Rapport Taskification

Introduction

Ut in incididunt deserunt aute veniam qui quis. Lorem ipsum sint pariatur anim aliquip. Tempor sit eu reprehenderit laborum enim sint minim duis. Ad qui deserunt enim cillum voluptate incididunt mollit occaecat excepteur duis nulla Lorem anim ullamco. Id occaecat pariatur amet nostrud velit amet dolor. Est irure ullamco laboris enim est. Magna officia mollit anim sit velit aliqua. # Description du Projet

Titre du Sujet

Analyse Statique Pour la Classification des Procédures Candidate à la « Taskification »

Mot Clef

LLVM, Analyse statique, compilation, MPI + X, parallélisation automatique

Description Générale

Les architecture hybrides convergées à venir posent la question des modèles de programmation. En effet MPI depuis l'avènement des architectures many-core a dû être combiné avec du parallélisme intra-noeud en OpenMP (MPI + X). Le mélange de ces modèles se traduit nécessairement par une complexité accrue de l'expression des codes de calcul. Dans ce travail nous proposons de prendre cette tendance à contre-pied en posant la question de l'expression de tâche de calcul en pur MPI. Les étudiants se verront fournir une implémentation de Remote Procedure Calls (RPC) implémentés en MPI, le but du travail et de détecter quelles fonctions sont éligibles à la sémantique RPC statiquement lors de la phase de compilation (c.a.d. les fonction dites « pures »: indépendantes du tas, des TLS, etc ...). Le travail visera le compilateur LLVM dans lequel une passe sera rajoutée pour lister l'ensemble des fonctions éligibles à la sémantique RPC. Pour exemple, une implémentation d'un algorithme de cassage de mot de passe en MPI sera fournie avec pour but sa conversion en RPC producteur/consommateur (github.com/besnardjb/MPI_Brute/) avec l'outil.

Markdown Examples

h1 Heading 8-)

h2 Heading

h3 Heading

h4 Heading

h5 Heading h6 Heading

Horizontal Rules

Typographic replacements

Enable typographer option to see result.

(c) (C) (r) (R) (tm) (TM) (p) (P) +-

test.. test... test..... test?..... test!....

!!!!!! ???? „ – —

“Smartypants, double quotes” and ‘single quotes’

Emphasis

This is bold text

This is bold text

This is italic text

This is italic text

~~Strikethrough~~

Blockquotes

> Blockquotes can also be nested... > ... by using additional greater-than signs right next to each other... > > ... or with spaces between arrows.

Lists

Unordered

- Create a list by starting a line with `+`, `-`, or `*`
- Sub-lists are made by indenting 2 spaces:
  - Marker character change forces new list start:
    - Ac tristique libero volutpat at
    - Facilisis in pretium nisl aliquet
    - Nulla volutpat aliquam velit
- Very easy!

Ordered

1. Lorem ipsum dolor sit amet
2. Consectetur adipiscing elit
3. Integer molestie lorem at massa
4. You can use sequential numbers...
5. ...or keep all the numbers as `1.`

Start numbering with offset:

57. foo
58. bar

Code

Inline `code`

Indented code

```
// Some comments
line 1 of code
line 2 of code
line 3 of code
```

Block code "fences"

```
Sample text here...
```

Syntax highlighting

```
var foo = function (bar) {
  return bar++;
};

console.log(foo(5));
```

Tables

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

Right aligned columns

Option	Description
data	path to data files to supply the data that will be passed into templates.
engine	engine to be used for processing templates. Handlebars is the default.
ext	extension to be used for dest files.

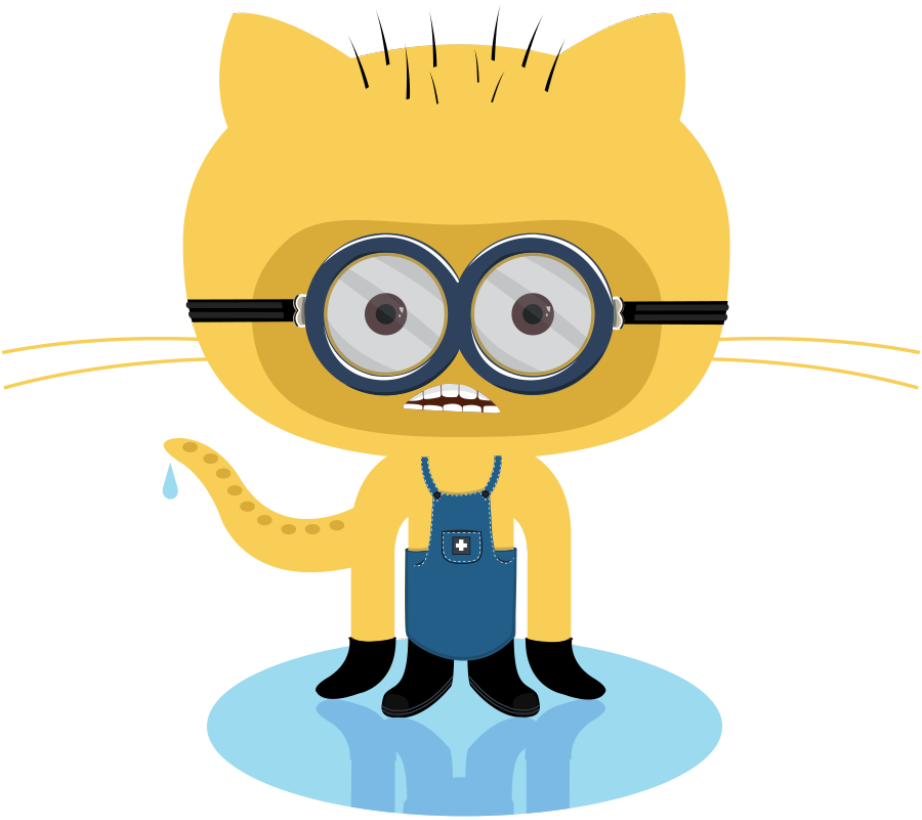
Links

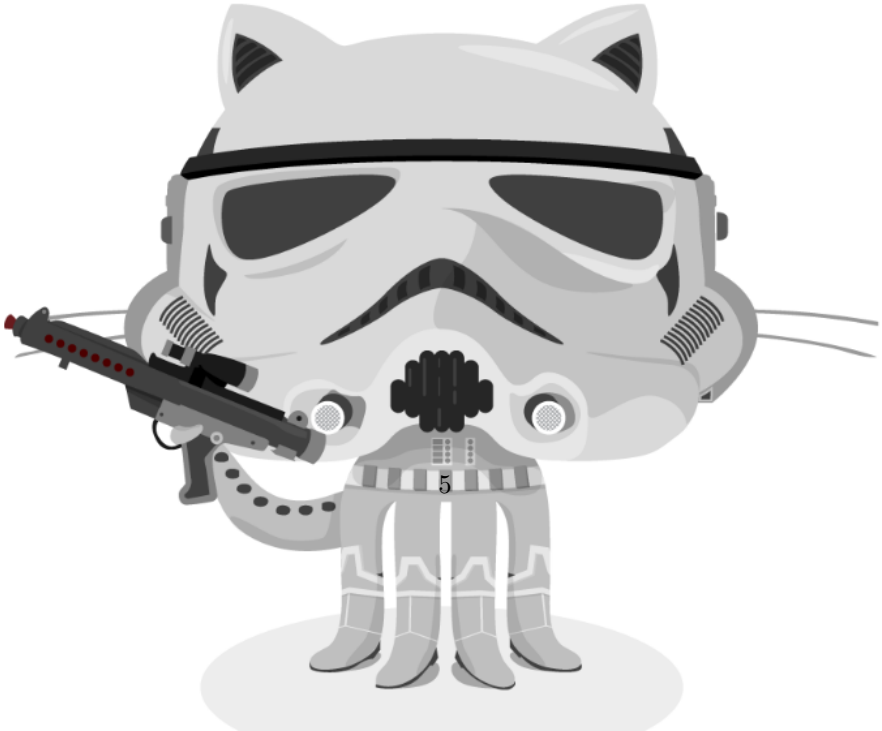
link text

link with title

Autoconverted link https://github.com/nodeca/pica (enable linkify to see)

Images





Like links, Images also have a footnote style syntax

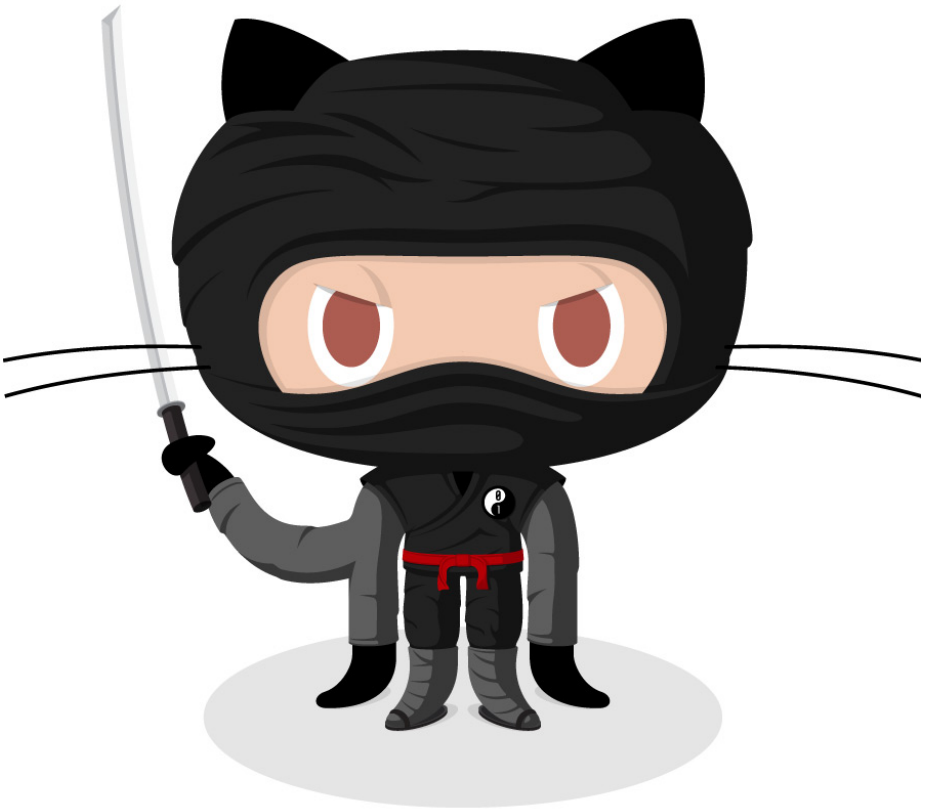


Figure 1: Alt text

With a reference later in the document defining the URL location:

Plugins

The killer feature of `markdown-it` is very effective support of syntax plugins.

Emojies

> Classic markup: :wink: :crush: :cry: :tear: :laughing: :yum:
>
> Shortcuts (emoticons): :-) :-(8-) ;)

see how to change output with twemoji.

Subscript / Superscript

- 19^{th}
- H_2O

<ins>

++Inserted text++

<mark>

==Marked text==

Footnotes

Footnote 1 link[1].

Footnote 2 link[2].

Inline footnote[3] definition.

Duplicated footnote reference[4].

Definition lists

Term 1 Definition 1 with lazy continuation.

Term 2 with *inline markup* Definition 2

```
{ some code, part of Definition 2 }
```

Third paragraph of definition 2.

Compact style:

Term 1 Definition 1
Term 2 Definition 2a
Definition 2b

Abbreviations

This is HTML abbreviation example.

It converts "HTML", but keep intact partial entries like "xxxHTMLyyy" and so on.

*[HTML]: Hyper Text Markup Language

[1] Footnote **can have markup**
and multiple paragraphs.
[2] Footnote text.
[3] Text of inline footnote
[4] Footnote text.

Custom containers

here be dragons