**<https://www.youtube.com/watch?v=p5iwbuAFpUo&t=738s>**

**Static analysis for concurrent C++**

**A summary of Topic**

1. **What is the point the speaker is trying to make?**

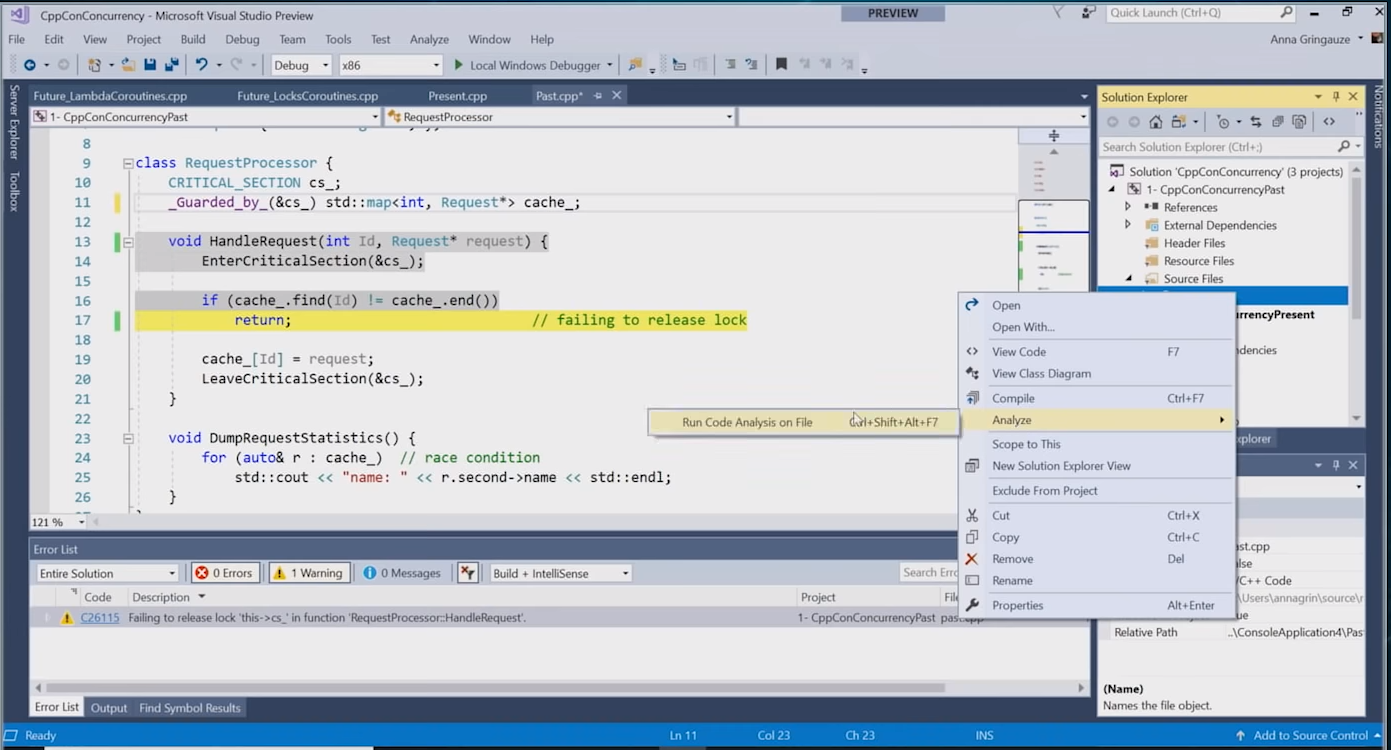
If we used Static analysis, we can prevent the bug in coding in advance.

1. **What are some of motivations for speakers point?**

When you write the code, there can be a bug. To fix a bug, we need to find a bug, but not all bugs are easily found. Then we need ‘ static analysis ‘

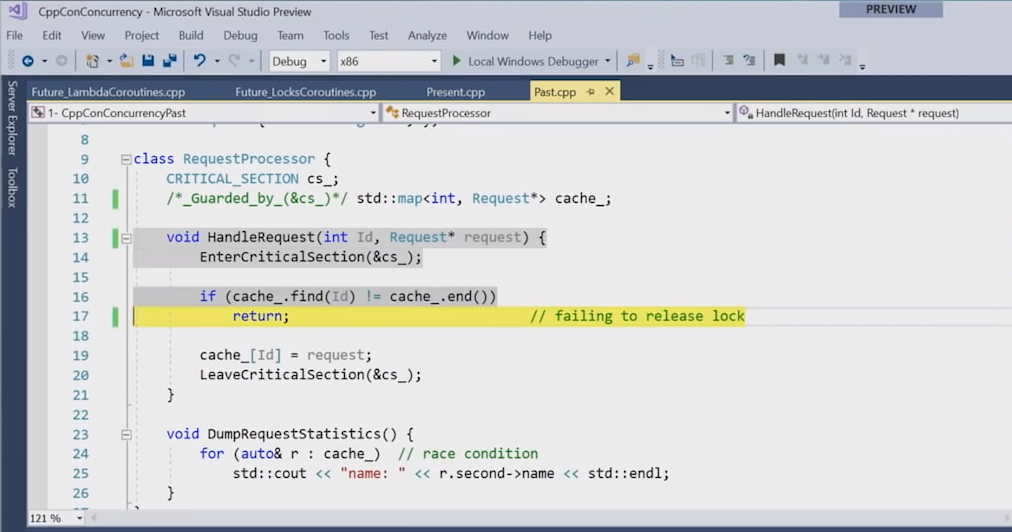
1. **What are some examples?**
2. **Race condition in c**

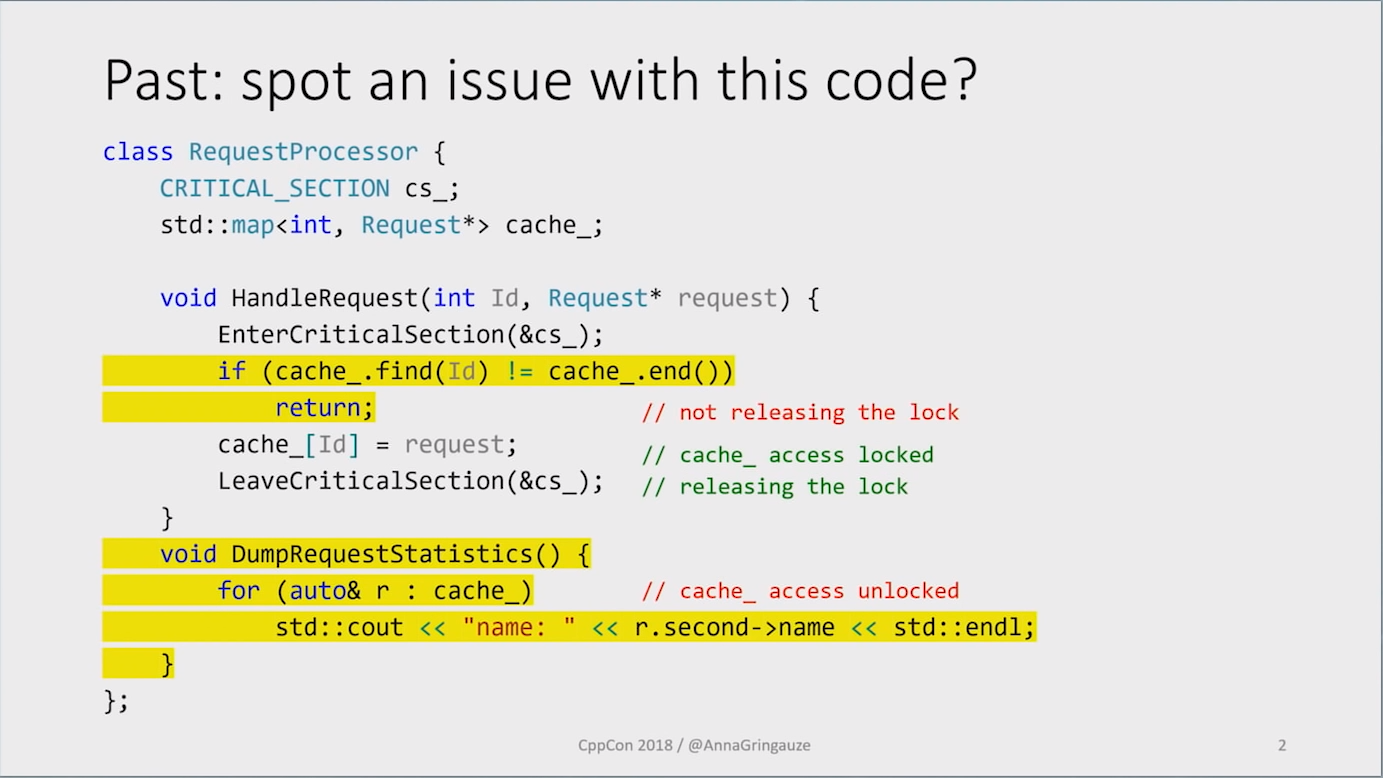
**When we need analysis, people often used Visual Studio’s “Run Code Analysis on File”**

****

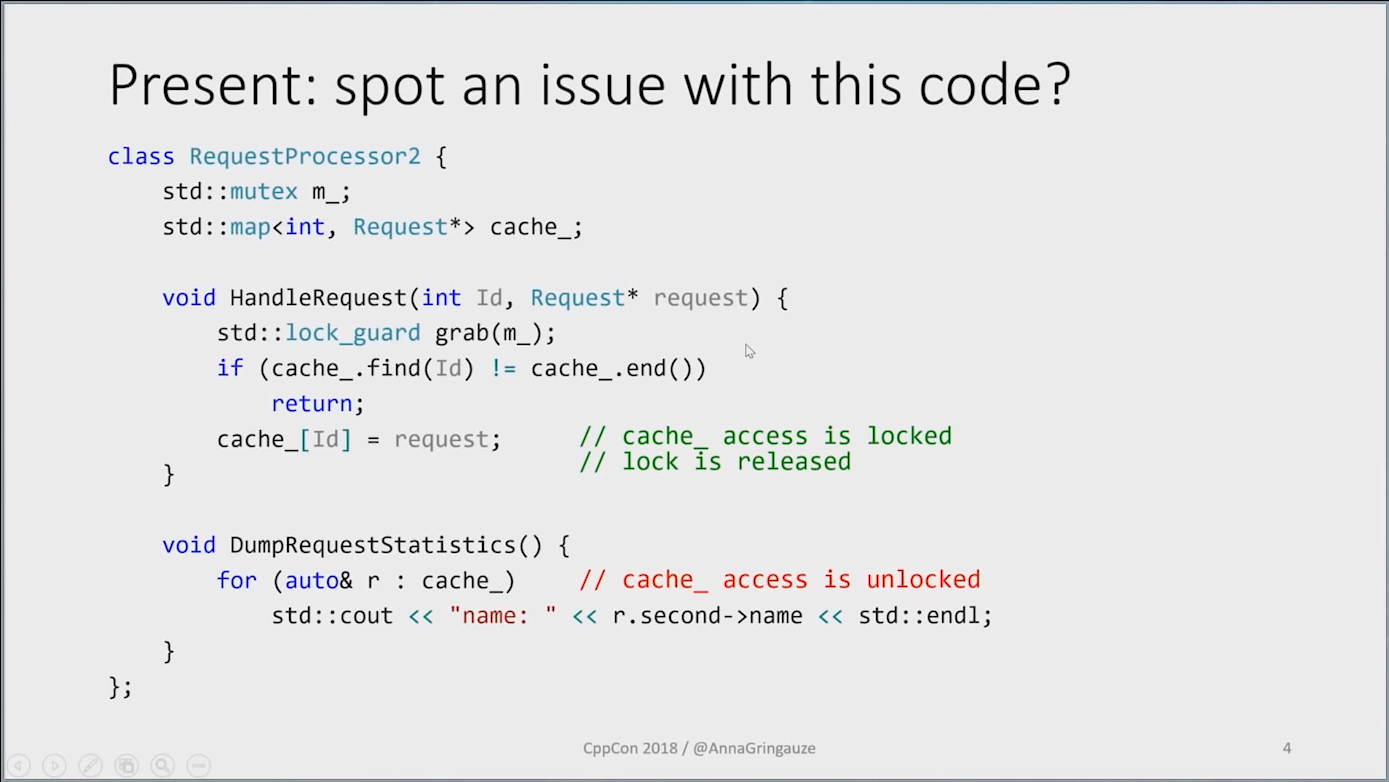
**But it can’t find out all bugs.**

**Ex ) 1.**

****

****

**In this case, “Analysis” find out the first bug but it can’t find out second bug.**

****

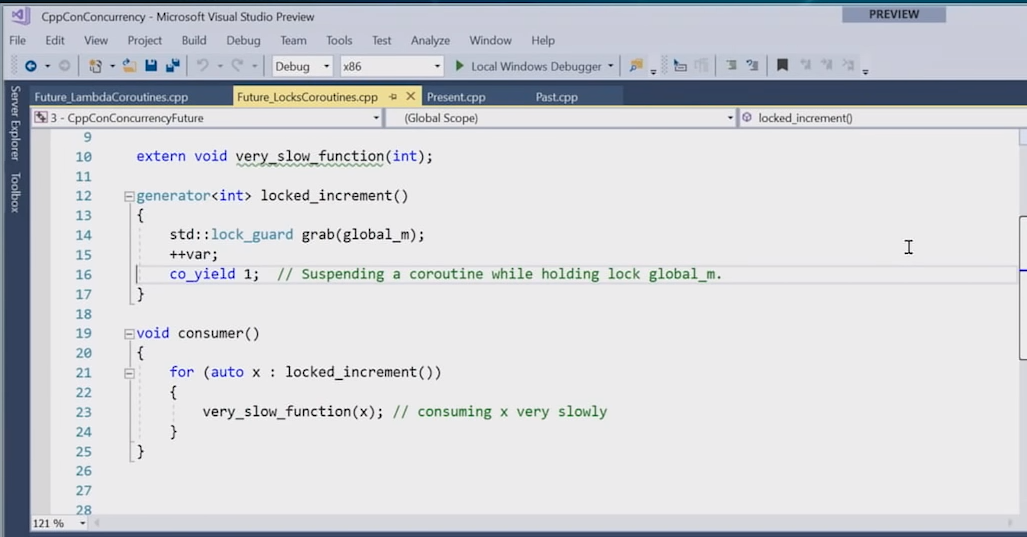
**It can’t detect second bug( race condition )**

**We can find out the bug with \_Guaded\_by\_ in c( static analysis )**

**However, in c++, there are many added features like mutex and lock\_guard**

**Visual Studio Analysis can also detect both bugs in c++(using mutex and lock\_guard)**

**Ex 2 )   
In the future, We will use a lot of concepts like coroutine.( not using thread but do same uses like thread)**

****

**We can use Static Analysis in coroutine.**

**If we analysis upper code, we can get** “Suspending a coroutine while holding a lock global mutex” **message.**

**If there is locking during yielding causes many problems, so we need to prevent it. We can use static analysis to prevent it easily.**

**Why was talk interesting?**

To check for bugs when using threads in CS180 assignment, I have tried using the exe file directly and checked for what points are wrong. However, to use analysis, we can find out what points are wrong from the coding stages, so I was able to have fun with my experience.

**What did you learn from watching the video?**

Static Analysis – in c++, it can find out the bug at a coding stage.