

Best Practices for Scientific Computing

Von Marc Grimpo

Zugrunde liegende Paper

- **Best Practices for Scientific Computing**, Wilson et al.
- **A survey of the Practice of Computational Science**, Prabhu et al.
- **Software Development Environments for Scientific and Engineering Software**, Carver et al.

Gliederung

- Bedeutung der Softwareentwicklung in der Wissenschaft
- Wie entwickeln Wissenschaftler Software?
- Was versteht man unter “Best Practices”
- **Vorstellung der Best Practices**
- Priorisierung
- Fazit

Bedeutung der Softwareentwicklung in der Wissenschaft

- Wissenschaftler verbringen immer mehr Zeit mit der Entwicklung von Software
- Computersimulation als dritte Säule der Wissenschaft neben Theorie & Experiment
 - z. B. Klimamodelle, Kosmologie
- Software für die Verarbeitung und Verwaltung von Versuchsdaten

Wie entwickeln Wissenschaftler Software?

- Softwareentwicklung nicht als Teil der wissenschaftlichen Tätigkeit gesehen
- Wissenschaftler eignen sich Programmierfähigkeiten selbst an
 - Gängige Softwareentwicklungspraktiken oft unbekannt
- Programme kaum getestet
- Der falsche Code wird optimiert

Best Practices

- Was bedeutet der Begriff “Best Practice”?
 - Best mögliche (bereits erprobte) Methode/ Maßnahme zur Durchführung von etwas
 - Für uns: Methoden, die
 - Softwareentwicklung effizienter machen
 - Helfen Fehler zu vermeiden
 - Code wiederverwendbarer zu machen

Write programs for people not computers

- Andere müssen Code nachvollziehen können
 - nur eine handvoll Details zur Zeit
 - Konsistente, eindeutige, aussagenkräftige Benennung von Variablen etc.
 - Konsistente Formatierung & Code-Stil

➔ [Style Guide](#)

Let the computer do the work

- Sich häufig wiederholende Aufgaben erledigen
Computer am besten
 - Kommandozeile benutzen
 - History verwenden / Befehle speichern
 - Skripte schreiben (Shell, Python)
 - Build-Tool verwenden (make, cmake)

Make incremental changes

- Schrittweises Vorgehen
 - Schritte klein halten
 - Häufiges Feedback
 - Pläne anpassen
- Versionskontrolle (Git, SVN)
 - Fehler rückgängig machen
 - Vereinfacht Zusammenarbeit
 - Basis vieler Open Source Projekte

Don't repeat yourself (DRY)

- Alles was wiederholt wird ist schlecht zu verwalten
 - Jeder Datensatz hat einzige gültige Fassung im System
 - Code modularisieren statt Copy-Paste
 - Das Rad nicht neu erfinden
 - Libraries verwenden

Plan for mistakes

- Fehler sind unvermeidbar
 - Assertions verwenden

```
double calculate_weight(struct data* values){
assert(values != Null);

.
. Calculation
.

assert(result >= 0);
return result;
}
```

Plan for mistakes

- Fehler sind unvermeidbar
 - Unit-Tests
 - Bugs zu Test-Fällen machen
 - Debugger verwenden und nicht (nur) printf

Optimize only after it works correctly

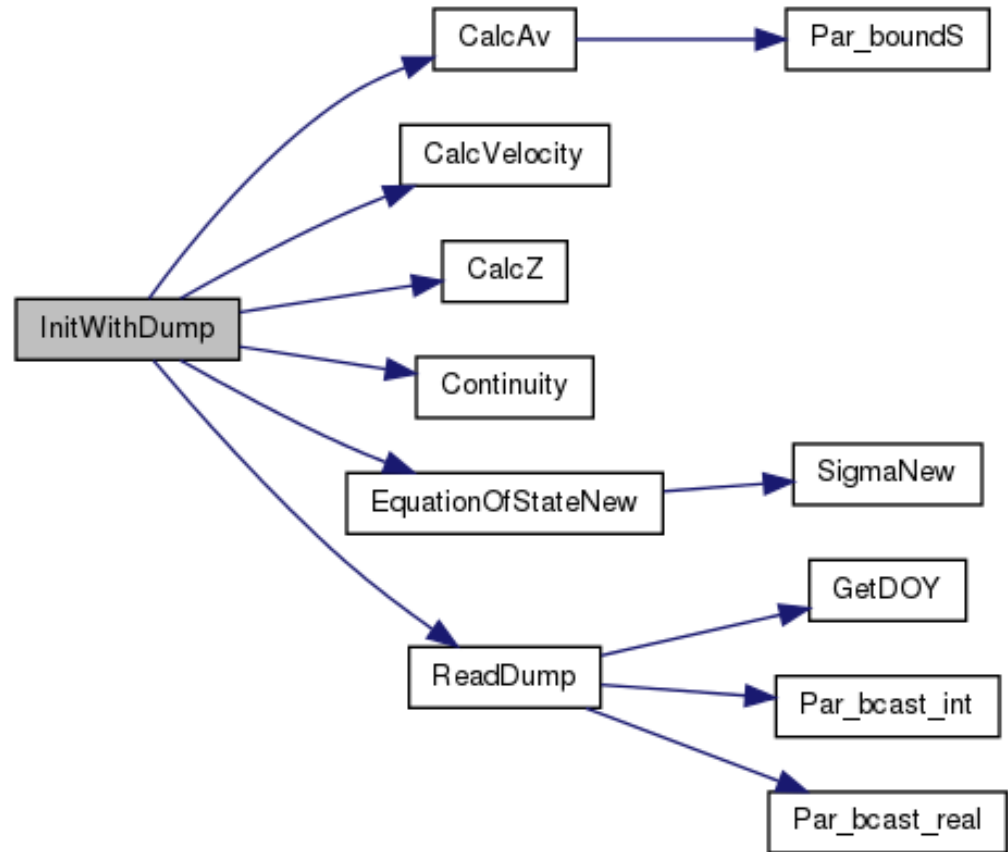
- Wo ist der Flaschenhals ?
 - Oft wird “kalter Code” optimiert
 - Besser: Profiler verwenden (gprof, Vampir)
- Höchst-mögliche Programmiersprache wählen
 - Ist Performance kritisch?
 - Datenmenge

Document design and purpose not mechanics

- Zweck und Auswirkung von Funktionen dokumentieren
- Code nicht im Kommentar rekapitulieren
 - Negativbeispiel: `i = i+1 # Increment i by one`
- Komplexität eliminieren statt erklären
- Dokumentation in Software integrieren
 - [Doxygen](#), Sphinx, Javadoc

Document design and purpose not mechanics

Beispiel für Verzweigung von Subroutinen in Oxygen



Collaborate

- Vier-Augen-Prinzip
 - Code reviews
 - Pair Programming
- Issue-Tracking Tools
 - Trac, [Github](#)

Priorisierung

- Wenig Aufwand:
 - Pair Programming / Code Reviews
 - Style Guide folgen
 - Schrittweise vorgehen
- Mittlerer Aufwand:
 - Doxygen u.Ä.
 - Unit-Tests / Assertions
 - Build-Management-Tools (make)
- Größerer Aufwand:
 - Versionsverwaltung
 - Issue-Tracker

Ergebnisse

- Best Practices verbessern Produktivität
 - Einarbeitungszeit dadurch ausgeglichen
- Am Anfang etwas überwältigend
 - Practices ergänzen sich aber gegenseitig
- Liste ist nicht erschöpfend

Vielen Dank für eure Aufmerksamkeit

Quellen

- **Best Practices for Scientific Computing**, Wilson et al.
- **A survey of the Practice of Computational Science**, Prabhu et al.
- **Software Development Environments for Scientific and Engineering Software**, Carver et al.

Browse Issues Milestones

New Issue

Everyone's Issues 680

680 Open 14,374 Closed Sort: Newest

1 2 3 ... 23

Created by you 0

Mentioning you 0

No milestone selected

Labels

- 4-0-stable 8
- Important 1
- JRuby 5
- PostgreSQL 37
- actionmailer 5
- actionpack 87
- actionview 38
- activemodel 13
- activerecord 275
- activesupport 58
- asset pipeline 4
- attached PR 41
- bug 1

- count breaks with select multiple columns** #15066
Opened by tak2siva 41 minutes ago
- count breaks with select multiple columns** #15065
Opened by tak2siva 42 minutes ago 1 comment
- Calling number_to_delimited on a ActiveSupport::SafeBuffer results in mangled output** #15064
Opened by mjtko 3 hours ago
- Add raise to around callbacks** #15062
Opened by eileencodes 13 hours ago 2 comments
- WIP: Remove unnecessary include for integration tests.** #15061
Opened by tgxworld 14 hours ago 1 comment
- Make ActiveSupport::Configurable enabled classes serializable** #15055
Opened by leafac 2 days ago 8 comments
- ActiveRecord: Wrong primary key value when config.active_record.primary_key_prefix_type = :table_name_with_underscore is set** #15051
Opened by raeno 2 days ago
Labels: activerecord PostgreSQL
- Simplify Finding Default Layout** #15050
Opened by apotonick 2 days ago 8 comments