Dead and Core reproduction

Jeffrey M. Young

15 May 2020

In this walk-through we'll perform the plot generation for the dead and core analysis. We assume you have read the other walk-throughs.

Preliminaries

First, libraries, note that I have silenced the shadowing warnings from R:

```
library(ggplot2)   #plotting
library(dplyr)     #dataframe manipulation
library(tidyr)     #dataframe manipulation
library(broom)     #for the tidy function
library(scales)    #for scientific function
library(latex2exp) #for TeX in Labels
library(ggpubr)    #for stat_cor
```

And now we'll load the data, and perform some simple munging to select relevant columns.

```
## Load files
finResultsFile <- "../../data/fin_dead_core.csv"
autoResultsFile <- "../../data/auto_dead_core.csv"

## The usual manipulations
finData <- read.csv(file=finResultsFile) %>%
        mutate(Algorithm = as.factor(Algorithm), Config = as.factor(Config)) %>%
        mutate(Algorithm = gsub("-->", "\U27f6", Algorithm))

autoData <- read.csv(file=autoResultsFile) %>%
         mutate(Algorithm = as.factor(Algorithm), Config = as.factor(Config)) %>%
         mutate(Algorithm = gsub("-->", "\U27f6", Algorithm))

## combine the data frames
finDF <- finData %>% mutate(data = "Fin")
autoDF <- autoData %>% mutate(data = "Auto")

data <- rbind(finDF, autoDF) %>%
  select(Mean, Algorithm, data, ChcCount, PlainCount, Config)
```

Now let's look at the dataset

```
str(data)
```

```
## 'data.frame':    8 obs. of  6 variables:
##  $ Mean      : num  11.05 8.07 9.51 8.42 211.7 ...
##  $ Algorithm : chr  "v v" "p v" "p p" "v p" ...
##  $ data      : chr  "Fin" "Fin" "Fin" "Fin" ...
##  $ ChcCount  : int  3810 3810 3810 3810 4213 4213 4213 4213
##  $ PlainCount: int  1441 1441 1441 1441 26808 26808 26808 26808
##  $ Config    : Factor w/ 2 levels "V1*V2*V3*V4*V5*V6*V7*V8*V9*V10",..: 1 1 1 1 2 2 2 2
```

```
data
```

```
##         Mean Algorithm data ChcCount PlainCount                         Config
## 1  11.050036       v v  Fin     3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9*V10
## 2   8.070409       p v  Fin     3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9*V10
## 3   9.506401       p p  Fin     3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9*V10
## 4   8.415740       v p  Fin     3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9*V10
## 5 211.700881       v v Auto     4213      26808                    V1*V2*V3*V4
## 6 363.160775       p v Auto     4213      26808                    V1*V2*V3*V4
## 7 378.688111       p p Auto     4213      26808                    V1*V2*V3*V4
## 8 288.664233       v p Auto     4213      26808                    V1*V2*V3*V4
```

We can already see enough to reproduce the table in the paper, but some formatting would be nicer:

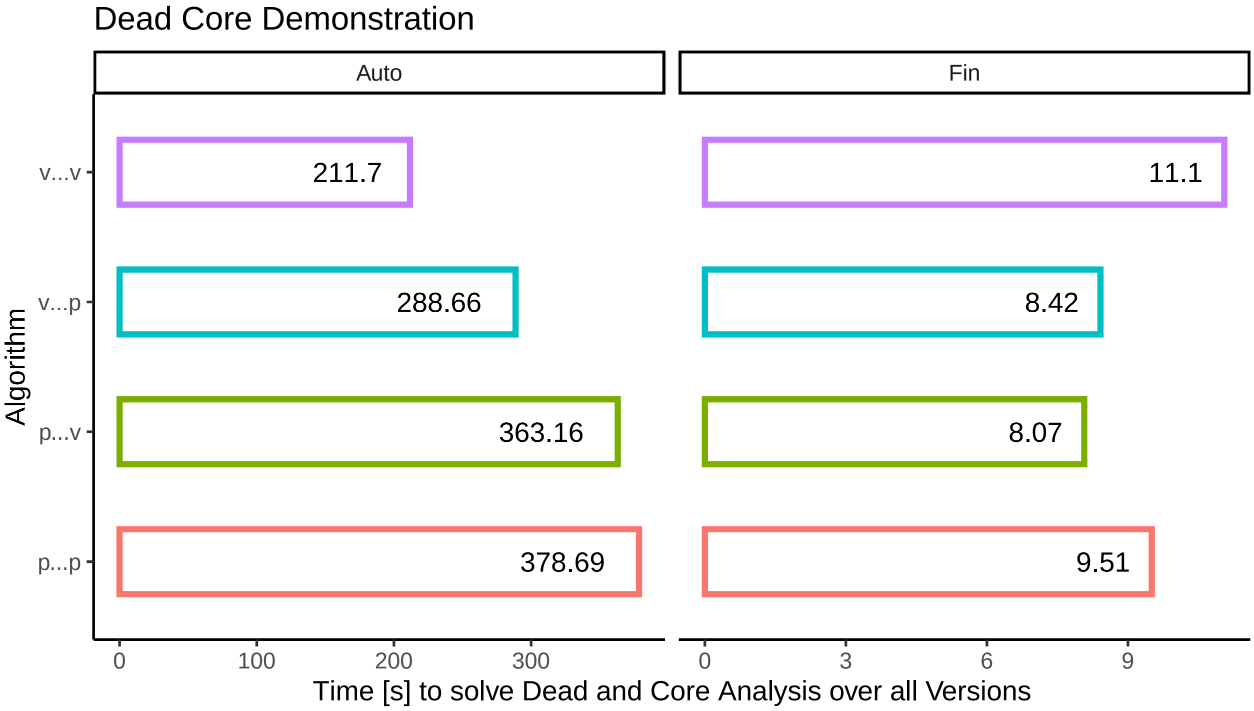
```
deadCoreDF <- data %>%
  # perform the following actions for each dataset
  group_by(data) %>%
  # sort by mean
  arrange(Mean) %>%
  ## change the significant figures by dataset
  mutate(MeanLbl = case_when(data == "Auto" ~ signif(Mean, 5),
                             data == "Fin" ~ signif(Mean, 3)))

deadCoreDF
```

```
## # A tibble: 8 x 7
## # Groups:   data [2]
##     Mean Algorithm data  ChcCount PlainCount Config                      MeanLbl
##    <dbl> <chr>     <chr>    <int>      <int> <fct>                         <dbl>
## 1   8.07 p v       Fin       3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9~    8.07
## 2   8.42 v p       Fin       3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9~    8.42
## 3   9.51 p p       Fin       3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9~    9.51
## 4  11.1  v v       Fin       3810       1441 V1*V2*V3*V4*V5*V6*V7*V8*V9~   11.1
## 5 212.   v v       Auto      4213      26808 V1*V2*V3*V4                  212.
## 6 289.   v p       Auto      4213      26808 V1*V2*V3*V4                  289.
## 7 363.   p v       Auto      4213      26808 V1*V2*V3*V4                  363.
## 8 379.   p p       Auto      4213      26808 V1*V2*V3*V4                  379.
```

and a simple bar plot to show the data:

```
ggplot(deadCoreDF , mapping = aes(x=Algorithm , y=Mean , color = Algorithm)) +
  theme_classic() +
  ## remove the legends for fill, size, color
  guides(fill = FALSE) +
  guides(size = FALSE) +
  guides(color = FALSE) +
  ## make the plot a bar plot
  geom_bar(stat="identity", fill="white", width=0.5
           ,size=1.15) +
  ## add the text labels to show exact values
  geom_text(aes(label=MeanLbl, hjust=1.4), color="black") +
  ## construct the plot for both data sets
  facet_wrap(. ~ data, scales = "free_x") +
  ## other niceties
  labs(title = "Dead Core Demonstration",
       y = "Time [s] to solve Dead and Core Analysis over all Versions") +
  theme(legend.position = "none") +
  coord_flip()
```



Finally we calculate the speedup/slowdown using `dplyr`

```
deadCoreDF %>%
  ## reduce the data frame to relevant columns
  select(data,Algorithm,Mean) %>%
  ## perform the following verbs for each dataset
  group_by(data) %>%
  ## turn long data, wide, i.e., make algorithm labels their own columns with values being Mean
  spread(Algorithm, Mean) %>%
  ## calculuate the speedup ratio with respect to v-->p
  mutate(speedup = `v p` / `v v`)
```

```
## # A tibble: 2 x 6
## # Groups:   data [2]
##   data   `p p`  `p v`  `v p` `v v` speedup
##   <chr>  <dbl>  <dbl>  <dbl> <dbl>   <dbl>
## 1 Auto   379.    363.    289.    212.     1.36
## 2 Fin      9.51   8.07   8.42  11.1   0.762
```