

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

The results are in! [See the 2018 Developer Survey results.](#) »



Generate calling graph for C++ code



Tired of recruiter spam?
Want jobs tailored to your needs?


[Get started](#)

I'm trying to generate calling graph with which to find out all the possible execution paths that are hitting a particular function (so that I don't have to figure out all the paths manually, as there are many paths that lead to this function). For instance:

```
path 1: A -> B -> C -> D
path 2: A -> B -> X -> Y -> D
path 3: A -> G -> M -> N -> O -> P -> S -> D
...
path n: ...
```

I have tried Codeviz and Doxygen, somehow both results show nothing but callees of target function, D. In my case, D is a member function of a class whose object will be wrapped within a smart pointer. Clients will always obtain the smart pointer object through a factory in order to invoke D.

Does anyone know how to achieve this?

`c++` `call-graph`

edited Nov 14 '15 at 13:46



m.s.

11.9k 5 33 57

asked Mar 21 '11 at 4:09



shiouming

736 1 9 20

6 Answers

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

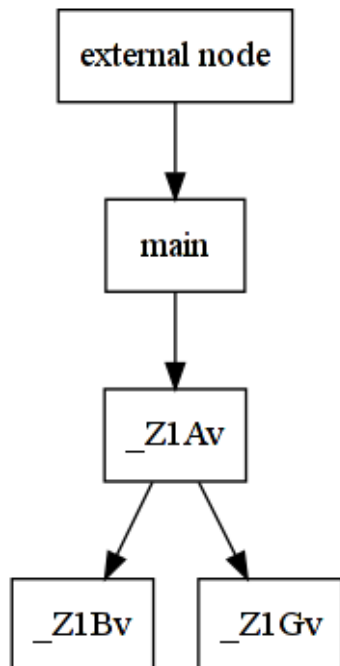
```
static void X() { Y(); }
static void C() { D(); X(); }
static void B() { C(); }
static void S() { D(); }
static void P() { S(); }
static void O() { P(); }
static void N() { O(); }
static void M() { N(); }
static void G() { M(); }
static void A() { B(); G(); }

int main() {
    A();
}
```

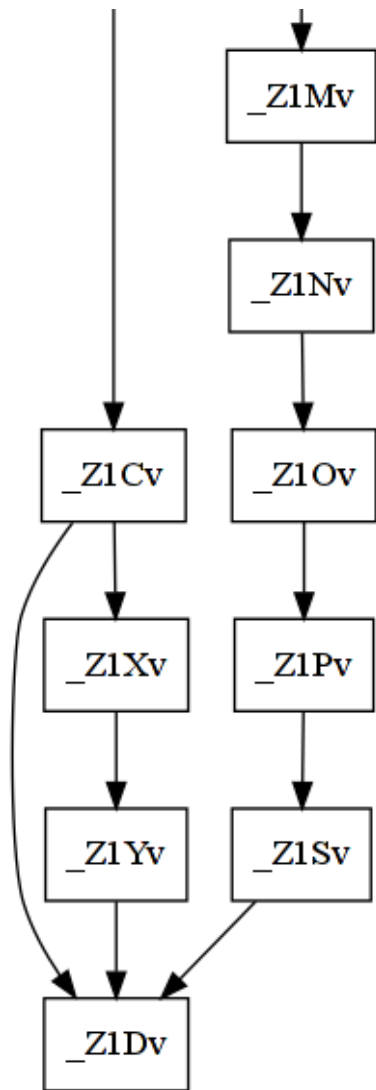
Then

```
$ clang++ -S -emit-llvm main1.cpp -o - | opt -analyze -dot-callgraph
$ dot -Tpng -ocallgraph.png callgraph.dot
```

Yields some shiny picture (there is an "external node", because `main` has external linkage and might be called from outside that translation unit too):



Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.



Call Graph

You may want to postprocess this with `c++filt`, so that you can get the unmangled names of the functions and classes involved. Like in the following

```
#include <vector>
```

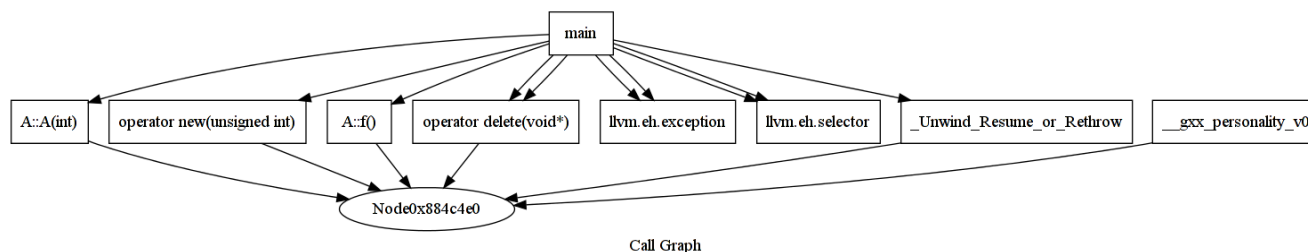
Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
void f(); // not defined, prevents inlining it!
};

int main() {
    std::vector<A> v;
    v.push_back(42);
    v[0].f();
}

$ clang++ -S -emit-llvm main1.cpp -o - |
  opt -analyze -std-link-opts -dot-callgraph
$ cat callgraph.dot |
  c++filt |
  sed 's,>,\>,g; s,-\>,->,g; s,<,\<,g' |
  gawk '/external node/{id=$1} $1 != id' |
  dot -Tpng -ocallgraph.png
```

Yields this beauty (oh my, the size without optimizations turned on was too big!)



That mystical unnamed function, Node0x884c4e0, is a placeholder assumed to be called by any function whose definition is not known.

edited Mar 5 '14 at 22:25



Catskul

7,229 9 61 95

answered Mar 21 '11 at 4:30



Johannes Schaub - litb

385k 89 740 1098

17 Have you done this on a multi file project ? looks very cool as a tool – [dirvine](#) Oct 10 '12 at 22:05

2 +1 For some reason I had to pass the -n option to c++filt for the names to unmangle. Thought I'd mention it here in case anyone else faces the same issue. – [Aky](#) Jan 4 '14 at 10:08

1 I get an error when trying this: Pass::print not implemented for pass: 'Print call graph to 'dot' file'! What's up with that? clang 3.8 – [Arne](#) Sep 11 '15 at 10:32


Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

Something other than `../callgraph.dot` : - [Amir](#) Sep 11 '15 at 10:31

- 1 When I do this with clang-3.5 on Ubuntu I get. opt: <stdin>:26:93: error: expected value token invoke void
@_ZNSt6vectorI1ASaIS0_EE9push_backERKS0_("%"class.std::vector"* %v, %struct.A* dereferenceable(1)
%1) - [ppetraki](#) Sep 16 '15 at 17:32



Tired of recruiter spam?
Want jobs tailored to your needs?



Get started

You can achieve that by using doxygen (with option to use dot for graphs generation).

Diagrams to generate

☐ No diagrams

☐ Use built-in class diagram generator

☒ Use dot tool from the GraphViz package

Dot graphs to generate

☒ Class diagrams

☒ Collaboration diagrams

☒ Overall Class hierarchy

☒ Include dependency graphs

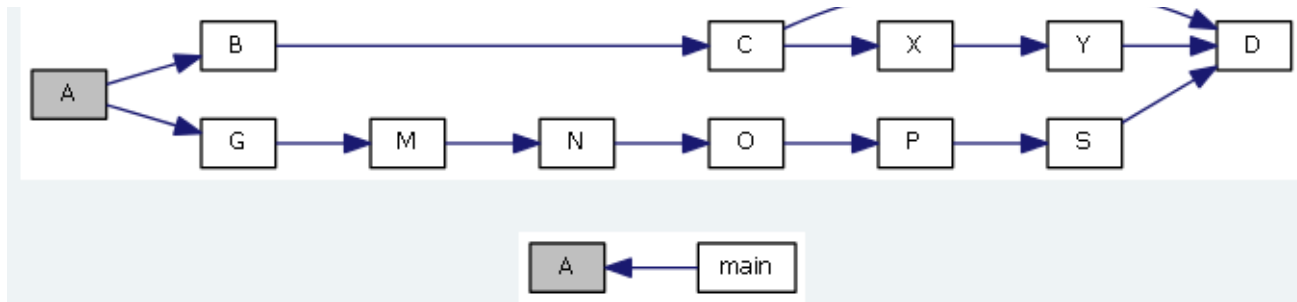
☒ Included by dependency graphs

☒ Call graphs

☒ Called by graphs

With Johannes Schaub - litb main.cpp, it generates this:

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.



doxygen/dot are probably easier than clang/opt to install and run. I did not manage to install it myself and that's why I tried to find an alternative solution!

answered Dec 3 '15 at 10:28



[jpo38](#)

12.1k 2 28 72

1 Could you add an example of how to run doxygen to get the window that you included? – [nimble_ninja](#) Mar 7 '17 at 23:21

@nimble_ninja: Isn't the screenshot from doxywizard configuration dialog enough? – [jpo38](#) Mar 8 '17 at 6:20

I didn't know that it was from doxywizard. Thanks! – [nimble_ninja](#) Mar 8 '17 at 12:18

Statically computing an accurate C++ call graph is hard, because you need a precise language parser, correct name lookup, and a good points-to analyzer that honors the language semantics properly. Doxygen doesn't have any of these, I don't know why people claim to like it for C++; it is easy to construct a 10 line C++ example that Doxygen erroneously analyzes).

You might be better off running a [timing profiler which collects a call graph dynamically](#) (this describes ours) and simply exercise a lot of cases. Such profilers will show you the actual call graph exercised.

EDIT: I suddenly remembered [Understand for C++](#), which claims to construct call graphs. I don't know what they use for a parser, or whether they do the detailed analysis right; I have no specific experience with their product.

I am impressed by Schaub's answer, using Clang; I would expect Clang to have all the elements right.

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.



shiouming

76.5k

10

119

243

Unfortunately I'm not aware of all the use cases that may trigger that function :(. In fact, my ultimate goal is to find out the exact list of use cases which utilizing that function for debugging purpose. I'm able to find out the direct callers with code indexing tool, but need to figure out all the execution paths for further analysis. –

[shiouming](#) Mar 21 '11 at 10:40

In order for the `clang++` command to find standard header files like `mpi.h` two additional options should be used `-### -fsyntax-only`, i.e. the full command should look as:

```
clang++ -### -fsyntax-only -S -emit-llvm main1.cpp -o - | opt -analyze -dot-  
callgraph
```

answered Jul 3 '14 at 14:09



mabalenk

117

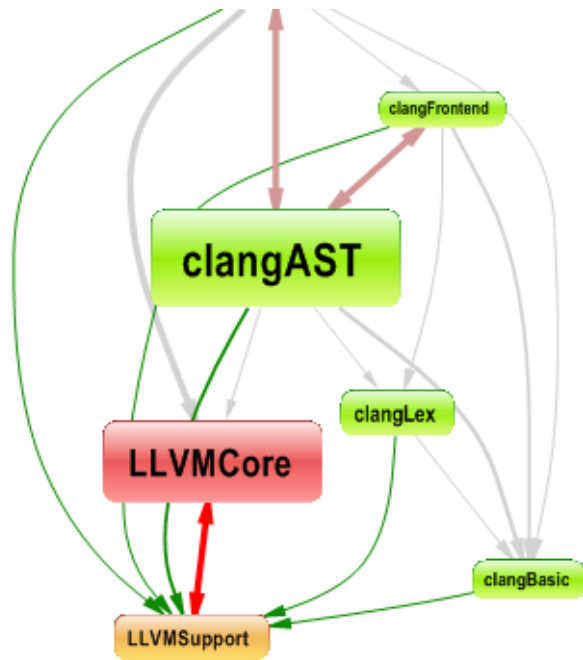
1

7

You can use [CppDepend](#), it can generates many kinds of graphs

- Dependency Graph
- Call Graph
- Class Inheritance Graph
- Coupling Graph
- Path Graph
- All Paths Graph
- Cycle Graph

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.



answered Feb 6 at 8:38



Issam

71 1 2

The "C++ Bsc Analyzer" can display call graphs - by reading the file generated by the bscmake utility.

answered Sep 24 '14 at 20:32



Resonantium

11 2