KNIME for the Masses: Flexible & Scalable Deployment of KNIME-Workflows

Nils Weskamp
Overview

• **Background & Motivation**

• **Issues and challenges**
  • Technical compatibility: Web Service ≠ Web Service
  • Software licenses and access control
  • Robust and scalable deployment of KNIME-workflows

• **Results and current status**
  • Frontend-integration examples
  • Usage statistics

• **Summary and discussion**
Motivation – Background

“MedChem World”

• Windows-based setup
• Large number of end users
• Well-defined installation
• Small number of relatively mature software packages

• Corporate IT: 😊

“CompChem World”

➢ Increasing need to make scientific calculation engines directly available to end users

• Linux-based setup
• Small number of power users
• High number of software packages, often from academic groups or small vendors

• Corporate IT: 😞
Motivation – Status quo

“MedChem World”

Frontend X ➔ Frontend Y

“CompChem World”

Tool X ➔ Tool Y ➔ Tool Z ➔ Tool C ➔ Tool A ➔ Tool B ➔ Tool …

- Various, isolated integrations of some calculation engines into some frontends
- Inconsistencies across frontends
- Need to setup and maintain various related calculation engines in parallel
- Need to use a certain frontend to access a given calculation
Motivation – Brave new world

“MedChem World”

• All calculation engines are available to all end users in relevant frontends
• Opportunities for service consolidation based on science, not technology
• Consistency of calculation results across frontends
Overview

- Background & Motivation

- **Issues and challenges**
  - Technical compatibility: Web Service ≠ Web Service
  - Software licenses and access control
  - Robust and scalable deployment of KNIME-workflows

- Results and current status
  - Frontend-integration examples
  - Usage statistics

- Summary and discussion
Motivation – Brave new world

“MedChem World”

“CompChem World”

Web Services!

- All calculation engines are available to all end users in relevant frontends
- Opportunities for service consolidation based on science, not technology
- Consistency of calculation results across frontends
Web Services – Reality check

Tool A

Web Services

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Tool B

Web Services

 пряма واتّجه بولندا، كان كل ما تعد إجلاء وفنلندا دون كثرة والروسية و، هذا عل مساعدة وفرنسا اوروبا، في إحكام مكتفة والديون لأن، غزّة، الأرضية جعل بل، بل لم مشرط ومحاولة. و بعض الإنزال الإيطالية، عل وقد الخاطفة ويكبيديا، السفن العسكري وبولندا مكن أن. وإقامة

Tool C

Web Services

 مقاومة واتّجه بولندا، كان كل ما تعد إجلاء وفنلندا دون كثرة والروسية و، هذا عل مساعدة وفرنسا اوروبا، في إحكام مكتفة والديون لأن، غزّة، الأرضية جعل بل، بل لم مشرط ومحاولة. و بعض الإنزال الإيطالية، عل وقد الخاطفة ويكبيديا، السفن العسكري وبولندا مكن أن. وإقامة

CCFW
KNIME – The magic glue to integrate tools and applications

- KNIME makes it very simple to combine nodes from various sources and contributors
  - Significant improvements over the last year: no more explicit format conversions / type casts necessary

- Many node collections (from commercial vendors) require a software license
  - Each license comes with its own terms, conditions and restrictions
  - Users, Tokens, Sites, Nodes, Servers, Clients, Copies, Installations, Intended Use, Cloud, Principal Place of Business as Registered, Annual, Perpetual, Increment ...

- Not trivial to decide whether a given person in a global organization has the right to execute the KNIME workflow
  - To make things worse, also internal rules for data access have to be considered
• Deployment of KNIME-workflows across a global Research IT landscape poses a significant challenge
  • 500+ end users benefit from a deployment, but are also affected by a service outage
  • Unpredictable usage / load patterns
  • Robust error handling essential

• The deployment mechanism has to
  • be adaptable to variable load / usage (dynamic scalability)
  • reattempt failed calculations up to n times
  • pre-load / cache workflows to ensure responsiveness
  • restart KNIME instances regularly

• Decision was made to implement a customized deployment mechanism
• Conductor ensures a given number (50-100) of KNIME worker instances is active at all times
• Workers terminate after a given period of time to avoid Java-specific memory / resource issues
• Jobs are placed in the cluster by a cluster queueing system (SGE)
Requests are placed by the CCFW in a request spooling mechanism.

- KNIME worker instances monitor spooler and retrieve requests (modified KNIME batch executor).
- Surplus requests are stored until workers become available.
- Up to n attempts to process a request in case of failure / long processing time.
• Workflow templates available for typical input / output types; allowing users to focus on content
Service publishing / registration can be done within minutes for typical input / output types.
Overview

• Background & Motivation

• Issues and challenges
  • Technical compatibility: Web Service ≠ Web Service
  • Software licenses and access control
  • Robust and scalable deployment of KNIME-workflows

• Results and current status
  • Frontend-integration examples
  • Usage statistics

• Summary and discussion
CCFW – Usage Statistics

- 1.5M+ requests processed in 2014 with a very high usage / load variability over time.
CCFW – Integration into MarvinSketch
CCFW - Integration into D360
CCFW – Integration into KNIME and Pipeline Pilot
Summary and Discussion

• Calculated parameters & predictive models contribute significantly to drug discovery research

  • Some hurdles to adoption are difficult to address and are likely to persist:
    • “Black-box” models with limited interpretability
    • Skepticism concerning the “unphysical” character of models
    • Applicability domain / error-bar estimations

• Other issues of practical relevance are much easier to address and might increase end-user acceptance:
  • Convenient access to calculations and models from relevant frontends
  • Consistent use of the same engines across a large organisation
  • Response times and throughput
Summary and Discussion (II)

• Large-scale deployment of KNIME-workflows to various tools and heterogeneous user groups is possible
  • Some challenges and issues had to be resolved
    • Technical incompatibilities of different tools and platforms
    • Licensing & access control
    • Robust and scalable deployment

• Development opportunities for KNIME for this application field:
  • Performance tuning of workflows; identification of major bottlenecks
  • Debugging of workflows; linking log file entries to individual nodes
  • “Workflow refactoring”
Acknowledgements

- **Boehringer Ingelheim**
  - Bernd Beck
  - Jörg Bentzien
  - Andreas Bergner
  - Gerald Birringer
  - Robert Happel
  - Oliver Krämer
  - Araz Jakalian
  - Jan Kriegl
  - Johannes Koppe
  - Ingo Mügge
  - Prasenjit Mukherjee
  - Edith Richter
  - Matthias Röhm
  - Andreas Teckentrup
  - Nils Weskamp
  - Matthias Zentgraf

- **Torry Harris Business Solutions**
  - Janez Banic
  - Ajay Kannan
  - Shree Lakshmi
  - Maya Madhusudan
  - Shubha Sridhar
  - Rishu Srivastava

- **KNIME**
  - Bernd Wiswedel