Example Loop from LU Decomposition

```plaintext
for (j = 1; j <= 100; j++)
    for (i = j; i <= 100; i++)
        for (k = 1; k < j; k++)

I = \sum_{j=1}^{100} \sum_{i=j}^{100} \sum_{k=1}^{j-1} 1

= \sum_{j=1}^{100} \sum_{i=j}^{100} (j - 1)

= \sum_{j=1}^{100} ( \sum_{i=1}^{j-1} (j - 1) - \sum_{i=1}^{j-1} (j - 1) )

= \sum_{j=1}^{100} ( \sum_{i=1}^{j} j - \sum_{i=1}^{100} 1 - \sum_{i=1}^{j-1} j + \sum_{i=1}^{100} 1 )

= \sum_{j=1}^{100} (102j - j^2 - 101)

= 102 \sum_{j=1}^{100} j - \sum_{j=1}^{100} j^2 - \sum_{j=1}^{100} 101

= 166,650
```
Implementation

• The timing analyzer is linked with a symbolic solver called Ctadel to compute summations.

• Input to Ctadel:
  — Summation expression
  — If timing analyzer determines that loop nest may be partially zero trip, notify Ctadel that it needs to perform bounds tests.

• Output from Ctadel:
  — In most cases, an integer representing the sum is returned.
  — If Ctadel could not solve the summation due to multiple nonunit strides, timing analyzer computes conservative bounds.