



# 分散テスト処理システムの開発による ソフトウェア開発効率の向上

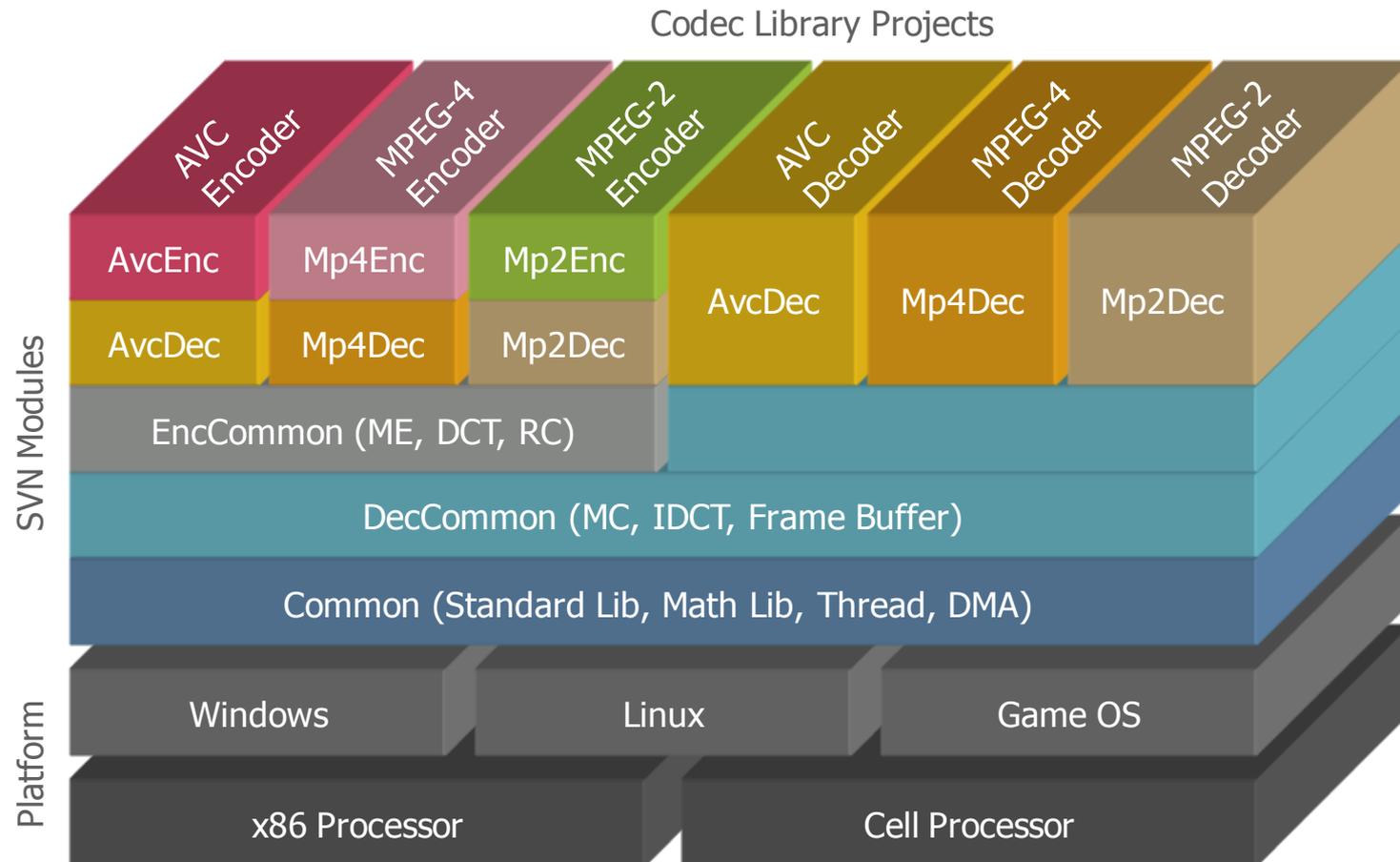
ソニー(株) 技術開発本部  
共通要素技術部門 信号処理技術1部  
梶畠 博

# 自己紹介

- **職業:**
  - ソフトウェア エンジニア
  - プロセス改善の専門家ではありません
- **業務:**
  - ソフトウェア ビデオコーデック ライブラリを開発
    - AVC, MPEG-4, MPEG-2, ...
- **開発言語:**
  - C/C++
- **開発環境:**
  - Windows (Visual Studio), Linux, PS3 Game OS ...
- **構成管理:**
  - Subversion

# マルチコーデック/マルチプラットフォーム開発

## ■ 開発効率向上のため一部ソースコードを共通化



# 開発効率向上のために必要なこと

- **リポジトリ上のコードが常にビルド可能であること**
  - チームで開発をする上でとても重要
  - 頻繁にビルドが通らなかったり、動作があまりに不安定だと、解析/デバッグ作業に時間を取られ、コーディングの生産性が低下する
- **そこで、コードの品質を維持するために、さまざまなテストを用意する**

# ソフトウェア開発における各種テスト

- **開発者テスト**
  - クラス、モジュール単体でテスト
  - 統合テスト、など
- **コミットテスト**
  - リポジトリへのコミット前に必ず行うテスト
  - テストが通らないコードはコミット不可
  - あまりテストに時間がかかり過ぎてもいけない
- **デイリーテスト**
  - 毎晩リポジトリ上の最新コードを使って自動ビルド・テスト
  - コミットテストよりも、より広範囲なテストが可能
- **リリーステスト**
  - リリース前に行う大規模テスト
  - 外部のQAチームとも協力してテスト

# コミットテスト

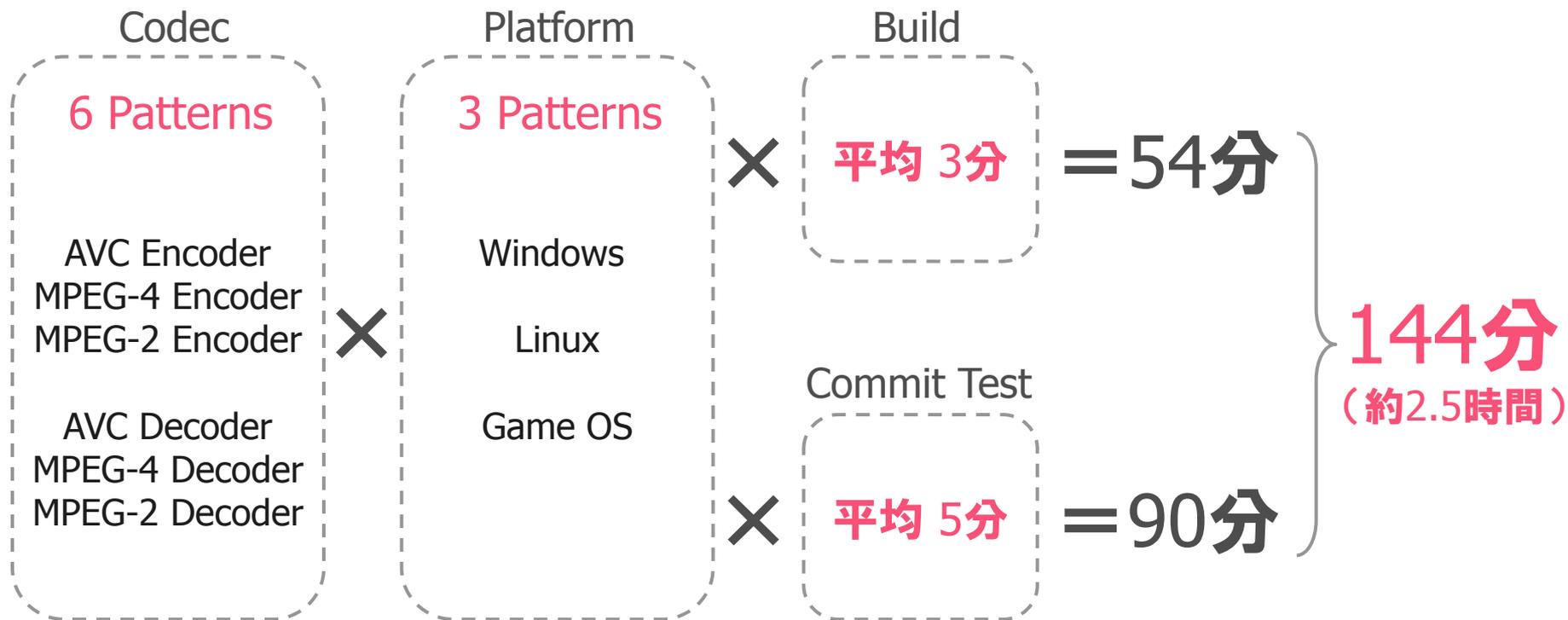
メリット:  
リポジトリ上のコードの  
品質を維持できる

デメリット:  
めんどくさい、  
テスト完了待ちが発生



生産性向上のためには  
ビルド/テストの「**自動化/高速化**」が必要

# コミットテスト: 実行時間



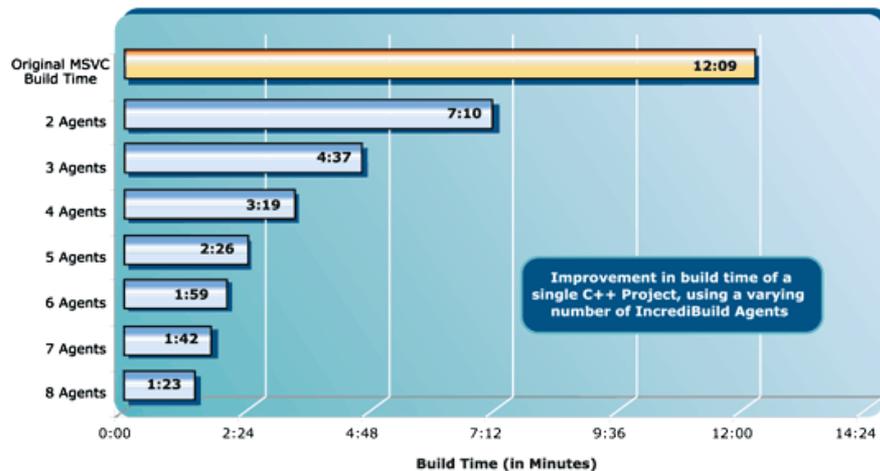
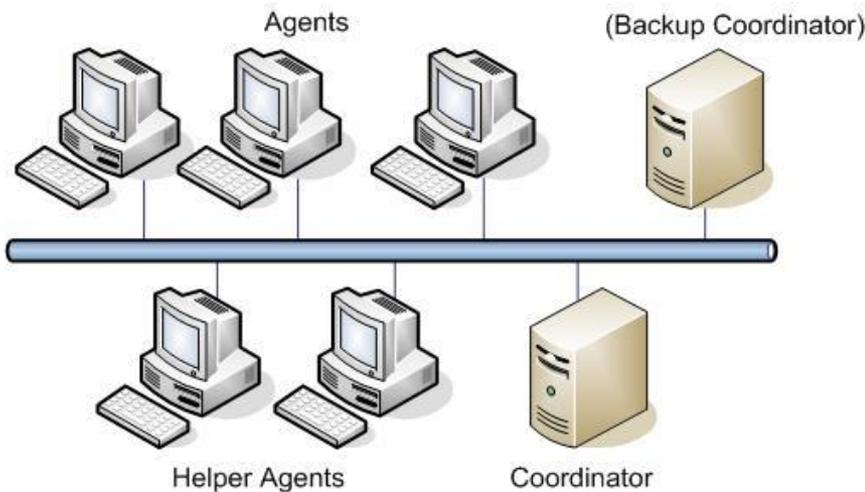
時間がかかりすぎる。。。

# 取り組んだ改善事項

- **ビルドの高速化**
  - IncrediBuild 導入
- **テストの自動化/高速化**
  - 分散テストシステム dTest 開発
- **テストのかんたん実行**
  - Remote Test System の開発
- **コミットルールの変更**

# ビルドの高速化

- Xoreax社 IncrediBuildを導入
  - ネットワーク上の複数のPCで分散処理することでVisualStudioのビルド時間を短縮する



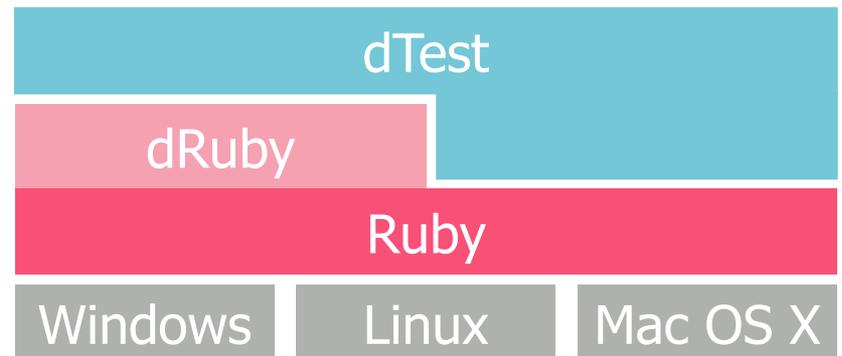
テストも分散処理  
すればいいんじゃない？



分散テストシステム  
dTestの開発へ

# 分散テストシステム dTest とは

- テストスクリプトを複数のPCで分散並列処理
- Webブラウザ上からテストを実行可能
- マルチプラットフォーム対応(混在可能)
- 開発言語: Ruby



# Why Ruby ?



[ウェブ](#) [画像](#) [ニュース](#) [地図](#) [グループ](#) [more »](#)

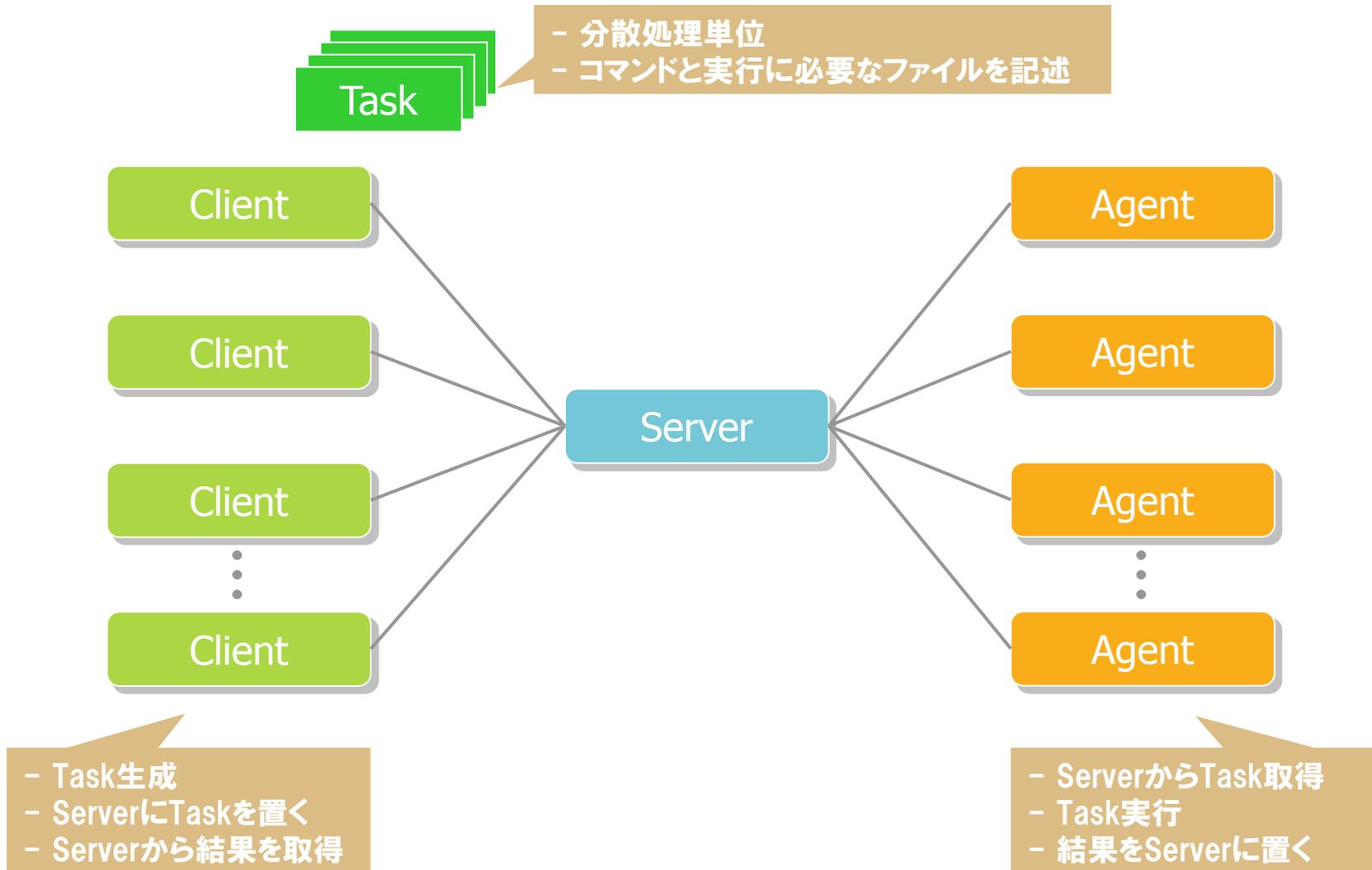
スクリプト 分散 プログラミング

Google 検索

I'm Feeling Lucky

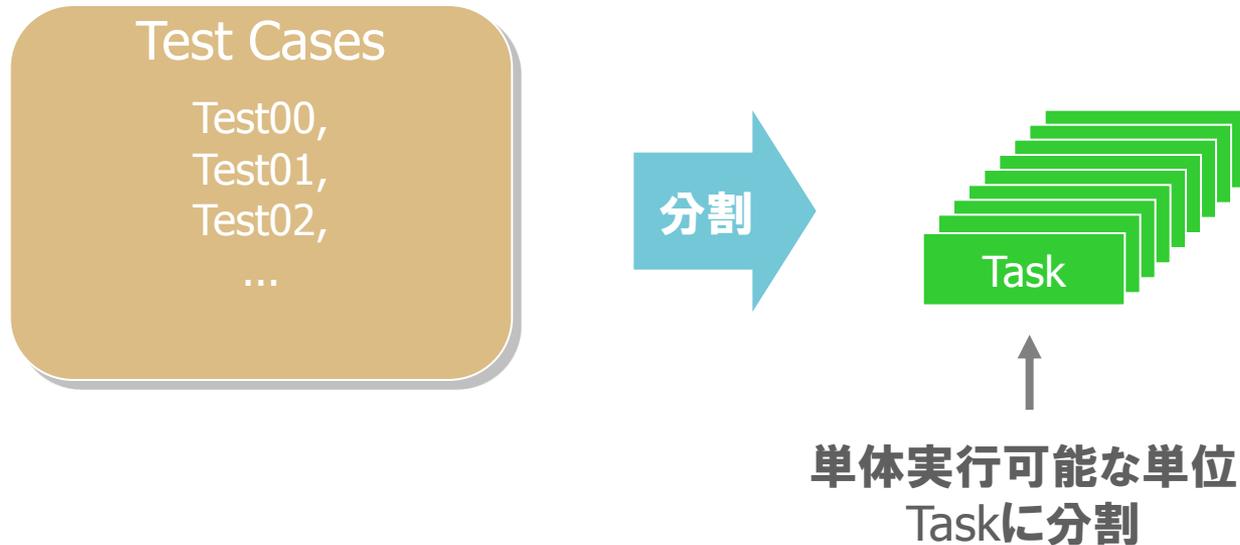


# dTest システム構成



# 分散並列処理するために

- スクリプト処理の単位を細かく分割する
  - Taskに分割
  - Task間の依存関係をなくす



# Task

- 分散処理の最少単位
- 実行に必要なファイルリスト、コマンド等で構成
- テストに必要なファイルは自動転送される
  - 各PCに事前にテスト環境を構築する必要がない

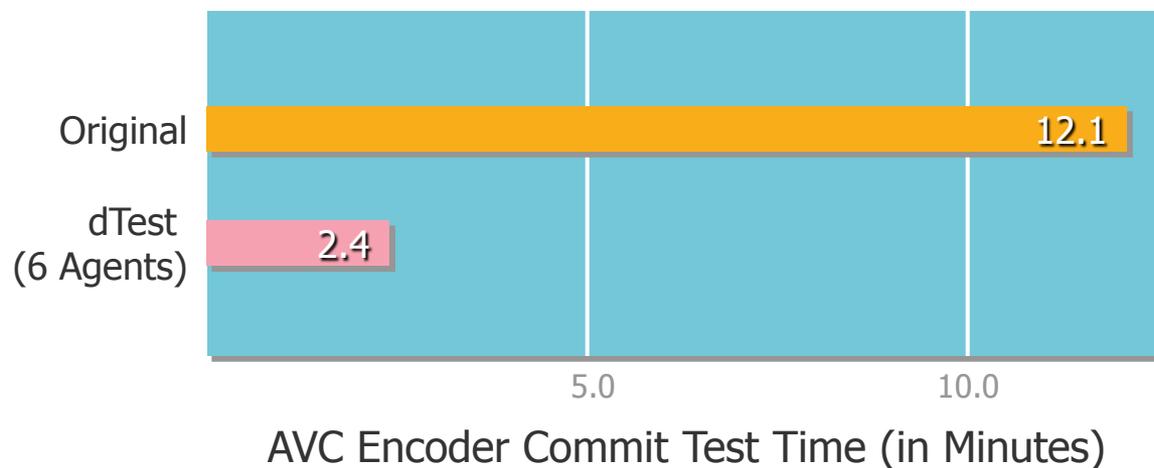


# コーデックライブラリ向け自動テスト

テスト名	テスト内容
match	以前のバージョンと結果が一致するか？
debug	ReleaseビルドとDebugビルドで結果が一致するか？
cpu	CとSIMD(最適化)コードの結果が一致するか？
thread	マルチスレッドとシングルスレッドで結果が一致するか？
profile	以前のバージョンから速度が低下していないか？
snr	以前のバージョンから画質が劣化していないか？
conf	エンコード結果に規格違反はないか？

# AVC Encoderコミットテスト実行時間

## ■ 6台のPCで約5倍 高速化

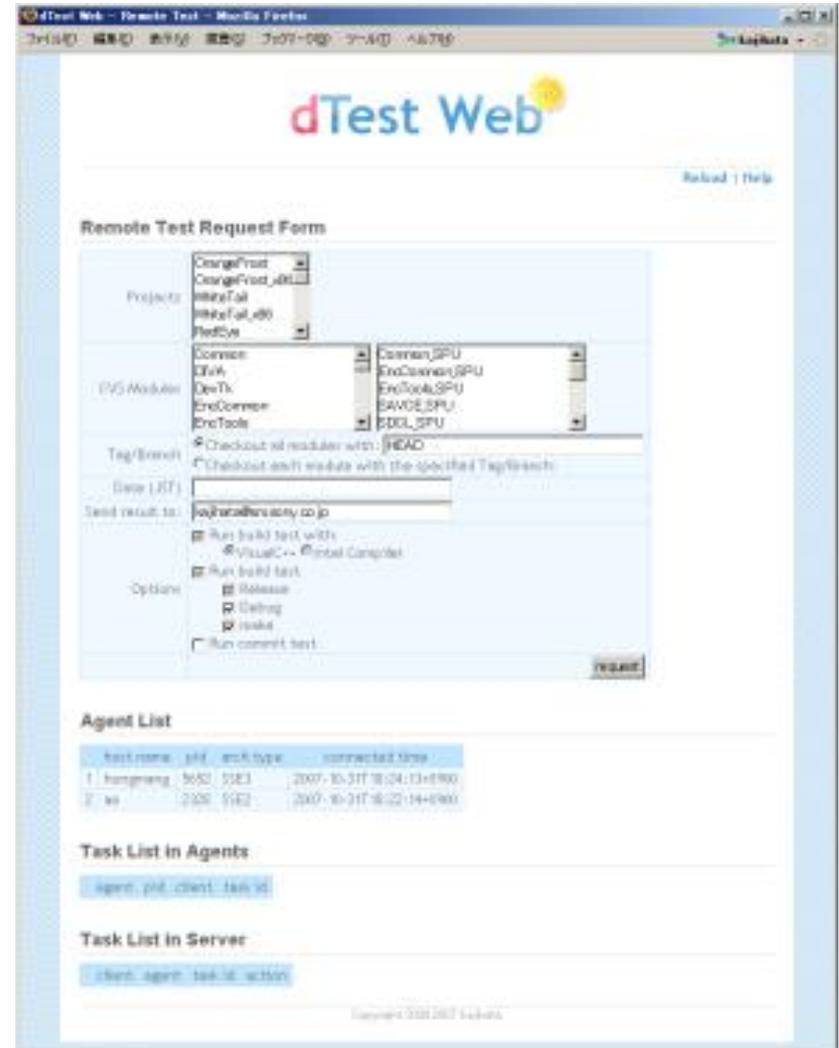


# テストのかんたん実行

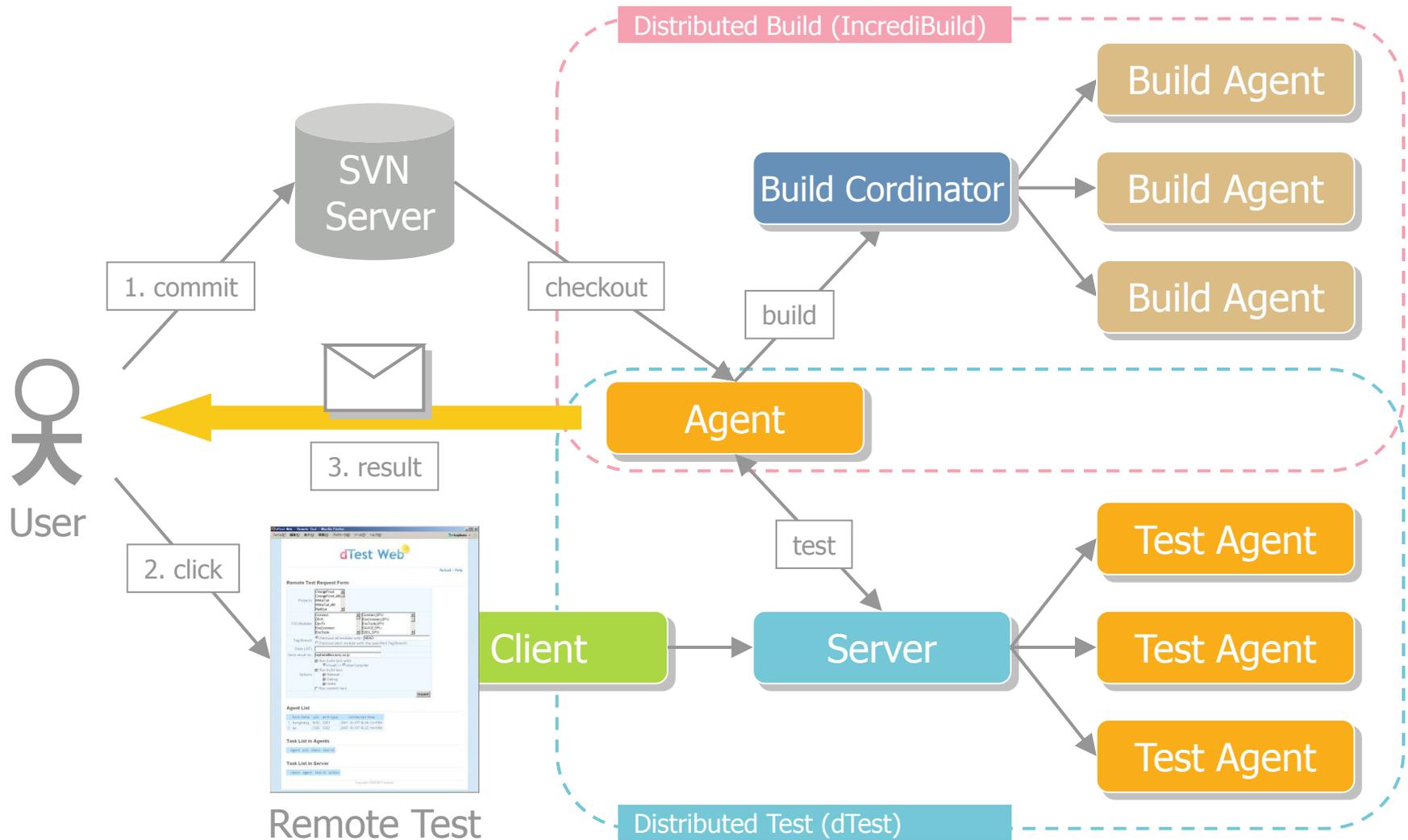
- 誰にでも使えるよう、Webブラウザ上から各種機能を実行可能に
  - Remote Test
  - Server Monitor
  - Test Log Viewer
- Webサーバ/CGIとともにRubyスクリプトで動作
  - 特別なWebサーバのインストールが不要

# Remote Test

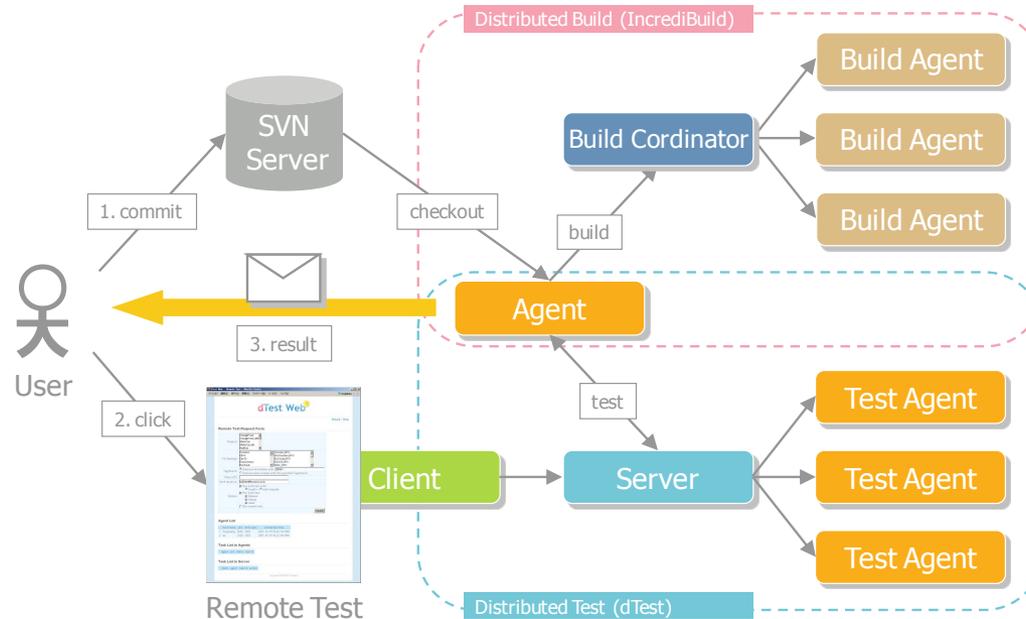
- Webブラウザ上からビルド/テストを自動実行
- 対象プロジェクト、テスト内容を選択可能
- チェックアウトするリビジョン、ブランチを指定可能
- テスト結果はメールでユーザに通知



# Remote Test: システム構成



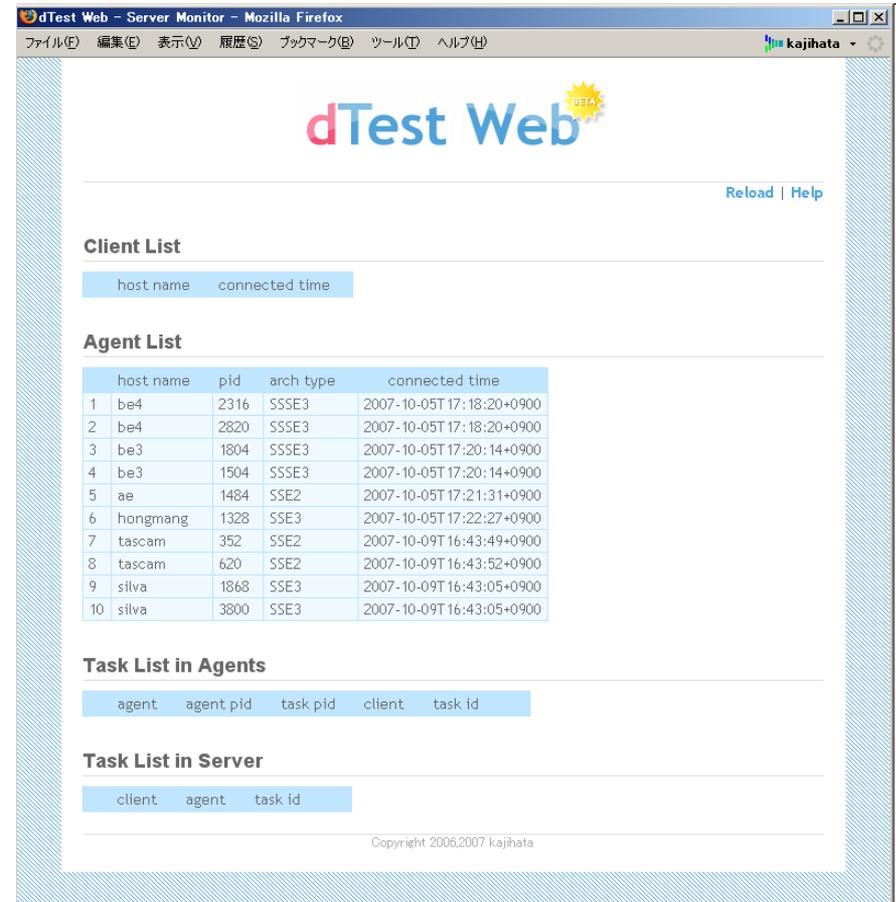
# Remote Test: ユースケース



- ブランチ上にある開発中のコードをトランクにマージする前の一斉テスト
- バグ混入時期特定のための一斉テスト
- コミット不足のファイルがないかの確認
- テストPC/環境を複数人で共有、など

# Server Monitor

- Server, Client, Agentの  
状態を表示
- Taskの実行をキャンセル  
可能
  - Server上にある未実行  
Task
  - Agent上で実行中のTask



dTest Web 

[Reload](#) | [Help](#)

#### Client List

host name	connected time
-----------	----------------

#### Agent List

	host name	pid	arch type	connected time
1	be4	2316	SSSE3	2007-10-05T17:18:20+0900
2	be4	2820	SSSE3	2007-10-05T17:18:20+0900
3	be3	1804	SSSE3	2007-10-05T17:20:14+0900
4	be3	1504	SSSE3	2007-10-05T17:20:14+0900
5	ae	1484	SSE2	2007-10-05T17:21:31+0900
6	hongmang	1328	SSE3	2007-10-05T17:22:27+0900
7	tascam	352	SSE2	2007-10-09T16:43:49+0900
8	tascam	620	SSE2	2007-10-09T16:43:52+0900
9	silva	1868	SSE3	2007-10-09T16:43:05+0900
10	silva	3800	SSE3	2007-10-09T16:43:05+0900

#### Task List in Agents

agent	agent pid	task pid	client	task id
-------	-----------	----------	--------	---------

#### Task List in Server

client	agent	task id
--------	-------	---------

Copyright 2006,2007 kajihata

# Test Log Viewer

- 分散テストの結果を一覧表示
- テストの実行状況も表示



The screenshot shows a web browser window titled "dTest Web - Test Log Viewer - Mozilla Firefox". The page displays the "dTest Web" logo and a "Last Test Results" section. Below this section is a table with columns for project, test, result, date, agent, and misc. The table lists various test runs for projects like OrangeFrost ( avc ), OrangeFrost ( avc\_avchat ), and OrangeFrost ( avc\_msvideo\_bL1 ), with results ranging from OK to NG (Not Good).

project	test	result	date	agent	misc
OrangeFrost ( avc )	Match	OK	2007/11/13 02:57:01	BE0	
OrangeFrost ( avc )	GccBuild	OK	2007/11/13 00:12:28	HONGMANG	
OrangeFrost ( avc )	Cell	OK	2007/11/13 06:49:19	ARAQ-WIN2	
OrangeFrost ( avc )	GccBuild	OK	2007/11/13 05:16:57	ARAQ-WIN2	
OrangeFrost ( avc )	GccBuild	OK	2007/11/13 02:13:32	HONGMANG	
OrangeFrost ( avc )	MemorySize	OK	2007/11/13 02:17:46	HONGMANG	
OrangeFrost ( avc )	Purify	NG	2007/10/14 16:50:56	HONGMANG	log(NG)
OrangeFrost ( avc )	CPU	OK	2007/11/13 09:27:47	TASCAM	
OrangeFrost ( avc )	DebRel	OK	2007/11/13 08:26:32	SEED	
OrangeFrost ( avc )	Thread	OK	2007/11/13 08:23:48	BE0	
OrangeFrost ( avc )	WorkBuf	OK	2007/11/13 05:08:08	BE0	
OrangeFrost ( avc )	Check	OK	2007/11/13 02:50:55	BE0	
OrangeFrost ( avc )	Conf	OK	2007/11/13 02:39:22	BE0	
OrangeFrost ( avc )	ErrParam	OK	2007/11/13 00:18:04	BE0	
OrangeFrost ( avc )	Thread	OK	2007/11/13 00:17:54	BE0	
OrangeFrost ( avc_avchat )	Cell	OK	2007/11/13 08:17:10	ARAQ-WIN2	
OrangeFrost ( avc_avchat )	DebRel	OK	2007/11/09 10:50:39	SILVA	
OrangeFrost ( avc_avchat )	Thread	OK	2007/11/09 10:39:49	SILVA	
OrangeFrost ( avc_avchat )	GccBuild	OK	2007/11/13 08:11:46	ARAQ-WIN2	
OrangeFrost ( avc_avchat )	WorkBuf	OK	2007/11/09 10:36:43	SILVA	
OrangeFrost ( avc_avchat )	Match	OK	2007/11/09 10:33:50	SILVA	
OrangeFrost ( avc_avchat )	Check	OK	2007/11/09 10:33:31	SILVA	
OrangeFrost ( avc_avchat )	CPU	OK	2007/11/09 10:29:20	SILVA	
OrangeFrost ( avc_avchat )	Thread	OK	2007/11/09 10:29:23	SILVA	
OrangeFrost ( avc_avchat )	Conf	OK	2007/11/09 10:32:42	SILVA	
OrangeFrost ( avc_msvideo_bL1 )	Conf	OK	2007/11/13 09:35:57	SEED	
OrangeFrost ( avc_msvideo_bL1 )	GccBuild	OK	2007/11/13 07:05:40	ARAQ-WIN2	
OrangeFrost ( avc_msvideo_bL1 )	Check	OK	2007/11/13 09:36:37	SEED	
OrangeFrost ( avc_msvideo_bL1 )	Cell	OK	2007/11/13 07:11:02	ARAQ-WIN2	
OrangeFrost ( avc_msvideo_bL1 )	Match	OK	2007/11/13 09:37:03	SEED	
OrangeFrost ( avc_msvideo_bL2 )	Match	OK	2007/11/10 10:12:23	SEED	
OrangeFrost ( avc_msvideo_bL2 )	Check	OK	2007/11/10 10:11:37	SEED	
OrangeFrost ( avc_msvideo_bL2 )	Conf	OK	2007/11/10 10:10:16	SEED	
OrangeFrost ( avc_msvideo_bL2 )	Cell	OK	2007/11/13 07:21:34	ARAQ-WIN2	
OrangeFrost ( avc_msvideo_bL2 )	GccBuild	OK	2007/11/13 07:13:45	ARAQ-WIN2	
OrangeFrost ( avc_msvideo_pt )	Match	OK	2007/11/13 09:02:33	SEED	
OrangeFrost ( avc_msvideo_pt )	Conf	OK	2007/11/13 09:01:24	SEED	

# コミットルールの変更

- **コミット時にRemote Testを積極的に活用**
  - **コード変更の影響範囲が広い場合**
    - まず、開発コードをプライベートブランチにコミット
    - テスト対象にプライベートブランチを指定して、依存関係のあるプロジェクトのRemote Testを一斉実行
    - 問題なければ、trunkにマージし、コミットする
  - **コード変更の影響範囲が狭い場合**
    - コミット後、不安な場合にはtrunkのコードを使ってテストを実行する
- **実行の判断は各開発者に任せる**
  - **最低限のテストを一定時間毎に自動実行**

# まとめ

- **分散テストシステムdTest導入後の効果**
  - **開発中のソースコードへのバグ混入が減少**
    - コミット前により多くのテストを実行可能になった
    - デバッグ時間が減少
  - **テスト完了待ちの時間/ストレスが減少**
    - コーディング時の集中力を維持可能に
  - **ソースコードを安心して変更できるようになった**
    - リファクタリングが容易に
      - ソースコードの品質が向上
    - コードの共通化が促進
      - 開発効率の向上

The logo for dTest features the lowercase letter 'd' in red and the word 'Test' in blue, all in a bold, sans-serif font. The letters are set against a light gray background with a subtle grid pattern.

**dTest**

気持ちよくソフトウェアを開発するために