

DAQ-Middleware 産学共同開発

株式会社 Bee Beans
Technologies
和田 正樹



内容

- 自己紹介
- Bee Beans Technologiesでの業務
- 共同研究について
- 最近の会社のお話
 - 学生バイト導入
 - GitLab&Jenkinsの導入

自己紹介

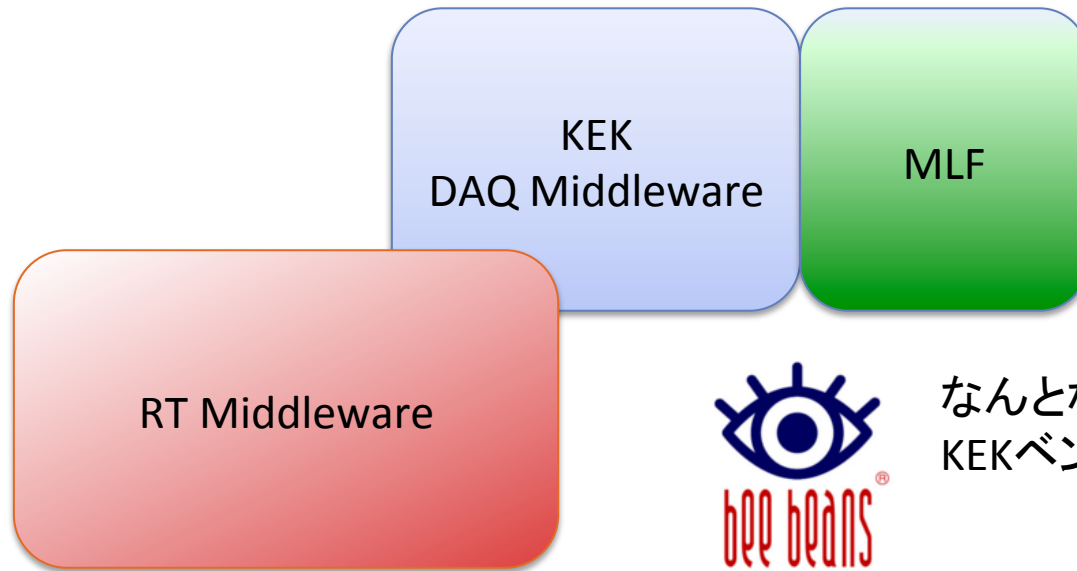
- 学部は理学部、学科は応用科学でした。
- 大学卒業後、大手の電子部品メーカーに1年半。
- コンピュータソフトウェアの会社に転職してから、真面目にコンピュータのスタディを開始(24歳)
- 最初の仕事は主にNECさんの伝送装置用ファームウェア(iTRON)やネットワークマネージャ(Solaris)の受託開発。要求仕様の検討で1年、開発は数年かかるような大規模開発に関わりました。
- 「自社製品を作りたいかった」ので学校の先輩の作った会社に転職。通信測定器の開発に携わる。非同期通信モード(ATM)やIPテストの開発。WindowsアプリケーションとLinuxを使用した測定器開発。
- 2009年からBee Beans Technologiesの役員となりました。当時は社長と社員2人の会社でした。

長らく、B to Bのお仕事でした

Bee Beans Technologiesでの業務

- DAQ M/Wの共同研究
- 東大地震研: Muon Radiography (Webアプリケーション: PHP)
- KEK MPGD関連のソフトウェア開発 (Linux C++, Java)
- KEK 中性子検出器用の検査ソフトウェア開発 (Linux C++, Qt SDK)
- J-PARC/MLF関連のサポート業務
 - DAQ M/Wコンポーネントの開発、サポート
 - サーバー設置、設定、運用サポートなど
- J-PARC/MLF用解析アプリケーション、制御アプリケーションの開発 (C++, Qt SDK, Python)

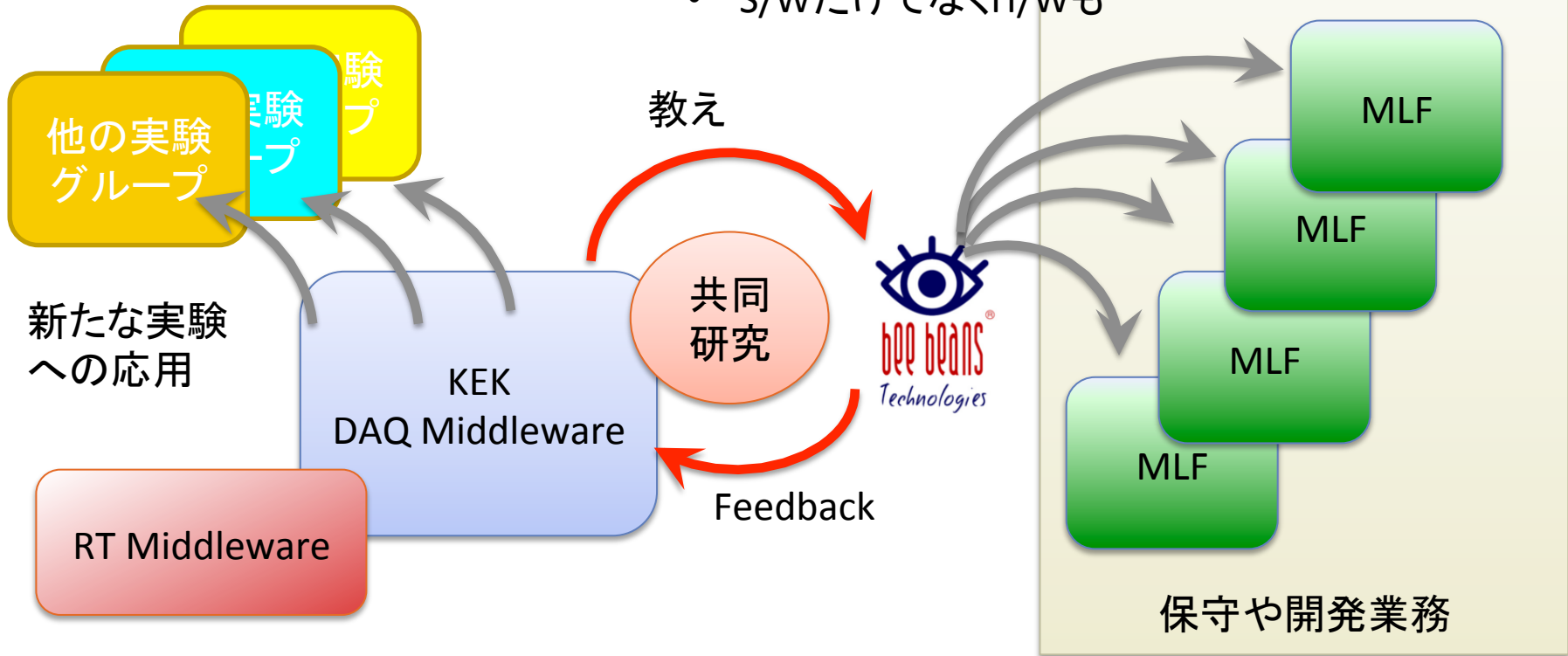
共同研究(最初の頃)



なんとなく入ってみる
KEKベンチャーだし

共同研究(現在)

- ビームラインの横展開盛ん
- 必要なコンポーネントの開発
- S/WだけでなくH/Wも



新たな実験
への応用

KEK
DAQ Middleware

RT Middleware

共同
研究

教え

Feedback



MLF

MLF

MLF

MLF

保守や開発業務

共同研究では

- MLFでの要望や問題のフィードバック
 - 教えを請う、いろいろ議論させていただく。
- 企業的な視点から意見をいってみる
 - 企業的な視点？
 - DAQ M/Wの話だけでなく、余談的な話もFeedbackします。

企業的な視点

- 品質を安定させたい
 - サポートコストを下げたい
- 拡張性の確保したい
 - 先々の開発コストも下げたい
- コードの保守性の向上
 - 開発、サポートコスト下げたい

- 企業的勘違い: テレビのチャンネルに例えると
 - 見えないチャンネルがあると大騒ぎ(企業の視点)
 - 見たくないチャンネルは見えなくていいです(研究者の視点)

※研究内容に依存すると思いますが、、

最近の会社のお話

人手不足を安く補いたい

会社の将来を担う若手不足

学生バイトでも使ってみるか？
筑波大学も近いし。。

学生バイト来ました！

- ハードウェアが2人とソフトウェアが3人
- それぞれ個性的ですが、
 - 今回はソフトウェア
 - おとなしいAくん
 - チャラチャラしたBくん
 - まじめなCくん

Aくん

- 情報通信学類の3年生(現在4年)です
- 留年しちゃって暇
- あまりお勉強していなかった模様
- アルバイトも弊社が初めてだそうです。

同じ境遇(筑波大)でやはり無口な若手社員がうまく使い始めました。

思ったよりできなかったけれど、思ったより成長しています

Bくん

- すごいチャラチャラしています
- Webアプリケーション開発が得意
- 「(社内コミュニケーションに)Skypeを使っているとところ**だけ**最新」と言い切られました。
- 「Windows無理なんで」とか。
- ソーシャルコーディング、ペアプログラミングなどが当たり前と思っている

今風のデジタルネイティブで情報も早く、良い刺激になります

Cくん

- 現在4年生で大学院進学希望
- 高専からの編入で真面目な雰囲気
(周囲に女子が増えたので「はじけたい」そうです)
- 高専のときからVisual Studioなどを使って開発しており、C++などのトラッドな開発が得意
- わりと会社の業務にあっている

若いのに頑張っているなあーと。※刺激は少なめ

学生バイト余談

- 半端な人というか、普通な人は応募してこない
 - 学生課に行った時に耳にしたという学生さんの会話「**頭使う仕事は嫌だよねー**」
- 実力者は奨学金で勉強、生活費はバイトで稼ぐというタイプの人が多い。>即戦力

学生バイトさんの成果たち

- KEK Manyo Library(解析ライブラリ)を使った試作、テスト、インストーラの作成のお手伝いなど
- SiTCP周辺ツールのマルチプラットフォーム対応
＞テスト中です
- SiTCP用ライブラリ(Python、C++など)およびテスト環境
- デバイスログビューア(Webアプリケーション)
- 社内CMSの高度化(カレンダーやタイムカードGoogleカレンダー風に)

- 質的には素人っぽいところがあります。
- 一人は4月から社員になるので、成果は上々。

ところで「Skype**だけ**最新？」

- Skype**だけ**最新というのもちょっと、、、乗り遅れはソフトウェア開発事業の命取りになる
 - ※社内コミュニケーションにLINEは使わないですが、、、
- 学生さんは卒業していなくなるので、成果の引き継ぎを速やかにしたい。
- 流行りモノの開発環境(github系など)に慣れておく必要がある。
 - ※ソーシャルコーディングやペアプログラミングなどはまだ勇気も予算もありません。
- そこで、進めたのはgitlabとJenkins
 - 産総研さんのようなソフトウェアの専門的な研究機関では随分前から導入されているようですが、、実験制御ソフトウェアや解析プログラムなどでは開発・保守環境は後回し的です。

GitLab & Jenkinsの導入

- GitLabはWikiやIssue管理のついたりリポジトリサーバー。GitHubのイントラネット用Cloneという位置づけ
 - もともとSubVersionを使用していたものの、ネットの情報をみるとgithubに乗りかえたほうが良さそう。
 - ただし、仕事の成果物はライセンス問題もあるのでGitHubは難しい
- Jenkinsでビルドやテスト環境の更新を一元管理。
 - Web UIのビルドボタン一発で成果物ができてテスト環境の更新も可能
 - 複数のOSに対応した製品の各プラットフォーム用ビルドも簡単。Linux, Windows, Mac OSのスレーブでビルドしてくれます

GitLabの導入

- SubVersion未登録で進行中のタスクを練習がてらプロジェクト登録してもらいました。
- 2ヶ月でなんと17プロジェクトになりました
 - 関わった開発者は6人、、
 - SubVersionからの移行もまだなのに。。。
 - ※残念ながら案件数ではありません。
- リポジトリの作成をだれでも気軽にできるようになったメリットは大きいです。
- SubVersionと比較した良さはまだ少なく、これから積極活用するといろいろメリットが体感できると思います。
- 時間を取られたのは、LDAP連携とSSH_KEY関係で少々トラブルがありました

Jenkinsの導入

- 順調に浸透しまししています
- 苦労したところはスレーブマシンのセットアップとシェルスクリプトの癖など。
- gitlabの導入と同時に試用開始したので、それもハマりどころでした。(ssh key周辺)

ジョブの種類

#	ビルド実行状態	ジョブ名	実行状況	実行時間	開始日時	終了日時	実行時間
1	待機中	master	●				
2	待機中	BBTWS	●		1ヶ月前 (#1)	1ヶ月前 (#1)	2.7秒
1	待機中	BRoad-MAC-OS-X	●	6時間 17分前 (#3)		3日 6時間前 (#1)	10秒
1	待機中	BRoad-SL53	●	5時間 46分前 (#3)			1分 9秒
1	待機中	BRoad-SL64	●	6時間 54分前 (#3)		6時間 57分前 (#2)	0.56秒
1	待機中	BRoad-Ubuntu	●	6時間 2分前 (#3)		3日 5時間前 (#1)	3秒
1	待機中	BRoad-win7	●	6時間 13分前 (#9)		1ヶ月 10日前 (#4)	4.5秒
1	待機中	Logvisualizer test	●	16日前 (#110)		16日前 (#108)	3分 19秒
1	待機中	Manyo deb with boost1.48.0	●	1ヶ月 2日前 (#22)		1ヶ月 8日前 (#11)	18分
1	待機中	Manyo win build	●	11日前 (#62)		11日前 (#61)	7分 59秒
1	待機中	many win install	●	11日前 (#30)		11日前 (#30)	1.4秒
2	待機中	mif_test	●	8日前 (#3)		8日前 (#3)	0.1秒
1	待機中	neudagpapp-win7	●	1ヶ月 10日前 (#19)		1ヶ月 10日前 (#15)	6.1秒
1	待機中	nGEM-app-win32	●	1日 16時間前 (#34)		1ヶ月 3日前 (#32)	57秒
2	待機中	nGEM-SL53	●	1日 16時間前 (#5)		12日前 (#2)	37秒
1	待機中	nGEM-SL64	●	1日 16時間前 (#21)		2ヶ月 1日前 (#12)	53秒
1	待機中	nGEM-SL64-copy	●	1日 16時間前 (#27)		1ヶ月 2日前 (#6)	22秒
2	待機中	nGEM-Ubuntu12.04	●	7日 3時間前 (#30)		7日 3時間前 (#23)	11秒
1	待機中	nGEM-Win32	●	1ヶ月 5日前 (#66)		1ヶ月 5日前 (#65)	11秒
1	待機中	nGEM-win32-deploy	●	1日 16時間前 (#18)			17秒
2	待機中	nGEMapp-MAC-OSX	●	4日 22時間前 (#10)		4日 22時間前 (#9)	9.1秒
1	待機中	nGEMTools-Mac-OSX	●	4日 7時間前 (#7)			13秒
1	待機中	nGEMTools-win7	●	4日 7時間前 (#15)			14秒
1	待機中	SITCUtility-MAC-OSX	●	2日 23時間前 (#38)		25日前 (#28)	9.5秒
1	待機中	SITCUtility-SL53	●	2日 23時間前 (#28)		2日 23時間前 (#27)	16秒
1	待機中	SITCUtility-SL64	●	5時間 40分前 (#6)			2秒
1	待機中	SITCUtility-Ubuntu	●	2日 22時間前 (#11)		2日 22時間前 (#10)	2秒

● 2013年9月2日現在で41個のジョブがあります。

● プロジェクトの性格がいろいろ、Qt、g++、Cygwin、Python、Php(Web)

● 実際には、未対応の成果物がまだまだあります。

スレーブノードの数と種類

名前	アーキテクチャ	クローン数	応答時間	メモリ使用量	ディスク容量	ディスク使用量
master	Linux (amd64)	0ms	5205MB	19GB	1802GB	
bbtwin01	Windows XP (x86)	2.2秒 遅んでいます	2ms	3297MB	104GB	110GB
bbtwin01	Windows XP (amd64)	同期中	8ms	12599MB	183GB	183GB
bbtwin01	Linux (i386)	N/A	N/A	N/A	N/A	N/A
bbtworks1	Linux (i386)	同期中	3ms	3999MB	70GB	70GB
bbtworks1	Linux (i386)	同期中	13ms	1023MB	6GB	6GB
bbtworks1	Windows XP (x86)	同期中	1ms	3042MB	217GB	217GB
IMAC	Mac OS X (x86_64)	同期中	1ms	676MB	875GB	875GB
KVM-Ubuntu1204(155)	KVM-Ubuntu1204(155)	同期中	4ms	4079MB	32GB	90GB
manywin01 (オフライン)	Mac OS X (x86_64)	同期中	1ms	423GB	423GB	423GB
many-build	Windows 7 (amd64)	1秒 遅んでいます	16ms	6385MB	28GB	28GB
many-install	Windows 7 (x86)	同期中	2ms	6873MB	6GB	6GB
manywin01	Linux (amd64)	9分5秒 遅れています	2ms	3967MB	26GB	26GB
manywin01	Linux (amd64)	同期中	4ms	3885MB	9GB	9GB
IMAC	SL55(64)-DAQInstaller	同期中	1010ms	5951MB	62GB	62GB
KVM-Ubuntu1204(155)	SL64(64)-Desktop	同期中	3ms	3839MB	1721GB	1721GB
macmini	SL64_x86_64	同期中	6ms	3982MB	2GB	2GB
many-build	TOKAI-hrc-cpu2	同期中	21ms	7999MB	1550GB	1550GB
many-install	TOKAI-hrc-cpu3	同期中	22ms	7999MB	437GB	437GB
manywin01	TOKAI-nova-ana1	同期中	103ms	5407MB	581GB	581GB
many-rpm	TOKAI-nova-dac-20	同期中	349ms	7953MB	32GB	32GB
manywin01	TOKAI-nova-dac-45	同期中	284ms	8045MB	28GB	28GB
manywin01	TOKAI-nova-dac-90	同期中	193ms	3833MB	34GB	34GB
manywin01	TOKAI-nova-dac-bs	同期中	200ms	5951MB	443GB	443GB
manywin01	TOKAI-nova-mon	同期中	430ms	5407MB	616GB	616GB
SL55_64-HRC-LogVisualize	Ubuntu12.04-Desktop	同期中	2ms	4085MB	1728GB	1728GB
SL55(64)-DAQInstaller	VM-Win7	N/A	N/A	N/A	N/A	N/A
SL64(64)-E	Win7	9分5秒 遅れています	3ms	3185MB	100GB	100GB
SL64_x86_64	Win764-real	同期中	3ms	3109MB	100GB	100GB
TOKAI-hrc-cpu2	Windows 7 (amd64)	同期中	2ms	3398MB	28GB	28GB

2013年9月2日時点で30台(仮想マシンを含む)

windows 7 7台
Windows XP 2台
Linux(RHEL + Ubuntu) 19台
Mac OS X 2台

32bit、64bit、RHEL5系、6系、Ubuntuなど組み合わせでLinuxが多い
Windowsも開発環境あり、無しなど

導入後の感想

- 思ったより管理されていないビルドジョブやソースなどがたくさんあったこと
 - 成果物が一箇所になってとても管理しやすくなりました。
 - メンバーへの浸透は早かった(それなりにひな形を作りましたが)。
- 乱立していた仮想マシンを含めて、ノード(計算機)の状況がわかりやすくなりました。ディスクの使用状態なども表示されます。
- 新しいジョブのビルドエラーも共有できて、アドバイスも簡単です
- 開発中のプロダクトでも、簡単に試用/テスト出来て便利です。
- バイトさんの仕事の管理もできている風
 - 仕事の指示も楽になりました
 - Jenkinsにしといてくれる?とか、UbuntuとMACでもビルドできるようにしておいて、とか。

更にすすめたいこと

- 運用やジョブ作成のルール化、ノードやラベルのネーミングルールなど。
- テストの自動化。目先は静的なところから。
- 自動テストのノウハウをつけて、今後の開発に活かす。
- FPGAなども自動化できる模様。
※Linuxのスクリプトで夜間ビルドしているところもあるらしいので

ご紹介ソフトウェア類

- Jenkins、gitlab
 - bitnami Linux用などのWebアプリケーションインストーラが多数あって、簡単に試せます。
bitnami(<http://bitnami.com>)
- Webアプリケーションの自動テスト(selenium)などはDAQ M/Wでも活用できると思います。
 - 簡単なテストパッケージを提供するとconfigファイルを書き換えるだけで、様々なコンポーネントの組み合わせで基本的な動作試験ができるようになります。
<http://docs.seleniumhq.org/>