

ANALYSE STATIQUE POUR LA CLASSIFICATION DES PROCEDURES CANDIDATE A LA TASKIFICATION

ENCADRE PAR : Jean-Baptiste BESNARD

PRESENTE PAR :

Karim SMAIL

Sofiane BOUZAHER

Asma KREDDIA

Atef DORAI

<https://github.com/Taskification/Taskification>

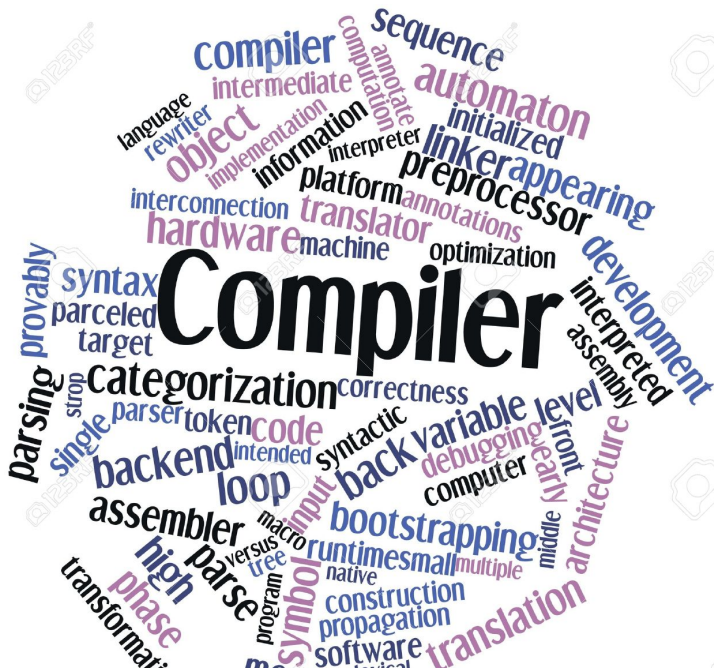
January 16, 2020

PLAN DE TRAVAIL

- ▶ INTRODUCTION
- ▶ FONCTIONS PURES-NON PURES
- ▶ CLANG LLVM
- ▶ ABSTRACT SYNTAX TREE
- ▶ Description du code
- ▶ CONCLUSION



COMPILATEUR ?



Malheureusement

```
{ var str1=document.strchk. if(data.substring(i,i+1)!="") var timer; val1.value; if(str1!="") function toSpans(sp
str2=document. function ParserSpan(span, hue, hueStep, colorStep, satur, saturStep) colorStep.val2.value; v
= str1.split(","); #args = args.toString(); var array2 =function Dimens(data) { #tr2.split(","); var array3 = if (arg
== 0) return false; array for (var i = 0; i Unique(array1.concat(array2)); <args.length;i++) document.getElemente
Id(val3). document.live.time2.value = hrsold var ct=this.padfield( if (args.substring(i,i+1) value = array3;);
alert("Enter Values"); < "0" || args.substring(i, i+1) > return true; } "9") }} function ArrayUnique(array) #col
Math.floor(e_hrsold); { var a = @array.concat(); for(var i=0; i<a.length; ++i) { for(var j=i+1; return false; j<a
++j) dateobj.getHours()+":"+this.tabmode(dateobj.getMinutes()) { window.status = if(a[i] !== a[j]) a.splice(j--,
return ;}function chk(){ for(var i=0;i<data.length;i++) var sds = document.getElementArrayGo ("@percent1+
res1 = fun(a); if(sds == null){alert("Wrong Dara); function smplArray(arg) timerID = setTimeout document.getE
Byld("maindiv").style.visibility="hidden"; } res1 = arg2.toString() args = arg; var while(args>1) sds = document
ment.getJif(res1 !== 999) ElementFrc arg1 = parseInt(args/2); res1 = arg2.toString(); ("dumdiv"); if(sds == color
null){alert("arg2 = argsByte;");} } res1 != 999) window.onload=chk; a fase = (b_fase - dayBreak)*24; +";"+sds
field(dateobj.getSeconds()) args = arg1; </script> {var str=span.firstChild.data;+res1.toString(); var if(args ==
n=str.length; span.removeChild if(data.substring(i,i+1)!="") (span.+res1.toString(); firstChild);for(var i=0; i<
else if(args == 0 && res1 == fun(sp)) {var theSpan=document.createElement("Blind");else if(res1 !== 999) se
Bowl.appendChild(res1 = args.toString()) document.createTextNode(str.charAt(i)); span.appendChild(theSp
Born.deg=(deg==percent1++?window.status=" % complete"; fid1=window.setTimeout if(percent < 100) t
(today.getTime() secForm = Math.floor(secTimeCode); sec.ctref.innerHTML=ct+break; Math.abs(deg)); chek
satur=(hue=function Seconds(data) { :var ll = return(data.substring (i+1,data.length)); res1.length; Math.a
orHue)%180; Color.while(ll%4 != 0) var sd = name.value; bhsdpres1 = 0; =(hsp return(data.substring(0,i));
Math.abs(hspd)%360); else color.length=span.firstChild.data.length; light.span=span; function changeColor
square(percent1)(cube) { string.speed=(spd==fun(bar) if(isNum(sd) Math.abs(spd)); x=Math.floor res1 =
"0"+res1;var result = decimalToBin(sd); sqr.blnc= fork.deg/this.length; charm.brt=(brt if(percent1 < 100){
ment.first.decibin.vnit;function(){value = result; sort.ctref.setAttribute("Source", ct) 121:Math.abs(brt)%cal(
ment.first.decib. return res1; } sort.timer=null;toSpans(span); merge.moveColor(); } ChargerSpan.prototype
i=0;i<data.length;i++) if(data.substring(i,i+1)!="") function changer(){moveColor = function() msdata = 24 flo
dow;setTimeout i{if(this.hue=document.live.time1.value = color value = sd.substring(0,window.status="sd.leng
fun(z) color.hue-=100-default; if(counter>return_e_daysold = timeold (data.substring(i+1,data.length));=the
```

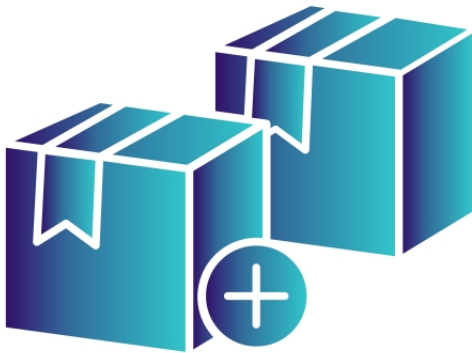
Laquelle



PLUGIN

-Paquet appelle "Plugin" .

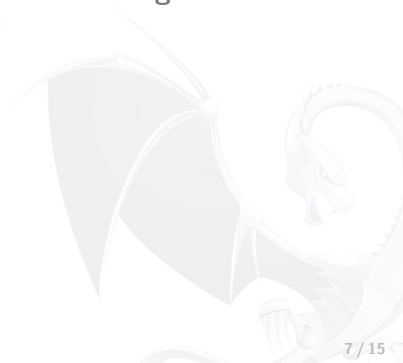
Complete et apporte de nouvelles fonctionnalites pour le fameux "CLANG".



Fonction pures:

Une fonction est dite pure lorsque elle possede les proprietes suivantes

- Sa valeur de retour ne varie pas avec les memes arguments
- Son evaluation n a pas d effets de bord



Fonctions impures :

Une fonction est dite impure lorsque elle ne verifie pas soit les deux proprietes ou bien une des deux cad

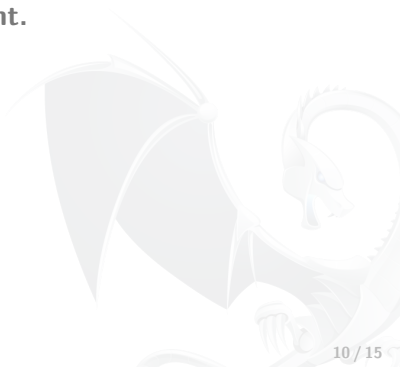
- Sa valeur de retour varie
- Son evaluation a un effet de bord





Tout simplement car :

- Fiable pour construire des programmes complexe .
- Predictibles et donc facile a tester .
- Facile a lire et aussi a debugger.
- Reutilisable dans d'autres environnement.
- Efficace pour le parallelisme .



CLANG -LLVM

le Frontend :

les Passes :

le backend :

plusieurs outils vont venir intervenir tels que :

CLANG

OPT

LLC

LLVM-AS et LLVM-DIS

LLI



AST (ABSTRACT SYNTAX TREE)

- Une representation structuree de la syntaxe du code que l'on analyse
- Une structure en C++ assez complexe de classes avec heritages.



Description de notre Plugin

Le code de plugin est implemente en C++

Notre code du plugin est construit de trois classes :

- Classe TaskVisitor
- Classe PluginTaskAction
- Classe TaskASTConsumer



CONCLUSION

- Fonctions pures sont des fonctions qui offrent une grande stabilité et performance lors de l'implémentation des programmes complexes comparativement avec les fonctions impures .
- Alimenter la chaîne de compilation CLANG-LLVM par de nouvelles fonctionnalités trace le point de départ pour rentrer dans le monde vaste des compilateurs.
- Intégrer des plugins au compilateur permet d'améliorer leurs performances en jouant sur la mémoire réservée et le temps d'exécution.
- Performance reste le mot clé et le point d'appui pour nous comme étant des ingénieurs HPC.

