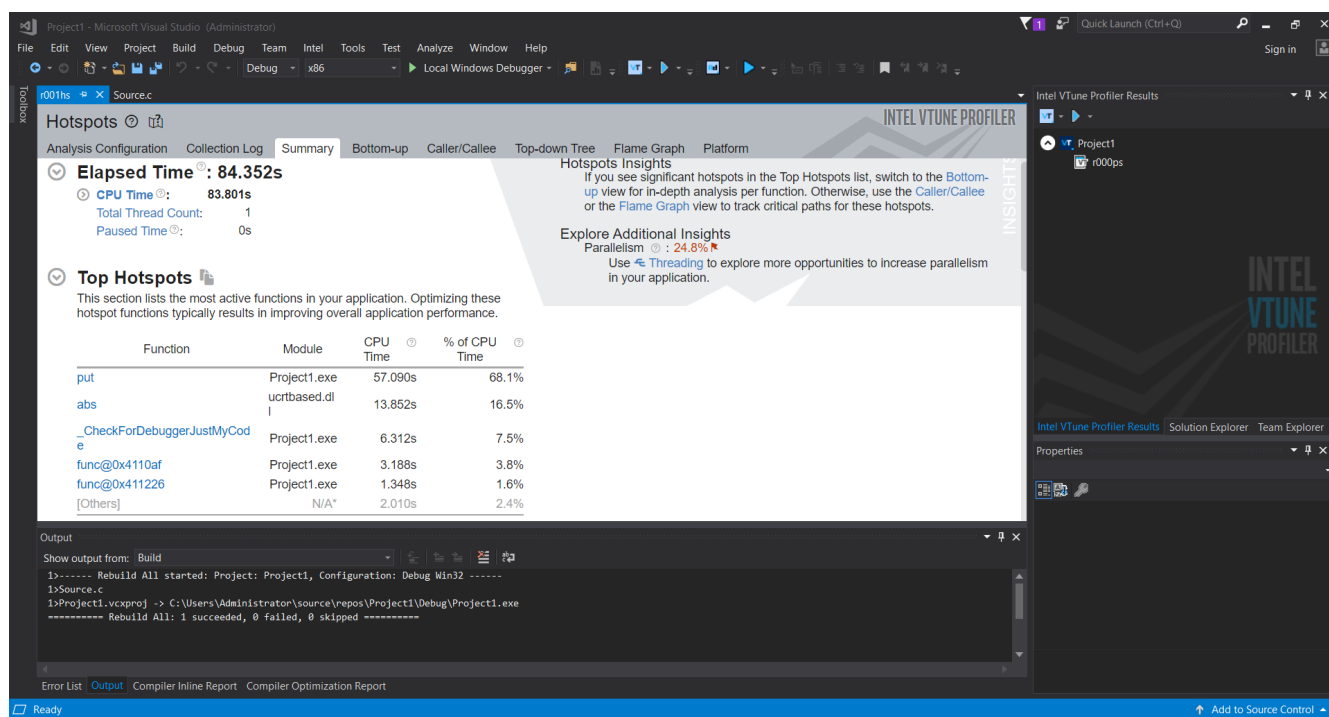
ابتدا امکان صحت‌سنجی و همچنین مقایسه نتایج، برنامه را به صورت عادی کامپایل کرده و نتایج را مشاهده میکنیم:



در گام نخست پروژه میبایست از قسمت ابزارهای نصب شده Vtuner را انتخاب کرده و بخش Hotspot Analysis انتخاب کنیم:



حال به Hotspot Analysis میپردازیم:



نتایج فوق نشانگر این است که تابع put منابع زیادی را به خود اختصاص میدهد. در نتیجه در این تابع موازی سازی هایی را به کمک OpenMP انجام میدهیم:

```

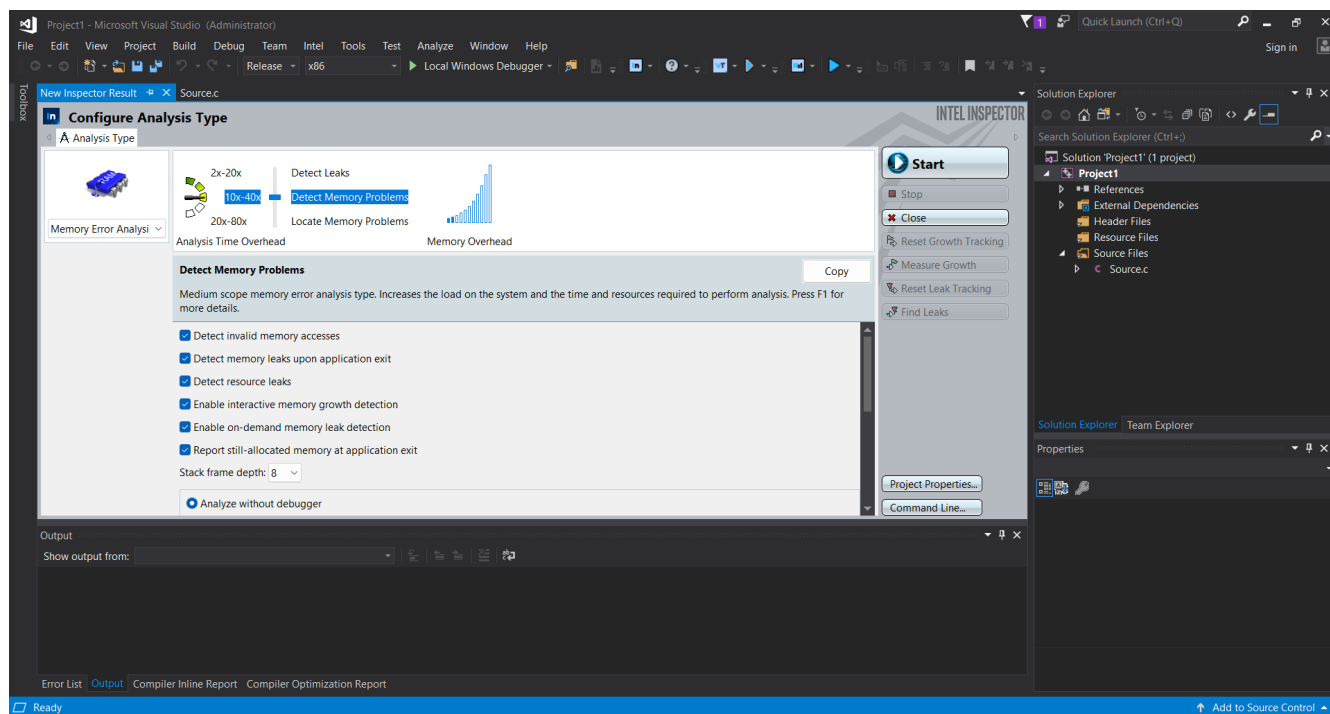
if (row == N - 1) {
    #pragma omp atomic
    solutions++;
}
else {
    #pragma omp parallel
    {
        #pragma omp for
        for (i = 0; i < N; i++) { // increment row
            put(Queens, row + 1, i);
        }
    }
}

```

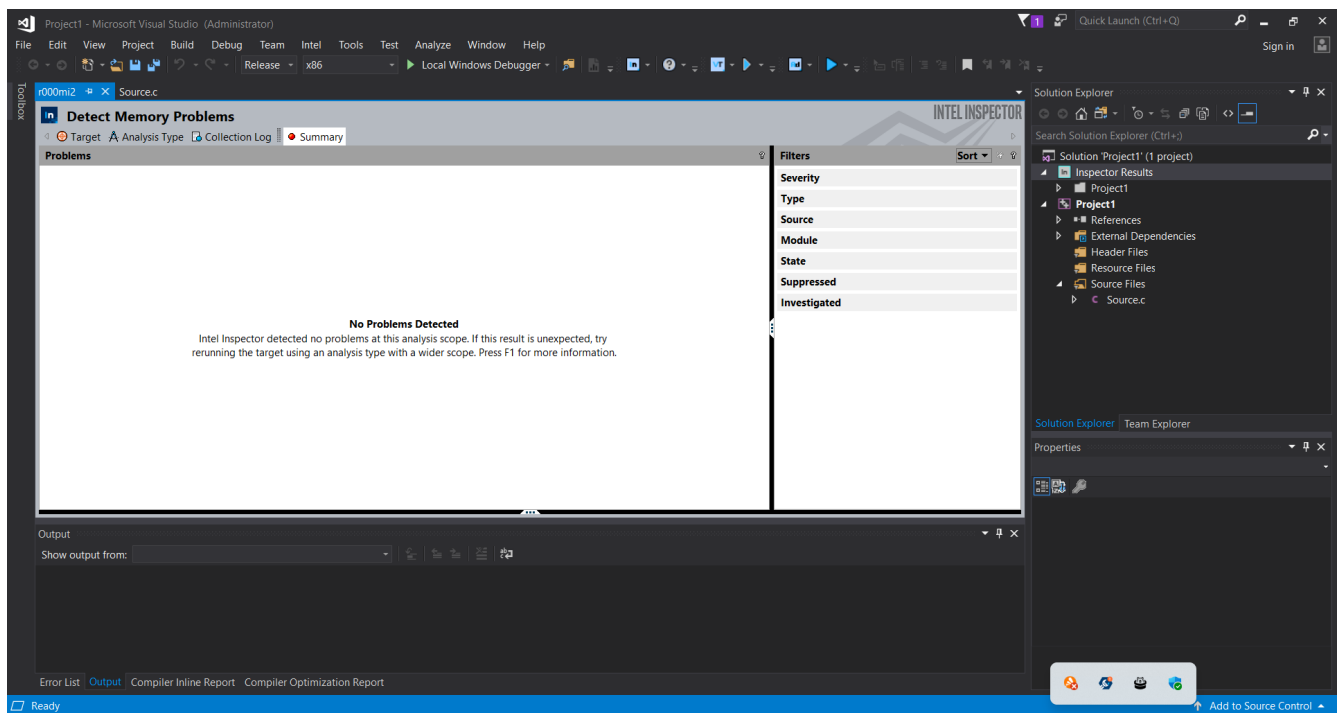
توجه شود که متغیر solutions ممکن است به صورت همزمان میان ریسمان های متفاوت دچار تغییر شود مشکل data race را به وجود آورد.

در گام بعد با استفاده از ابزار inspector به عیب یابی برنامه طراحی شده میپردازیم:

تنظیماتی که به منظور عیب یابی از نظر memory است به صورت زیر است:

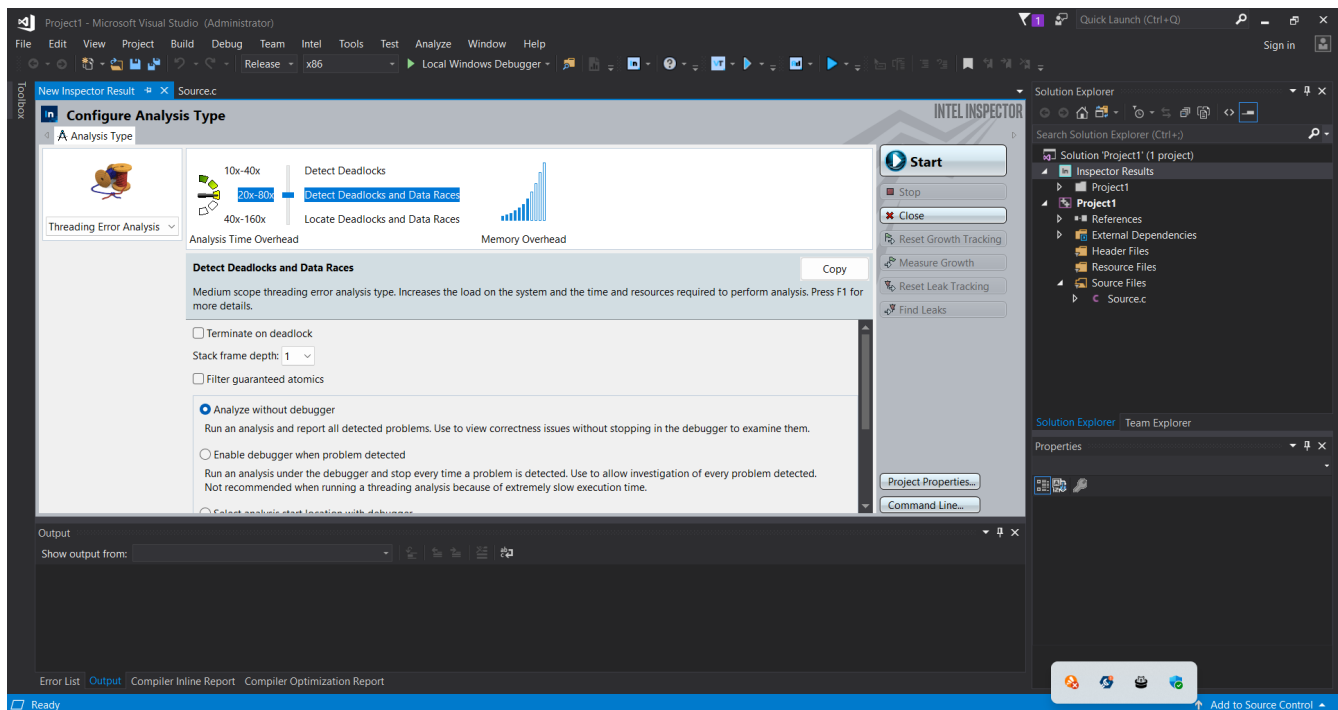


نتایج این تست به صورت زیر است:

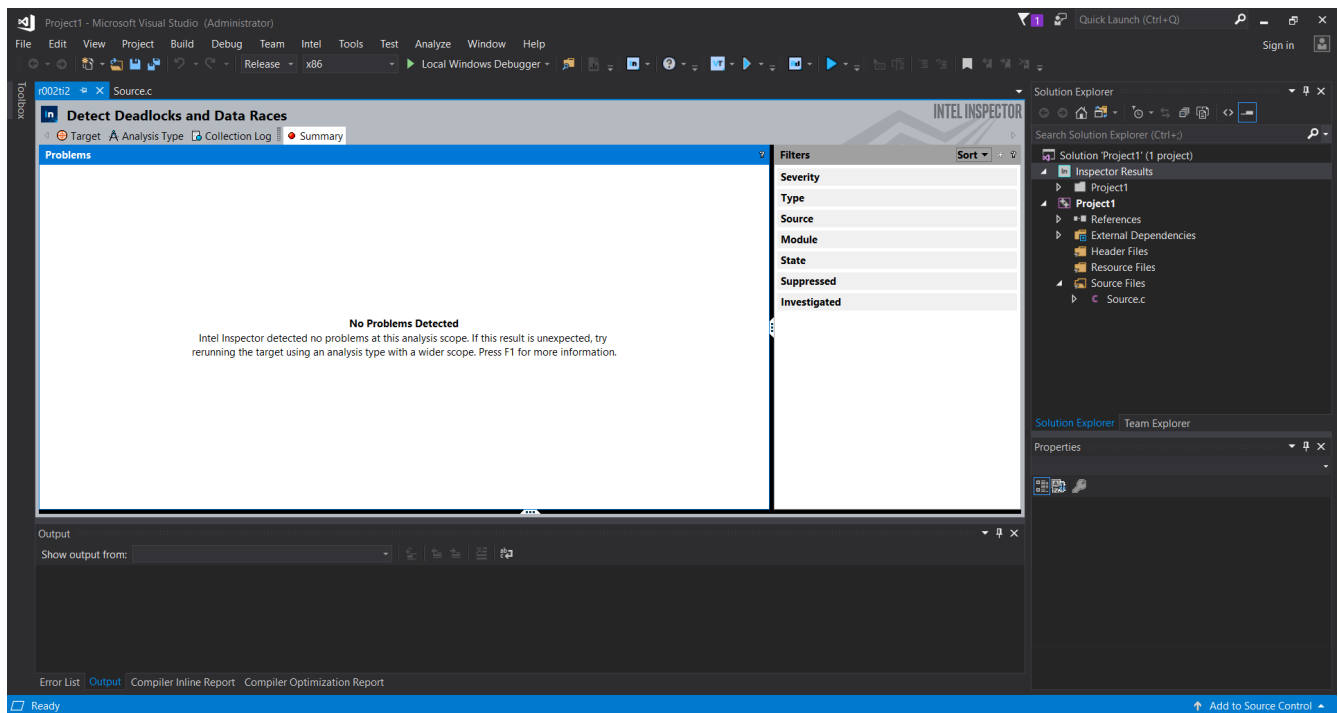


نتایج نشان میدهد که برنامه از نظر memory leak و memory دچار مشکل نیست.

حال بررسی را با تنظیمات زیر برای سناریو های چند ریسمانی انجام میدهیم:

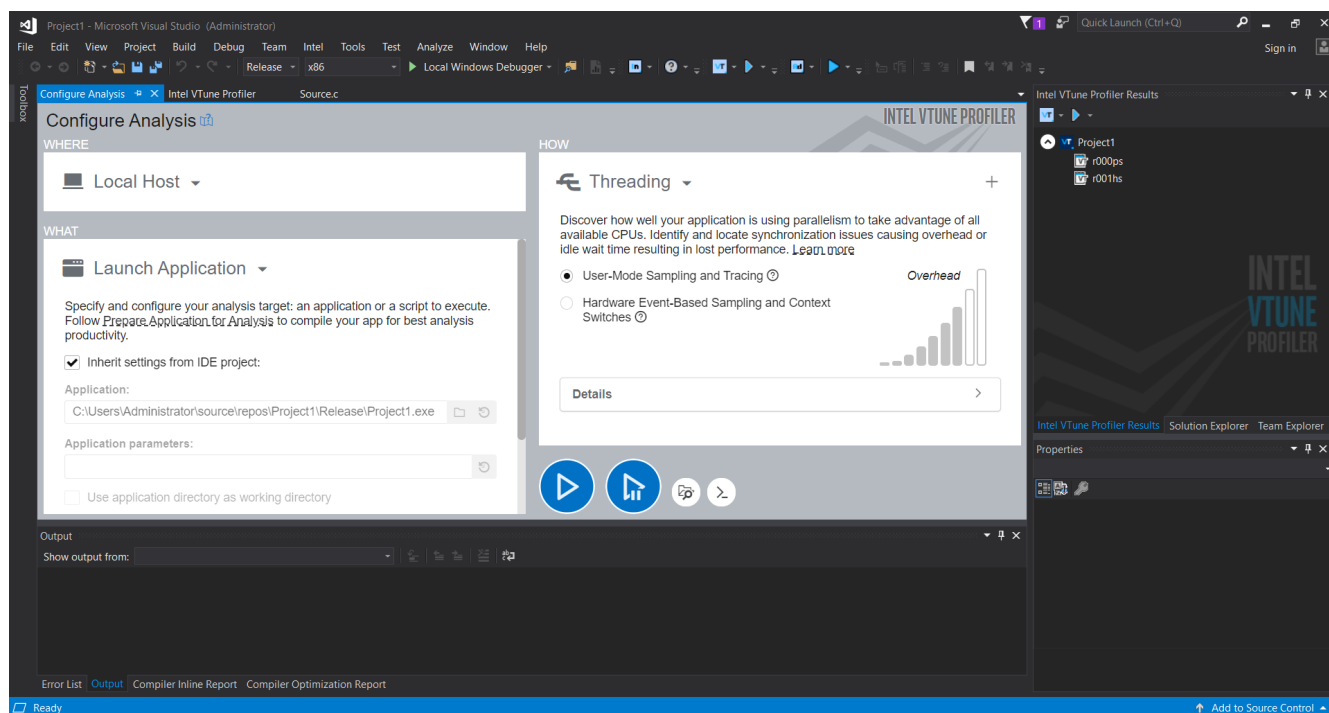


که نتایج زیر را در بر خواهد داشت:

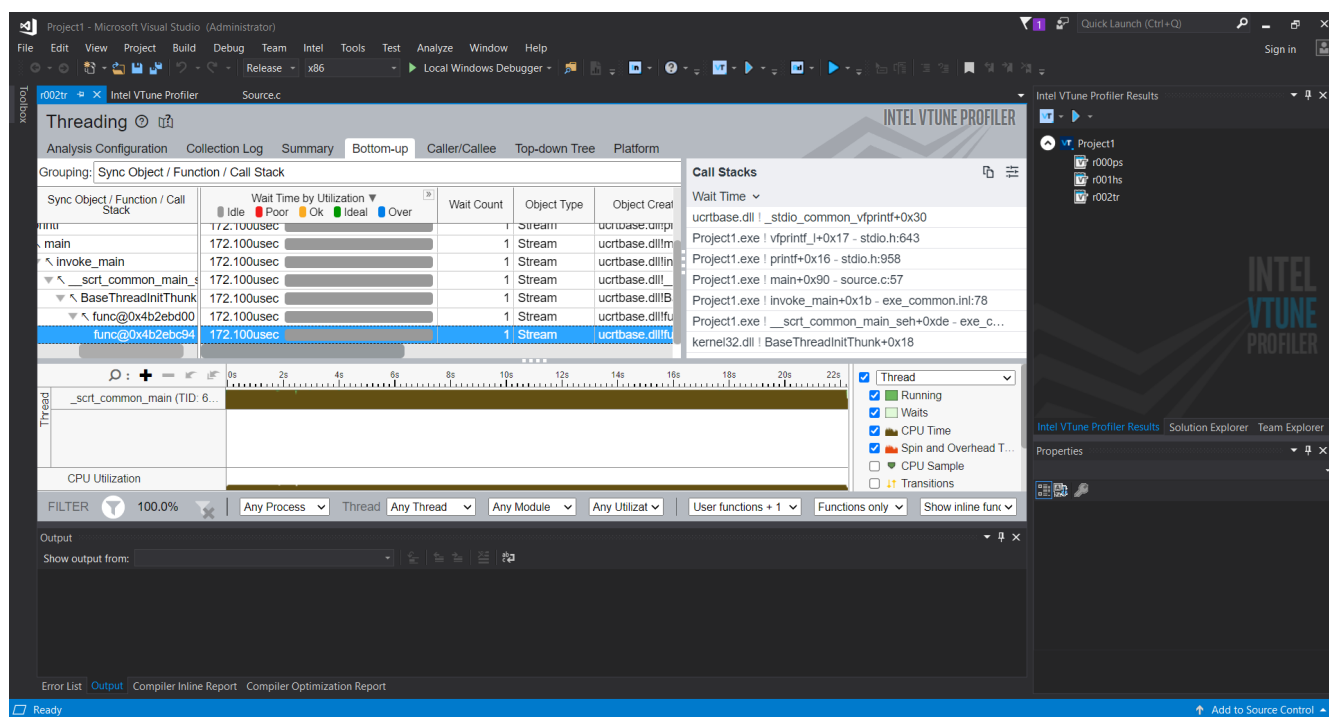


همانطور که مشاهده می‌شود در بخش Deadlock و Data Race نیز برنامه عمل کرد خوبی داشته و مشکلی از این بابت توسط برنامه تشخیص داده نشده است. خاطرنشان میکند در صورتی که در بخش موازی سازی به data race احتمالی بر متغیر solutions توجه نمی‌کردیم در این قسمت دچار ارور میشدیم.

در این بخش به منظور بهینه سازی عملیات موازی سازی از Tuner استفاده میکنیم:



در شکل زیر اطلاعات مربوط به گزارش بدست آمده را مشاهده میکنیم:



همانطور که مشاهده می‌شود اکثر وظایف به دوش یک thread افتاده است به منظور جلوگیری از این اتفاق در بخش parallelization کد زیر را جایگذاری میکنیم:

```
#pragma omp parallel schedule(dynamic) num_threads(3)
{
    #pragma omp for
    for (i = 0; i < N; i++) { // increment row
        put(Queens, row + 1, i);
    }
}
```

حال از برنامه اجرا میگیریم تا نتیجه ذیل حاصل شود:

```
Microsoft Visual Studio Debu
# solutions 2279184 time: 21.000000

C:\Users\Administrator\source\repos\Project1\Release\Project1.exe (process 14400) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

خاطرنشان میکند افزایش سرعت به دلیل محدودیت سخت افزاری چشم گیر نیست در صورت استفاده در یک سیستم با تعداد هسته های بیشتر این برنامه با این تنظیمات با سرعت بیشتری به اتمام میرسد.

در نهایت:

Speedup:

Serial: ۲۳ Speedup: ۱
 Parallel: ۲۲ Speedup: ۱.۰۴۵
 Parallel-Tuner: ۲۱ Speedup: ۱.۰۴۷
 Total Speedup: ۱.۰۹۵