

# 自動進行型ネットワークゲーム 「プロ野球チームをつくろう！ONLINE2」 開発の試み

(株) ZENER NET WORKS

佐野 浩章

# このセッションの目的

- 自動進行型のネットワークゲーム  
「プロ野球チームをつくろう！ ONLINE2」の開発で  
最初にどのようなサーバプログラムの設計をしたか！  
そして開発、運営からどのような結果を得る事が  
できたのかを説明していきたいと思います。

(このセッションはサーバプログラマ向けセッションです。)

# アジェンダ

- 自己紹介
- 「プロ野球チームをつくらう！ONLINE2」？
- プログラム的試み
  - 第1章：CPUリソースの分散
  - 第2章：サーバダウン対応
  - 第3章：パケット設計
  - 第4章：コミュニティーのエンジン設計

# 自己紹介-1

- 1989年 100桁の素因数分解論文(電気学会)
- 1990年 日本電気株式会社入社
- 1996年 テクニカルサポート(SEGA)

サターン:E0、街...

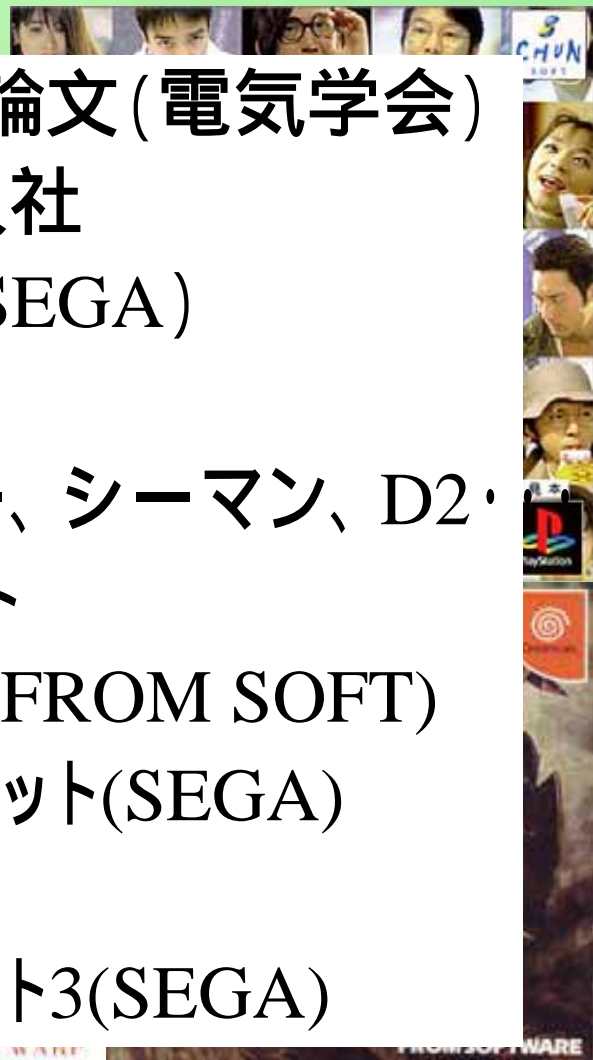
DC:ソニックアドベンチャー、シーマン、D2・  
ネットワークタイトルサポート

1999年 フレームグライド(FROM SOFT)

2000年 チューチューロケット(SEGA)

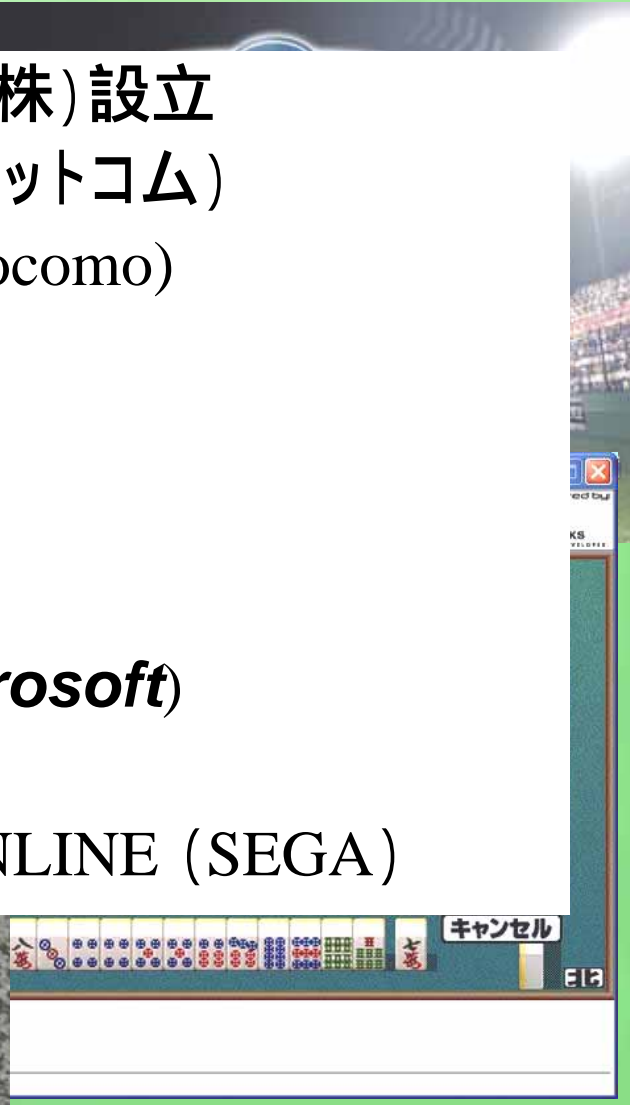
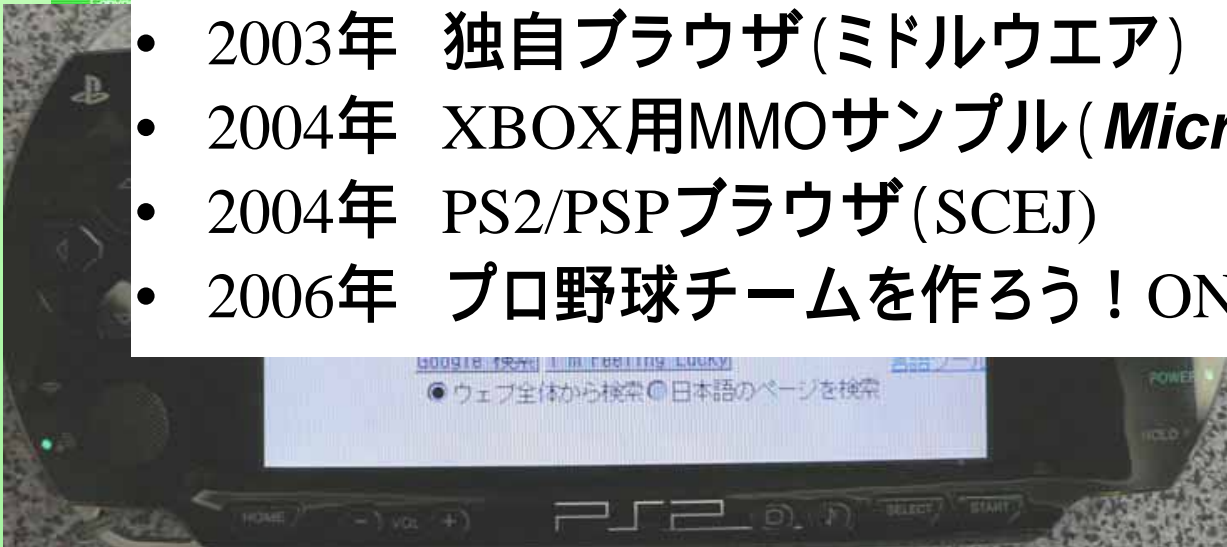
ネットワークに関する開発

2001年 ドリームパスポート3(SEGA)

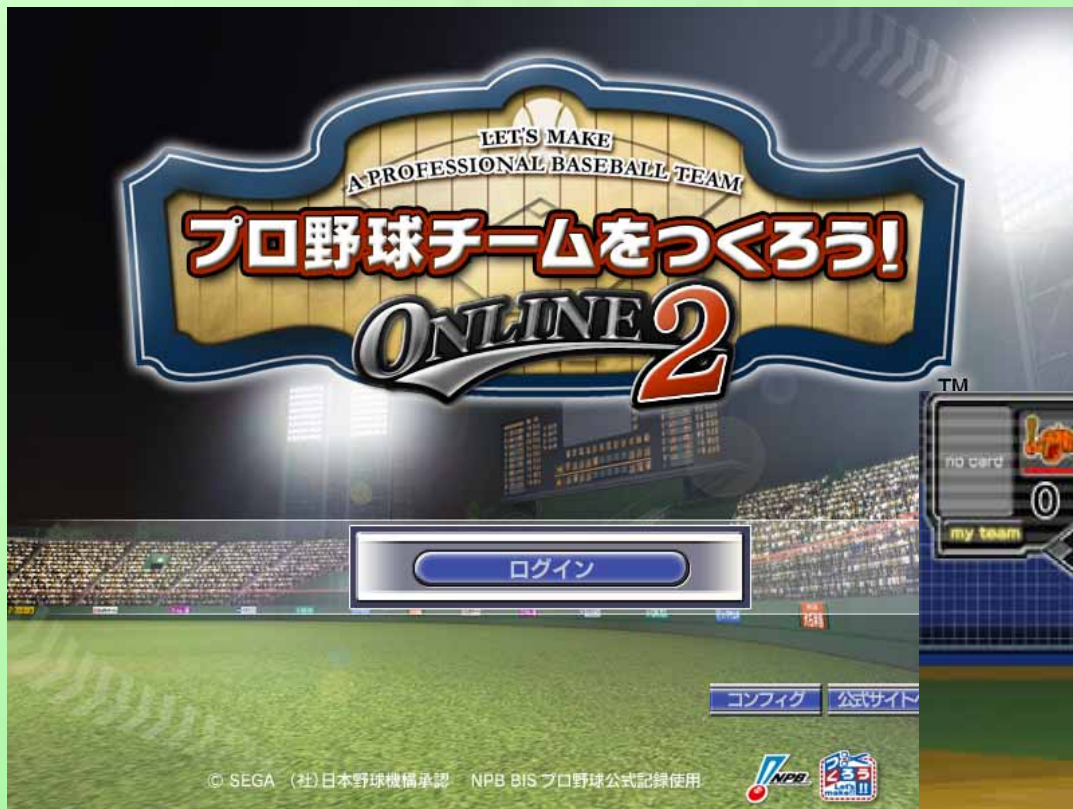


# 自己紹介-2

- 2001年7月 ZENER NET WORKS(株)設立
- 2001年 THEわれめDEポン(お台場ドットコム)
- 2001年 インターネット博覧会(NTT docomo)
- 2002年 THEわれめDEポン
- 2002年 ゲームパーク(J-COM)
- 2002年 O2
- 2003年 独自ブラウザ(ミドルウェア)
- 2004年 XBOX用MMOサンプル(*Microsoft*)
- 2004年 PS2/PSPブラウザ(SCEJ)
- 2006年 プロ野球チームを作ろう! ONLINE (SEGA)



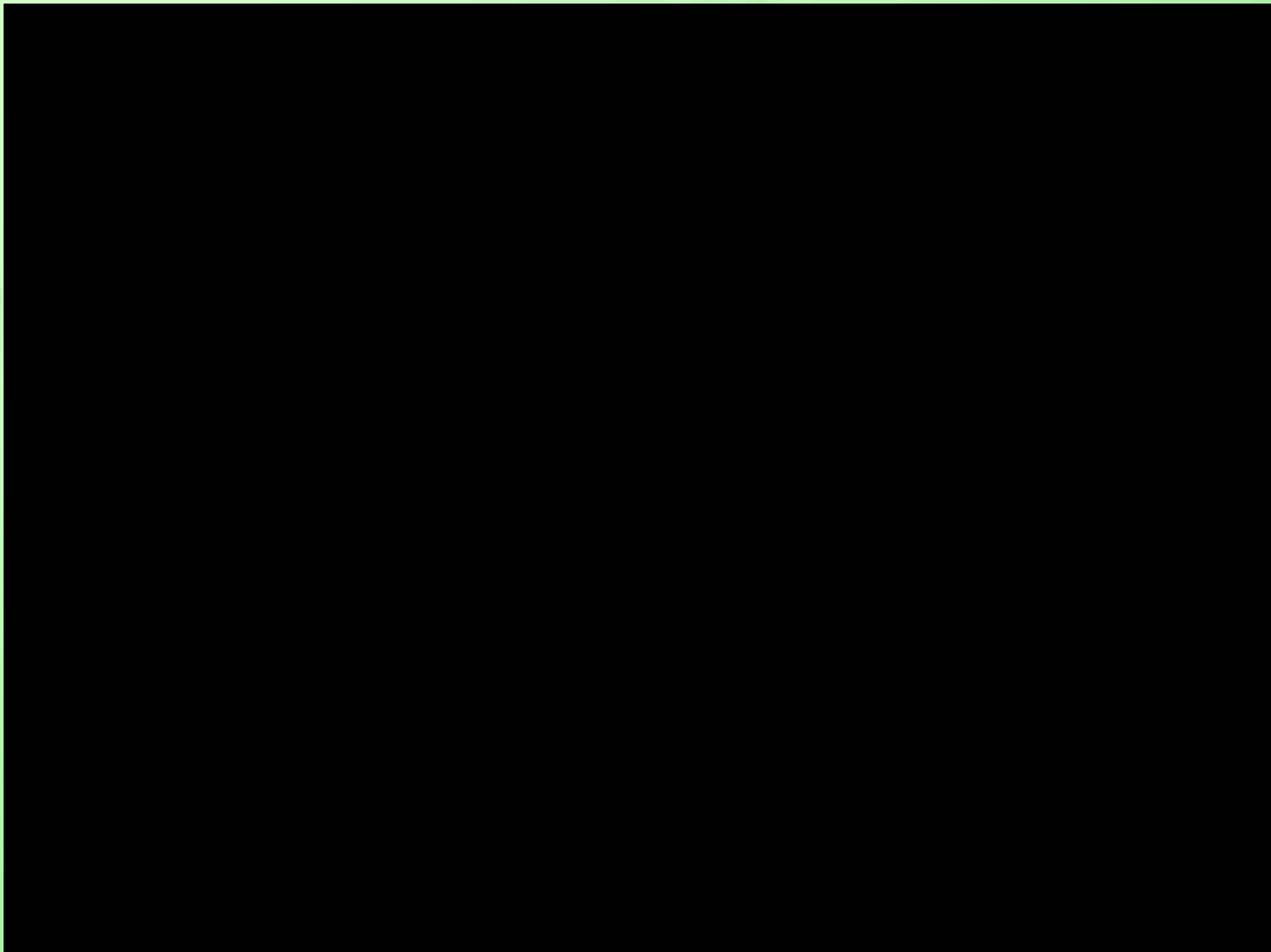
# プロ野球チームをつくらう！ ONLINE2





# 「プロ野球チームをつくろう！」の歴史

- ・プロ野球チームもつくろう!  
(1998年2月19日、セガサターン)
- ・プロ野球チームをつくろう!  
(1999年8月5日、ドリームキャスト)
- ・プロ野球チームをつくろう！アドバンス  
(2002年8月8日、ゲームボーイアドバンス)
- ・プロ野球チームをつくろう！2  
(2003年2月13日、プレイステーション2)
- ・プロ野球チームをつくろう！3  
(2005年7月28日、プレイステーション2)
- ・プロ野球チームをつくろう！  
(2008年5月22日、ニンテンドーDS)





# プロ野球チームをつくらう！ONLINE2？

- ユーザーはオーナーとなってチームを育成する
- ゲームプレイ時間
- 他にどんなタイトルがあるのか？

# ユーザーはオーナーとなって チームを育成する

- プレイヤーはオンライン中にチームに対し選手のオーダや采配等の指示設定をします。
- すべてのチームはサーバ内の時間に合わせ進行しています。
- 1日15分程度の短い時間でも、試合結果など最小限の確認でゲームを楽しむ事も可能です。

### 選手オーダー

1344 pt

野手オーダー      投手オーダー

★コスト 88/88    ★レア 1/1    ★レジェンド 0/0

スターティングオーダー

1	2	3	4	5	6	7	8	9
中堅	二塁	捕手	遊撃	左翼	二塁	右翼	DH	一塁

ベンチ

1	2	3	4	5

更新    取り消し    登録抹消

個別   全体   プラチナ   ファン球団   グループ   トレード   本拠地

全体

# 指示設定

### 采配・監督

1344 pt

- 野手    投手
- よく見て打て！
  - 大きいのを狙え！
  - 積極的に走れ！
  - 確実に進めろ！
  - 野球の基本は守りだ！

チーム采配

監督にまかせた！

監督カード

MANAGER CARD  
empty

監督設定

決定    取り外し    閉じる

[メンテナンス情報] 定期メンテナンスは毎週火曜日14:00~16:00

個別   全体   プラチナ   ファン球団   グループ   トレード   本拠地

全体

# 待つ・・・

- Logoutしてゲームの進行を待ちます。
- そしてまたLogin

試合結果

38064 pt

最新 6 試合を閲覧します。表示形式を選んでください。  
未見の試合があるうちは、メイン画面に進めません。

第13サイクル

第85試合	WIN	910pt	3位	3 - 1	...	...
第86試合	WIN	850pt	3位	4 - 2	...	...
第87試合	LOSE	760pt	4位	8 - 10	...	...
第88試合	-	- pt	-位	8/20 12:00~	...	...
第89試合	-	- pt	-位	8/20 13:30~	...	...
第90試合	-	- pt	-位	8/20 15:00~	...	...
第91試合	-	- pt	-位	8/20 16:30~	...	...
第92試合	-	- pt	-位	8/20 18:00~	...	...

# 結果確認

プロ野球チームをつくろう！ ONLINE 2 Ver.080731

スコアボード 38604 pt

ペナント戦 第89試合

ぼこぼこ VS 岩隈

	1	2	3	4	5	6	7	8	9	計	勝	敗	pt
ぼこぼこ	2	0	2	3	0	0	2	1	1	11	1	0	1550
岩隈	1	0	0	0	0	0	0	0	1	2	5	0	308

対戦成績 4勝1敗1分  
勝利投手 岩隈 (8勝1敗0S)  
セーブ  
敗戦投手 手塚 (4勝8敗0S)  
本塁打 ジェイジェイ2号 (3回ソロ) 大西6号 (4回ソロ)  
投手リレー 岩隈  
手塚→梅津→吉野→遠藤  
獲得ポイント 1550pt

試合カード

先攻成績 後攻成績 次の試合へ 閉じる

プロ野球チームをつくろう！ ONLINE 2 Ver.080731

取得pt 10

試合結果へ

no card 0 1 回表 0 no card

my team S B O my team

田島研究所

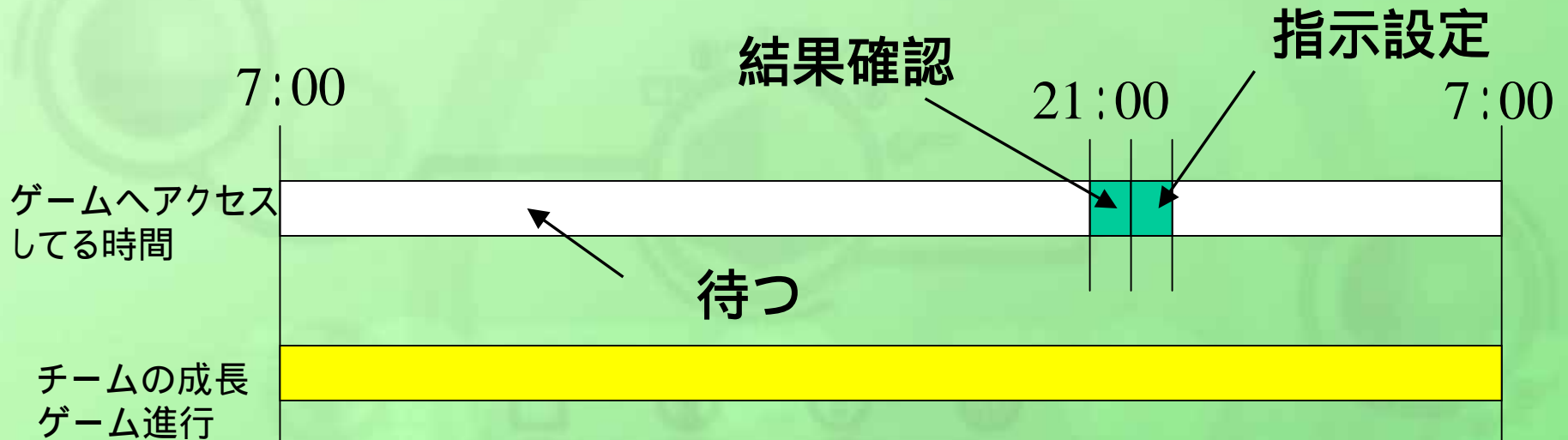
投手 石川 雅規

2番 坂本 勇人

打率	本塁打	打点	盗塁
.248	1	3	0

【お知らせ】

# ゲームプレイ時間

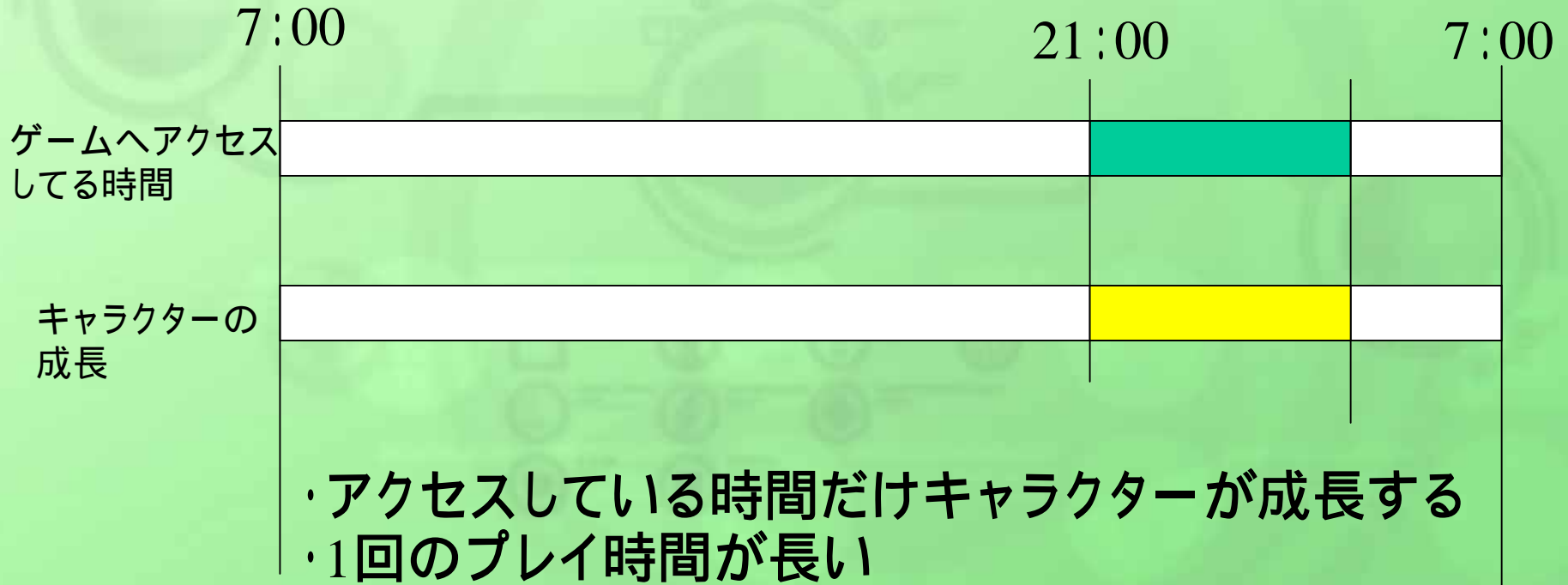


- ・アクセスしていない間もチームが成長する
- ・1回のプレイ時間が短い
- ・ユーザは「指示設定」「待つ」「結果確認」の作業になる



# ゲームプレイ時間の比較

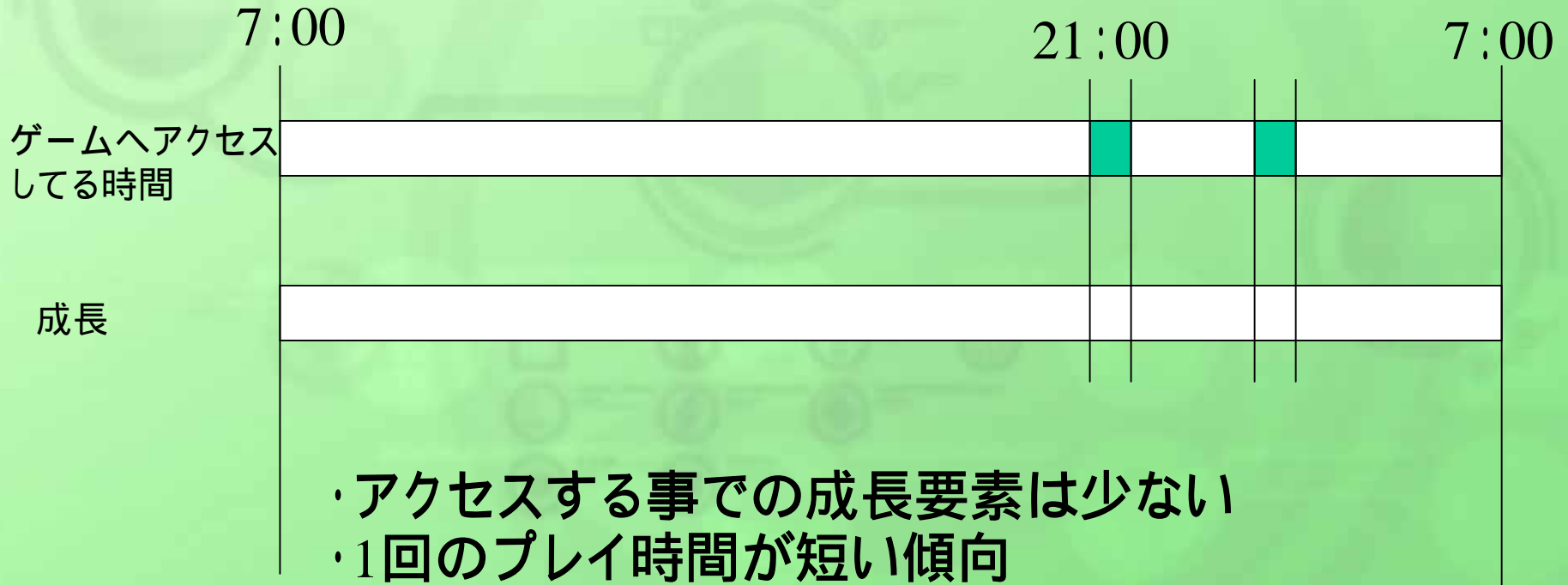
- MMO/MOの場合





# ゲームプレイ時間の比較

- ・ カジュアルゲームの場合



# 他にどんなタイトルがあるのか？

- プロサッカークラブをつくらう！ ONLINE



# プログラムの試み

- 第1章: CPUリソースの分散
- 第2章: サーバダウン対応
- 第3章: パケットの設計
- 第4章: コミュニティのエンジン設計

開発: ゲーム開発

運営: イベント、キャンペーン(管理TOOL)

運用: サーバの維持

# 第1章: CPUリソースの分散

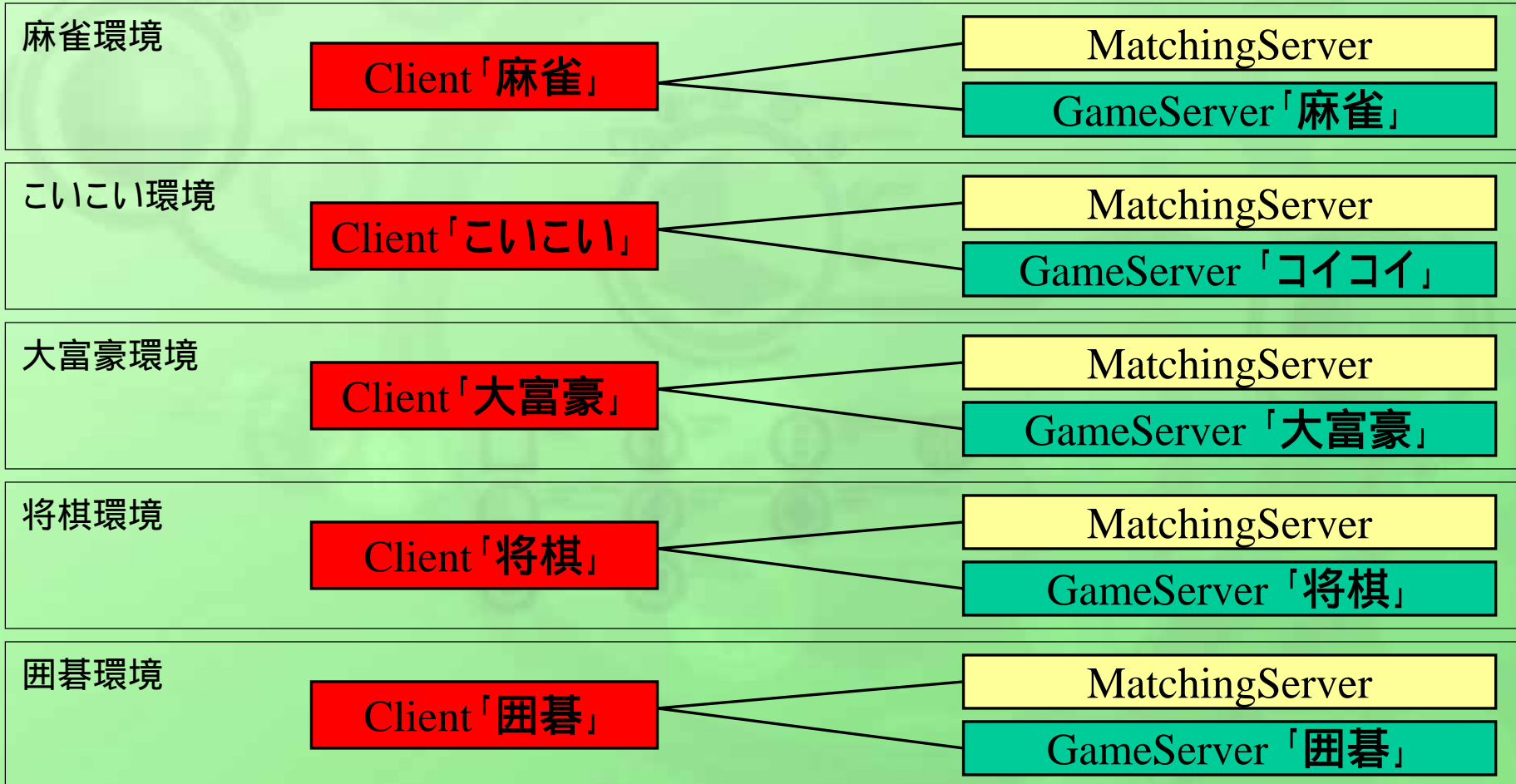
- サーバのリソースは常に余裕を持って設計されている。(ピーク時に備えて)







# カジュアルゲーム開発時の設計





# 初めてCPUリソースを意識する



The screenshot shows the O2 System website with a navigation bar at the top containing 'freeware' and 'onion software home page'. The main content area features the O2 System logo, which consists of two interlocking rings and the text 'O2 system' with 'onion software online 2002' underneath. To the right, there are two blue navigation menus. The first menu, titled 'O2オフィシャル', lists links for 'O2紹介とダウンロード' (dated 2002/10/01), 'O2ユーザー登録(無料)' (dated 2002/06/10), '最新の更新、サーバー情報' (dated 2006/07/24), 'よくある質問(FAQ)' (dated 2002/06/11), and 'O2ユーザー掲示板 (準備中)'. The second menu, titled 'O2対応ソフト', lists 'O2対応ソフトが次々に登場。O2ユーザー登録(無料)をすれば、すぐにダウンロードして楽しむことができます。'. Below the navigation menus, there are three news items, each starting with a red circular icon containing a white 'G' and followed by a title and date: 'HSP/O2 Beta Development' (2008/08/07), 'O2対応ソフト「はさみねこ」が公開されました。(2006/07/24)', and 'O2対応ソフト「実写でボクシング」が公開されました。(2006/05/08)'. A small thumbnail image of a game interface is visible in the bottom right corner.

freeware onion software home page



**O2 オフィシャル**

- [O2紹介とダウンロード](#)  
(2002/10/01)
- [O2ユーザー登録\(無料\)](#)  
(2002/06/10)
- [最新の更新、サーバー情報](#)  
(2006/07/24)
- [よくある質問\(FAQ\)](#)  
(2002/06/11)
- [O2ユーザー掲示板](#)  
(準備中)

**O2対応ソフト**

O2対応ソフトが次々に登場。  
O2ユーザー登録(無料)をすれば、  
すぐにダウンロードして楽しむことができます。

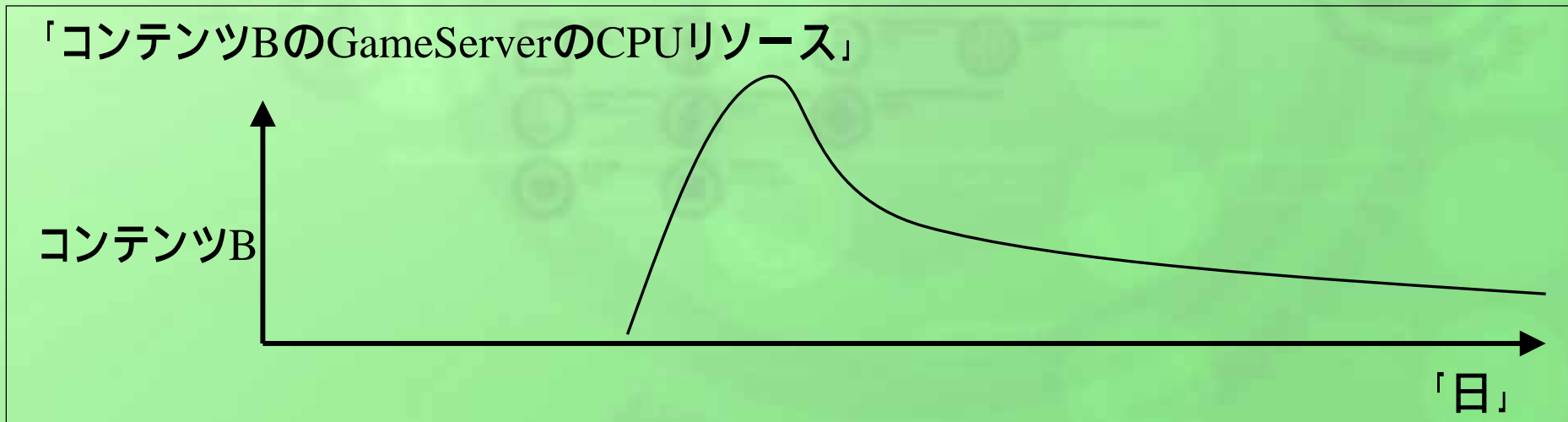
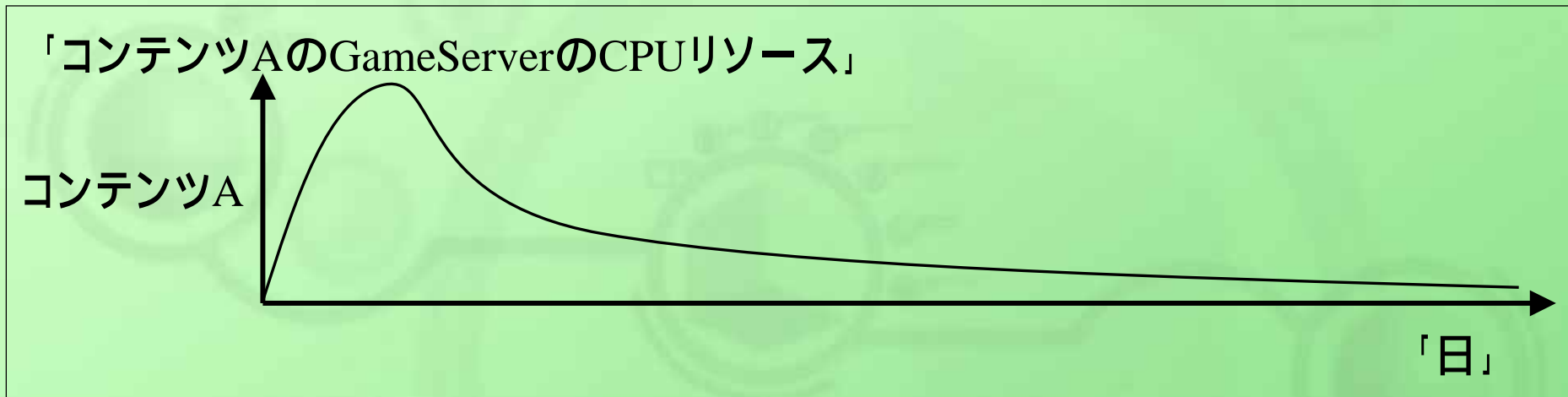
**G HSP/O2 Beta Development**  
2008年8月8日10:00~17:00まで、サーバーメンテナンスを行ないます。(2008/08/07)

O2対応ソフト「はさみねこ」が公開されました。(2006/07/24)

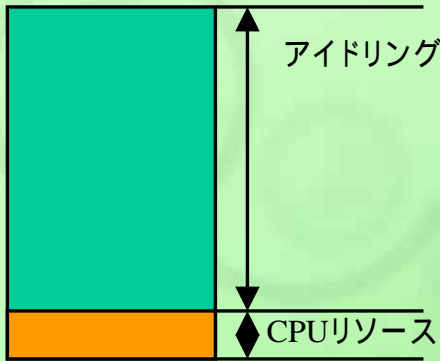
O2対応ソフト「実写でボクシング」が公開されました。(2006/05/08)



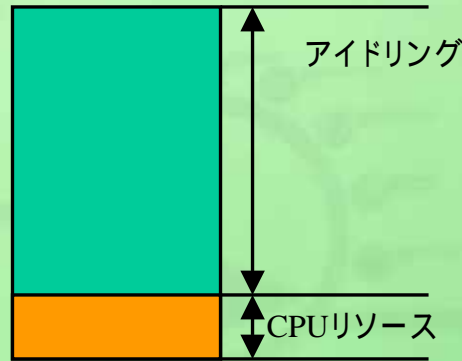
# ここのコンテンツは似たようなグラフ



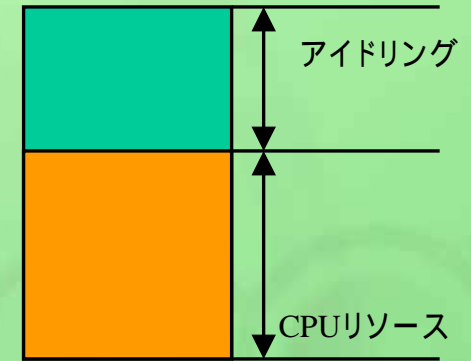
# CPUリソースは常に余っていた



コンテンツA



コンテンツB

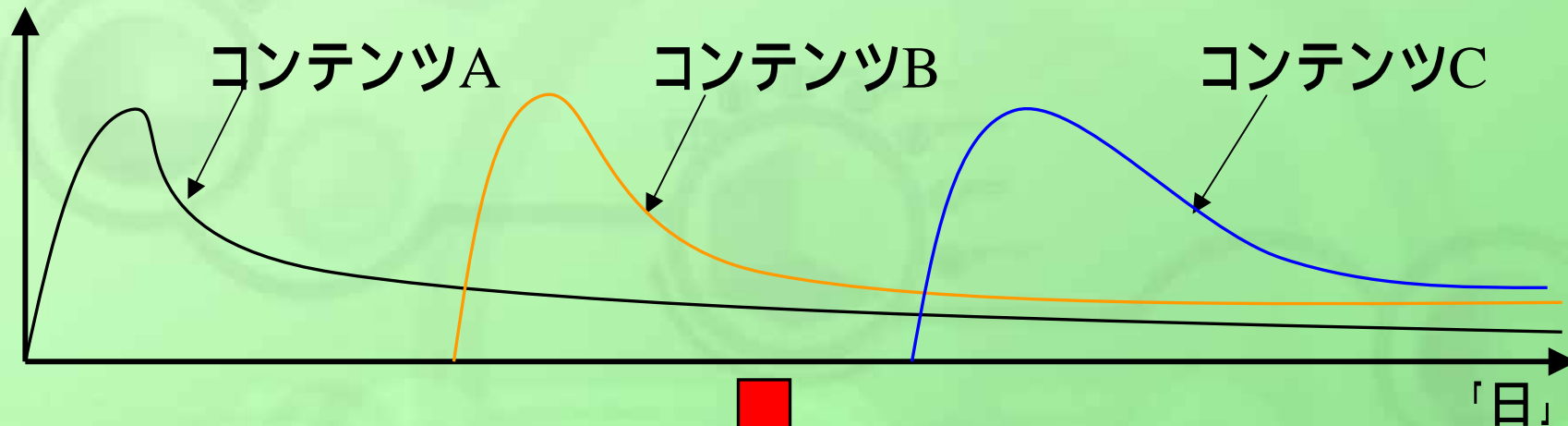


コンテンツC

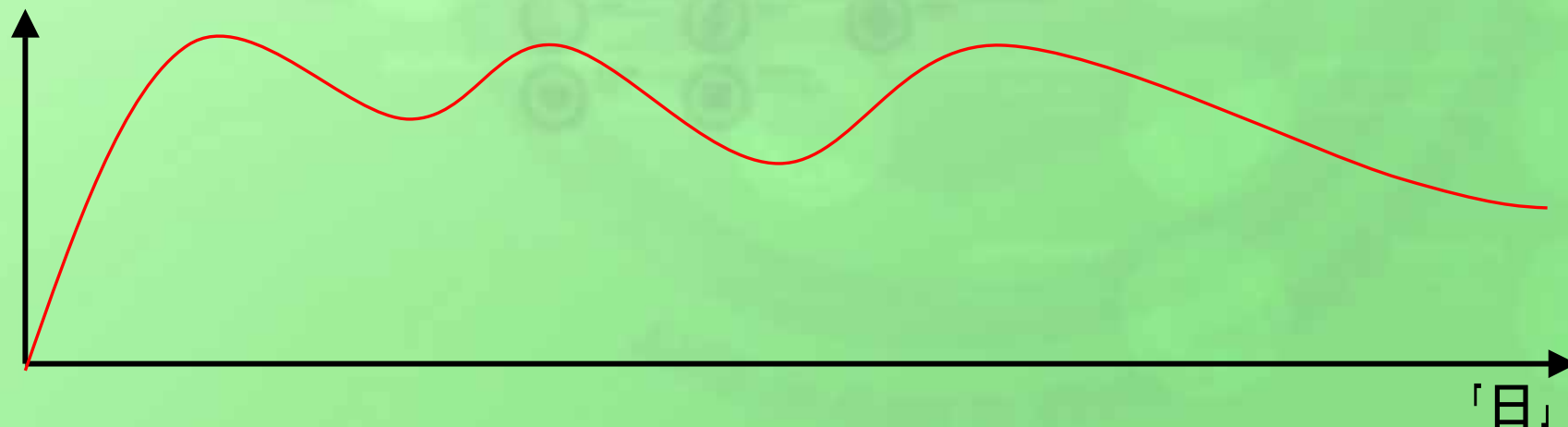
ピーク時の事を考えCPUのリソースが余り気味な状態

# CPUリソースをあわせる

「GameServer個々のCPUリソース」

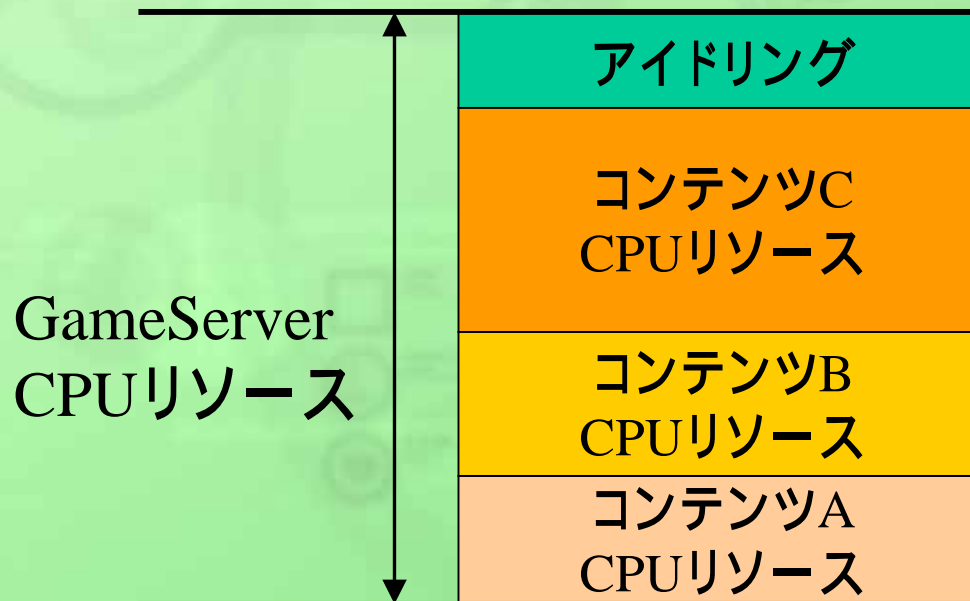


「CPUリソースの合計」



# 実際の設計では

- アクセス数でCPUリソースを配分

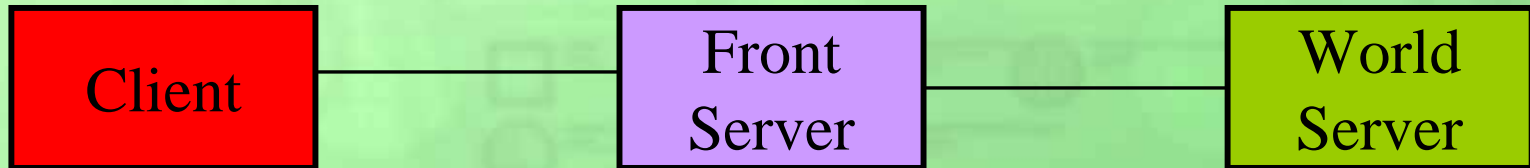


# 「プロ野球チームをつくろう！ ONLINE」開発開始

- 「プロ野球チームをつくろう！ONLINE」  
今までとは規模が異なる設計が必要

# 「同時アクセス数」と「登録ユーザー数」

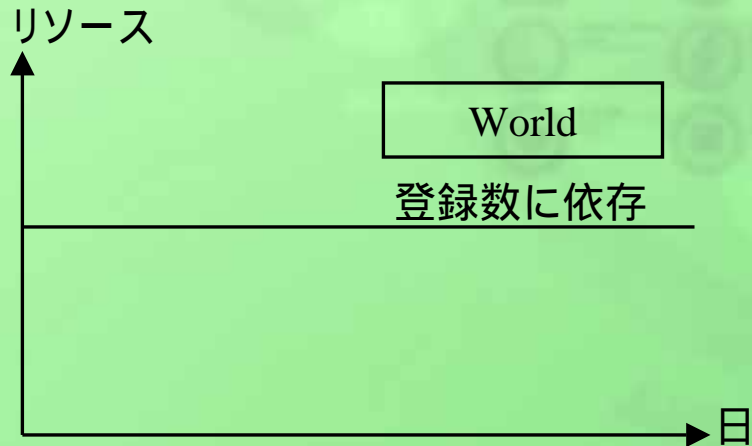
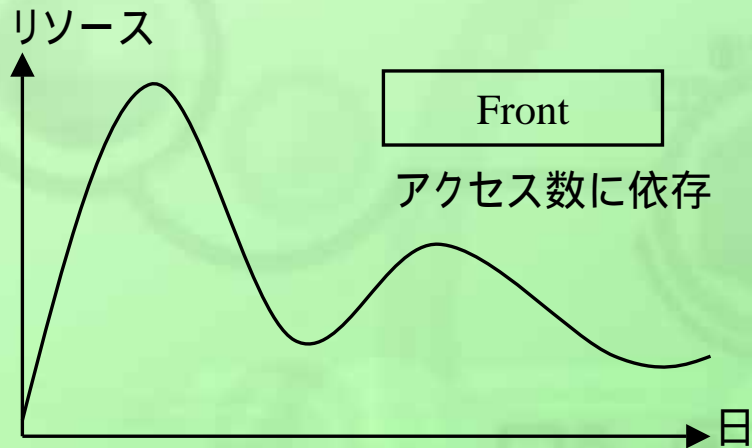
- 「同時アクセス数」に依存するフロントサーバ
- 「登録ユーザー数」に依存するWorldサーバに分離



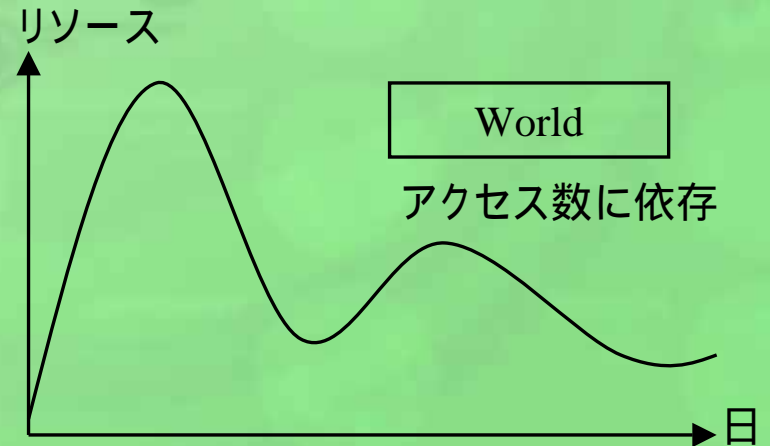
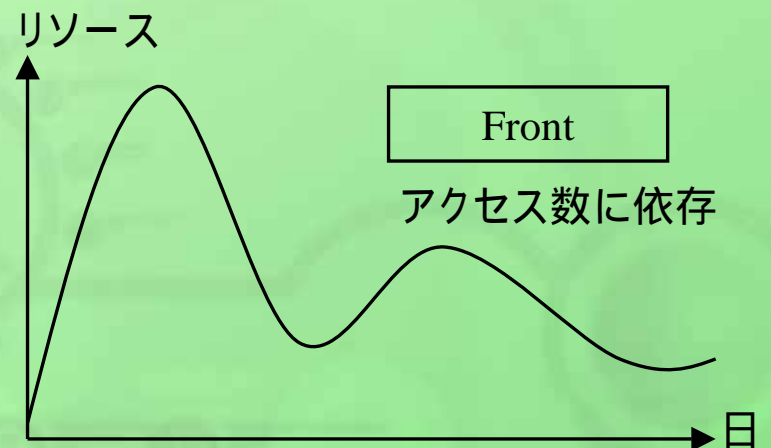


# リソースの違い

「自動進行型」



「MMO/MO」



## 「同時アクセス数」と「登録ユーザー数」

- 「登録ユーザー数」に制限をかけられるが「同時アクセス数」は実際運営してみないと、その数値がわかりにくい。

100% : 登録ユーザー数

15 ~ 25% : 1日のユニークなアクセスユーザー数

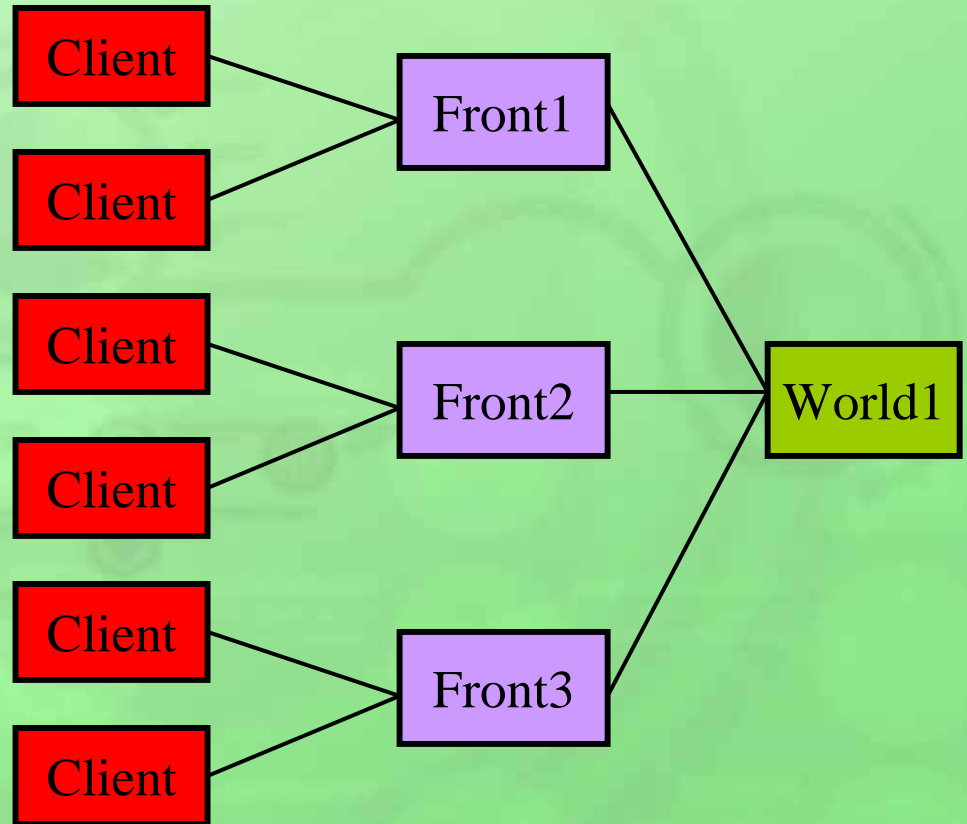
5% : 同時接続数のMAX値

# Frontのリソース

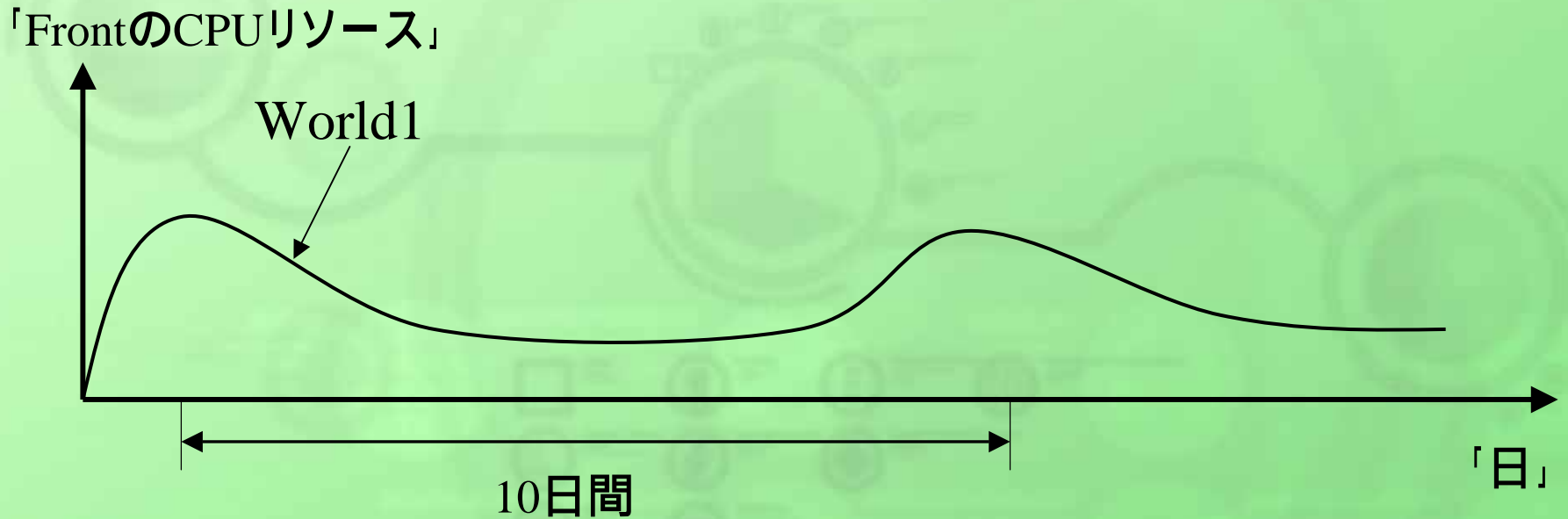
3つのFrontが処理できる  
CPUリソース

アイドル

World1のユーザとしてFrontが  
処理するCPUリソース

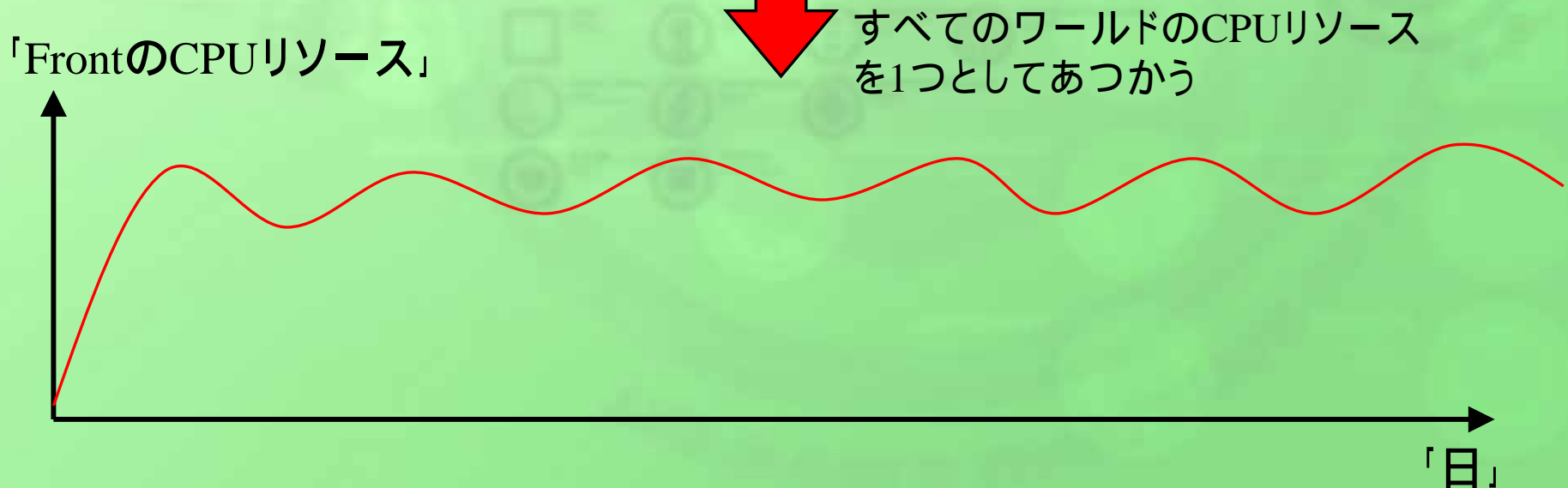
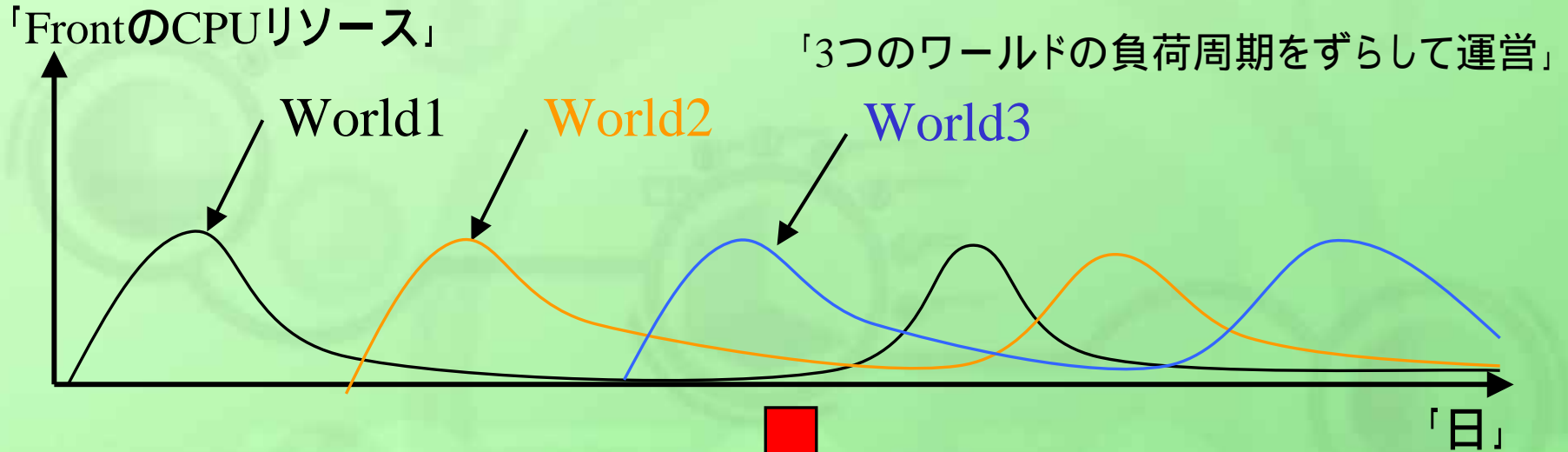


# Frontのリソース



- ・1つのワールドだと、FrontのCPUリソースに余裕が生まれる。

# Frontのリソースをまとめる

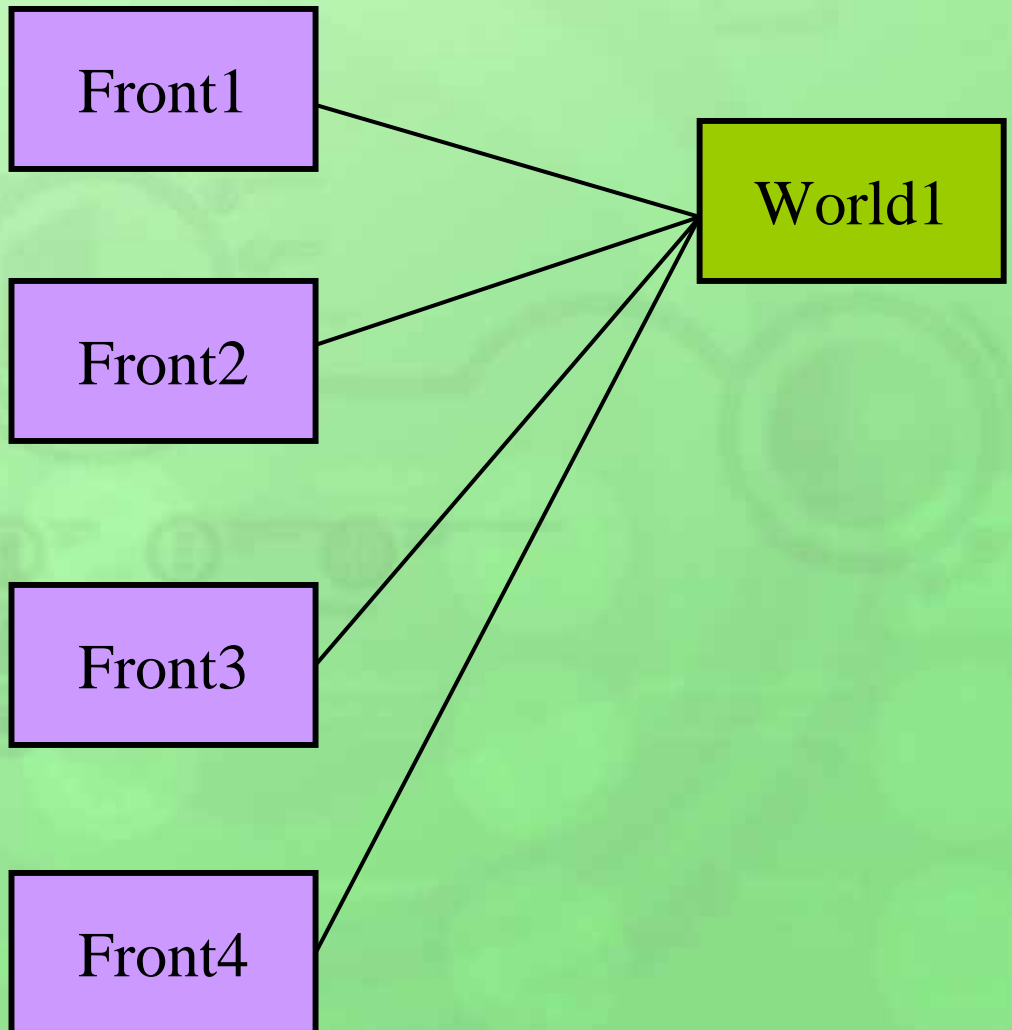


# Frontのリソースをまとめる-1

4つのFrontが処理できる  
CPUリソース

アイドル

World1のユーザとしてFrontが  
処理するCPUリソース



# Frontのリソースをまとめる-2

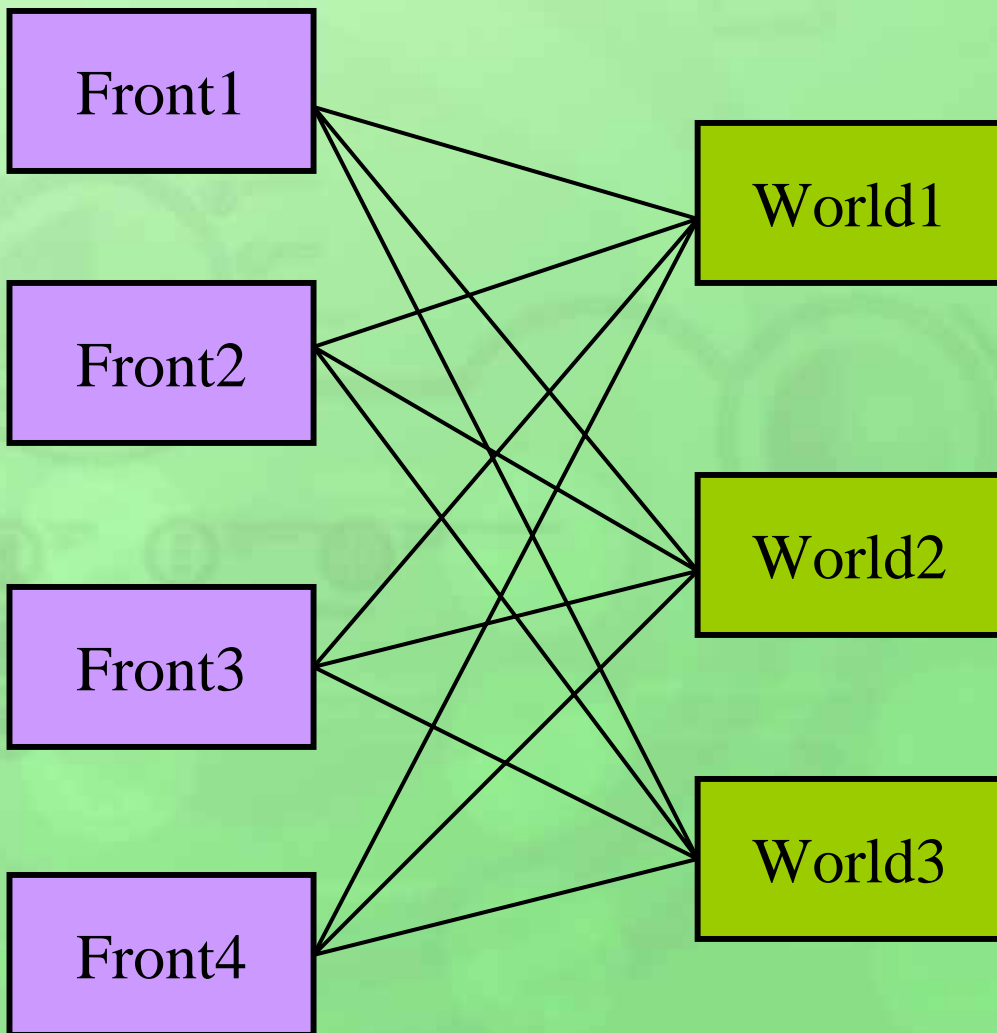
4つのFrontが処理できる  
CPUリソース

アイドル

World3のユーザとしてFrontが  
処理するCPUリソース

World2のユーザとしてFrontが  
処理するCPUリソース

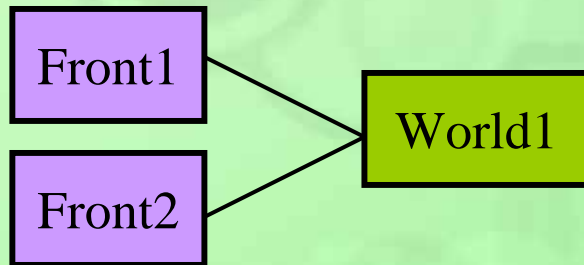
World1のユーザとしてFrontが  
処理するCPUリソース



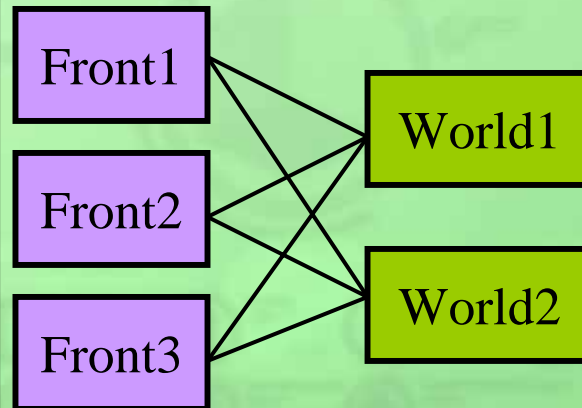


# 実際のサーバ推移

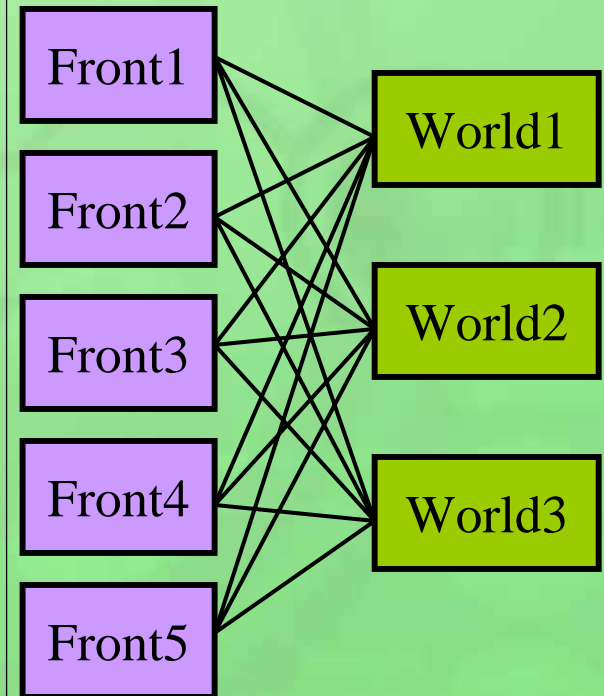
World 1個  
Front 2個



World 2個  
Front 3個



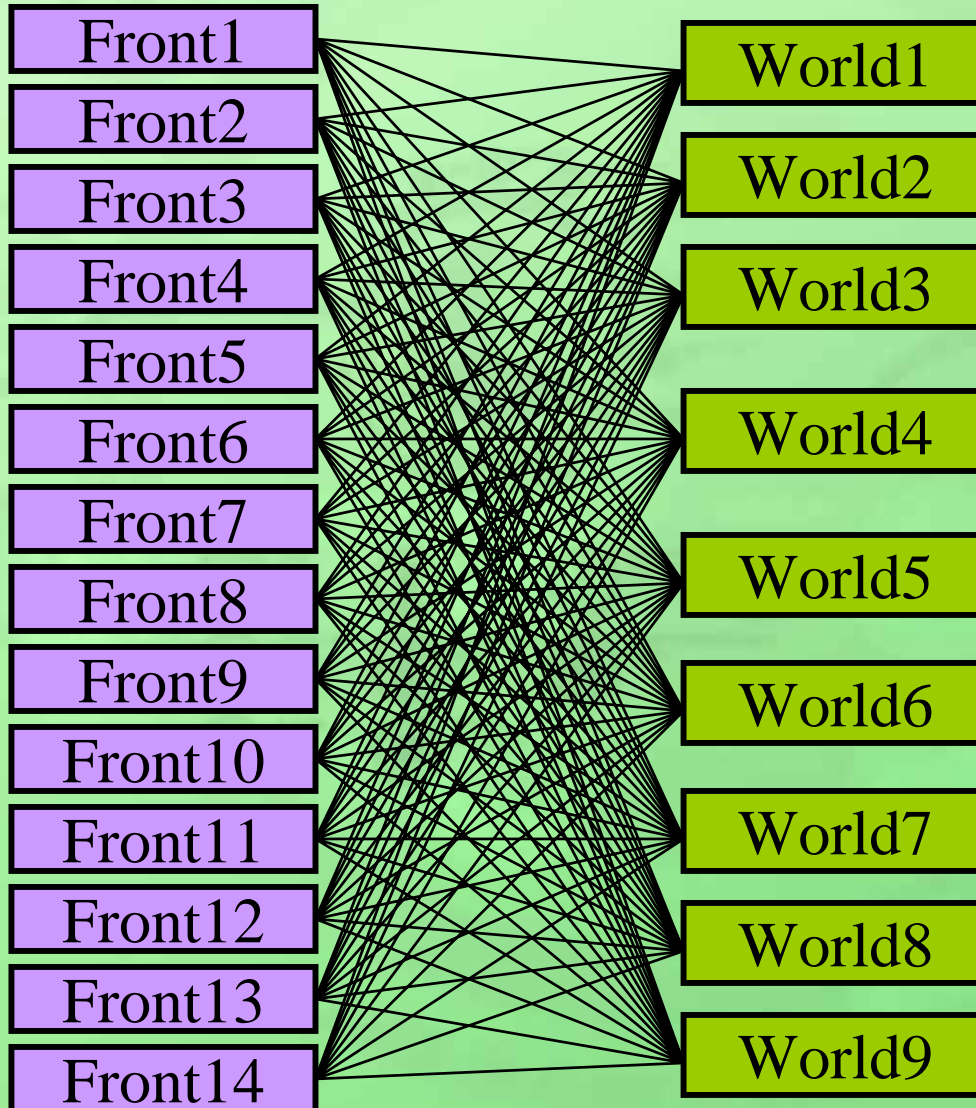
World 3個  
Front 5個



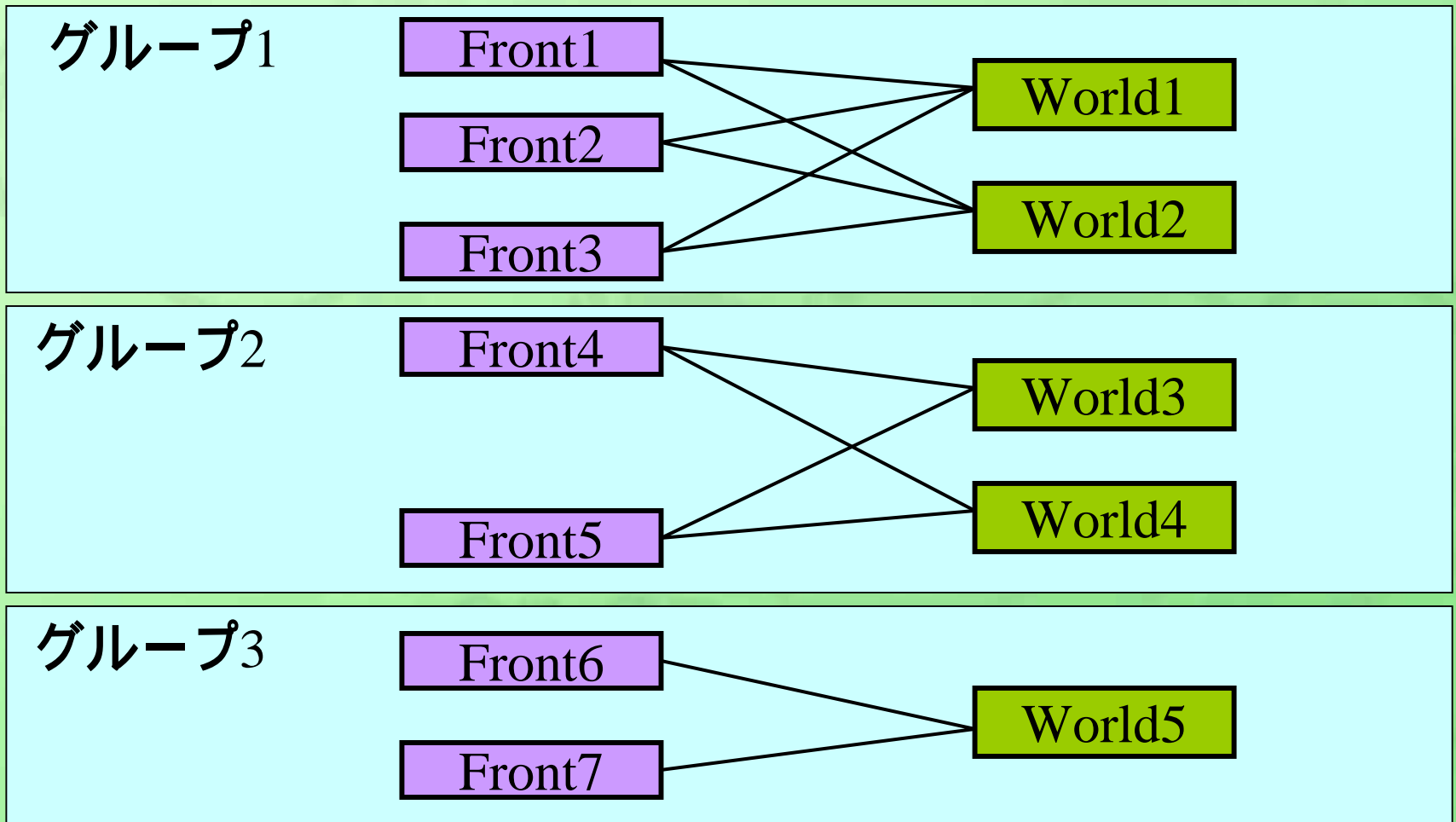
# 設計の危機

- 「プロ野球チームをつくろう！ONLINE2」の開発が決定
- 「月額課金」から「月額無料+アイテムコマース」へと変更
- 総登録ユーザ数の設定を大幅にみなおす必要がでてきました(ワールド数の大幅な増強が必要)

# このままでは・・・



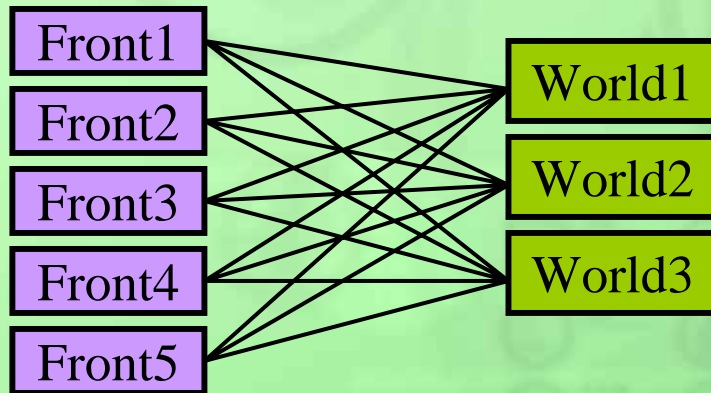
# グループ化



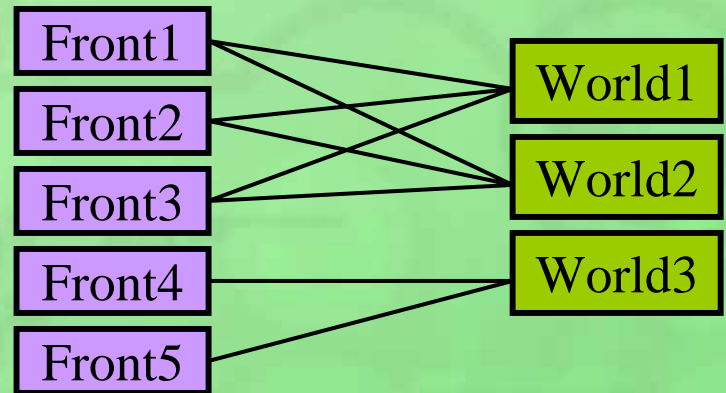
# グループができるか？

2008年4月8日、テストを兼ねてサーバ構成変更

旧システム



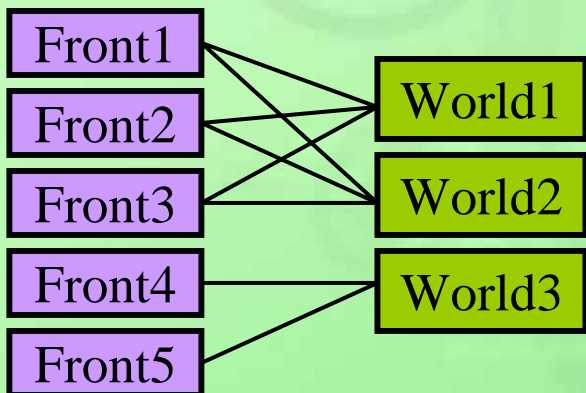
新システム



# 無料化開始

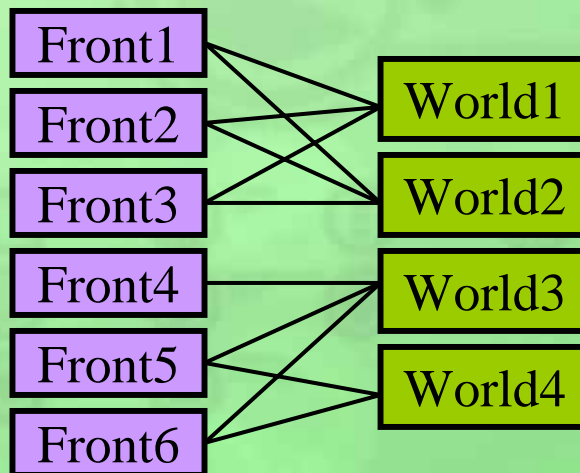
2008/4/8

Front 5個  
World 3個



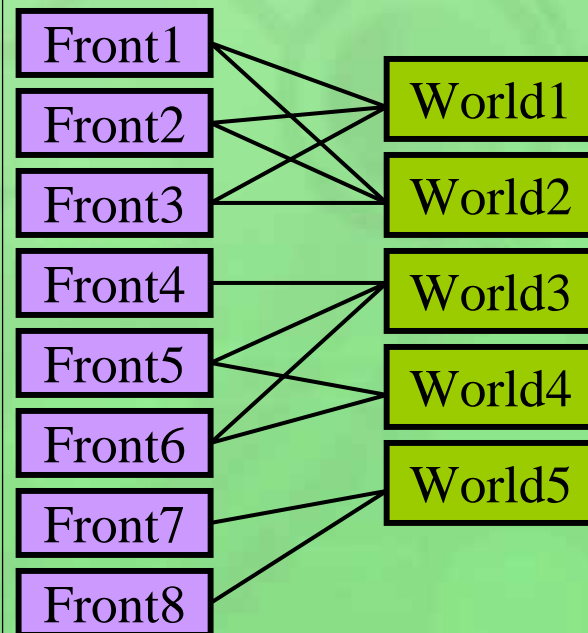
2008/5/27

Front 6個  
World 4個



2008/5/29

Front 8個  
World 5個



## 負荷がWorldで異なる

- 問題

新しいサーバにユーザが集中してしまう。

- 対策

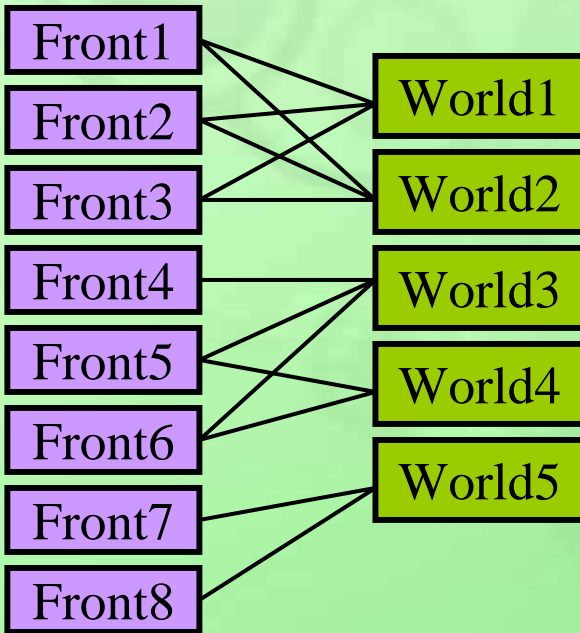
新しいWorldに対し多めにFrontServerのCPUリソースを割り振る。



# 新サーバ優遇構成

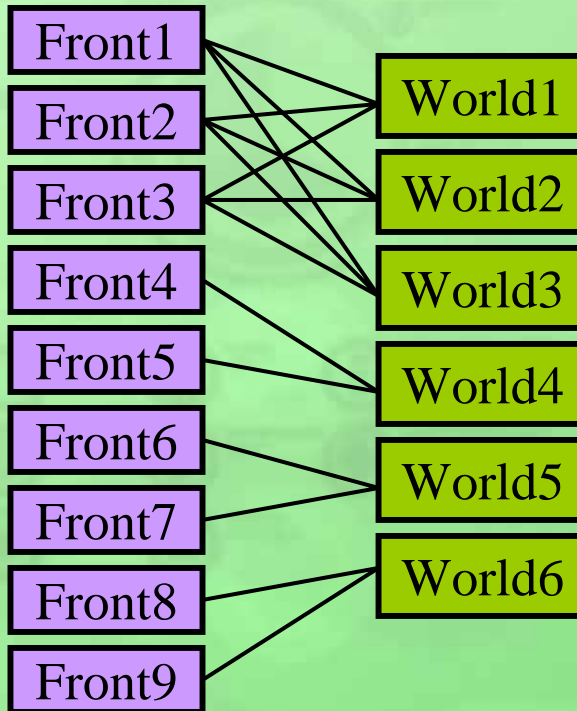
2008/5/29

Front 8個  
World 5個



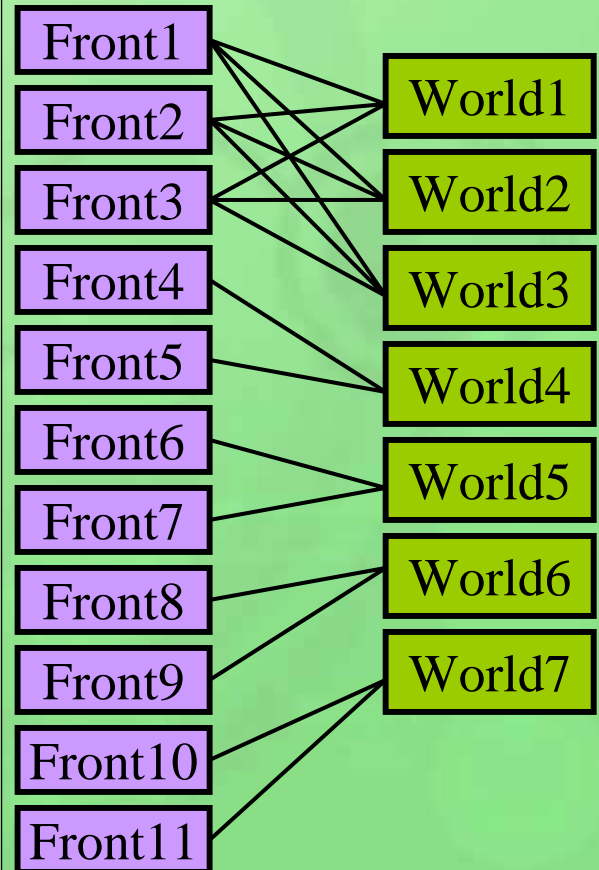
2008/6/3

Front 9個  
World 6個



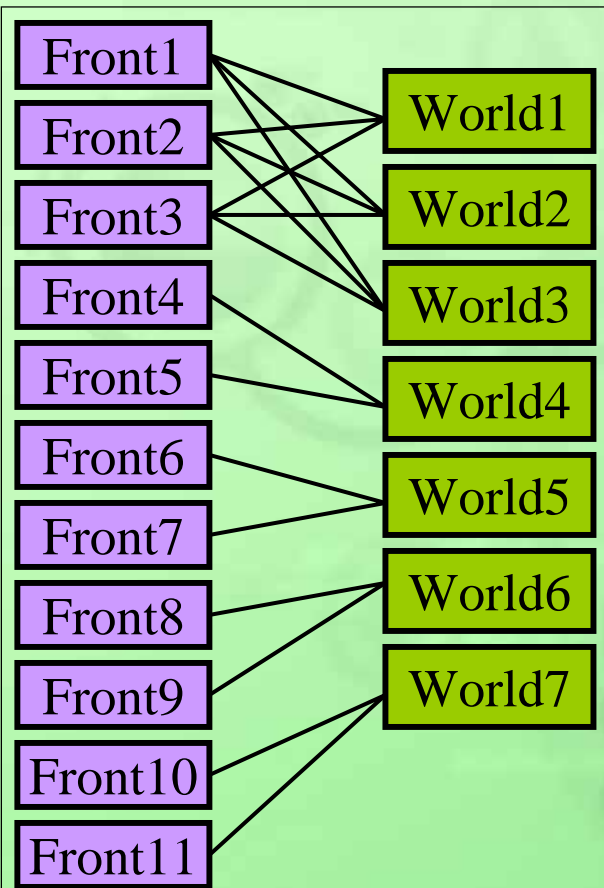
2008/6/13

Front 11個  
World 7個

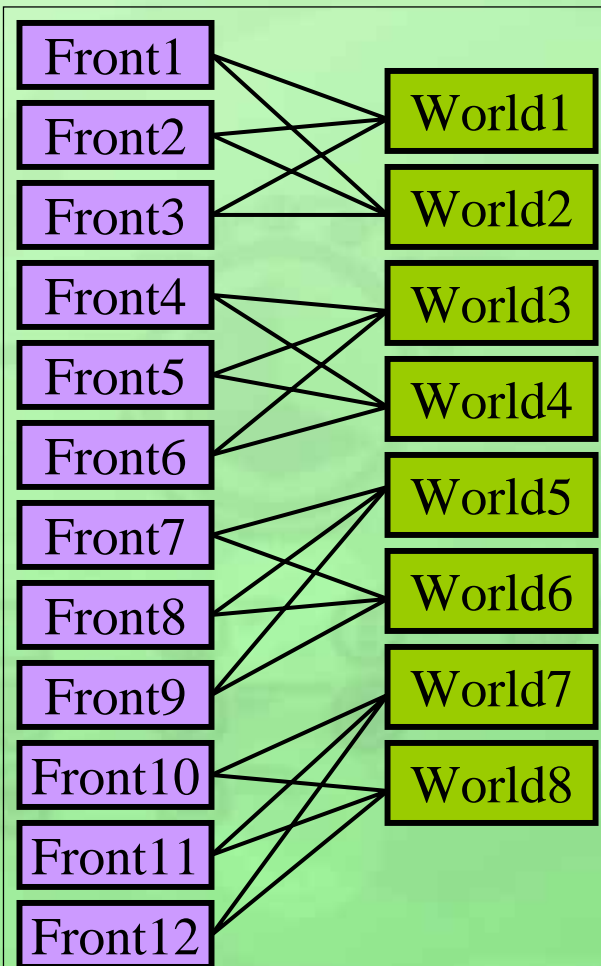


# 負荷の平均化

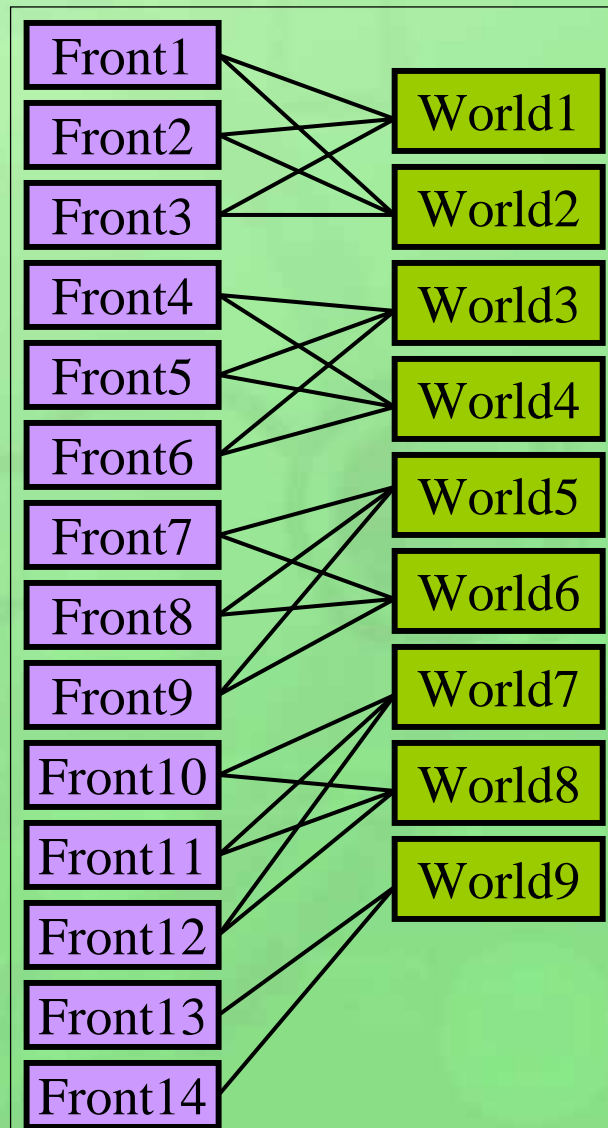
2008/6/13



2008/6/24



2008/7/17



# サーバ増えすぎ

- **問題**

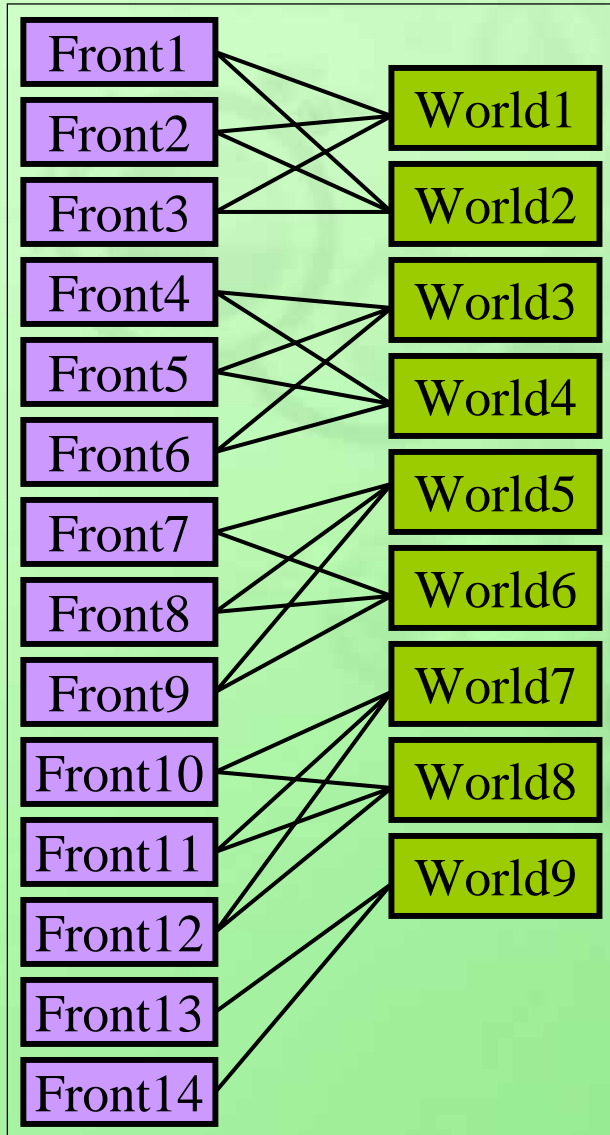
データセンターにサーバが入りきらない  
データセンターの電源が足りない

- **対策**

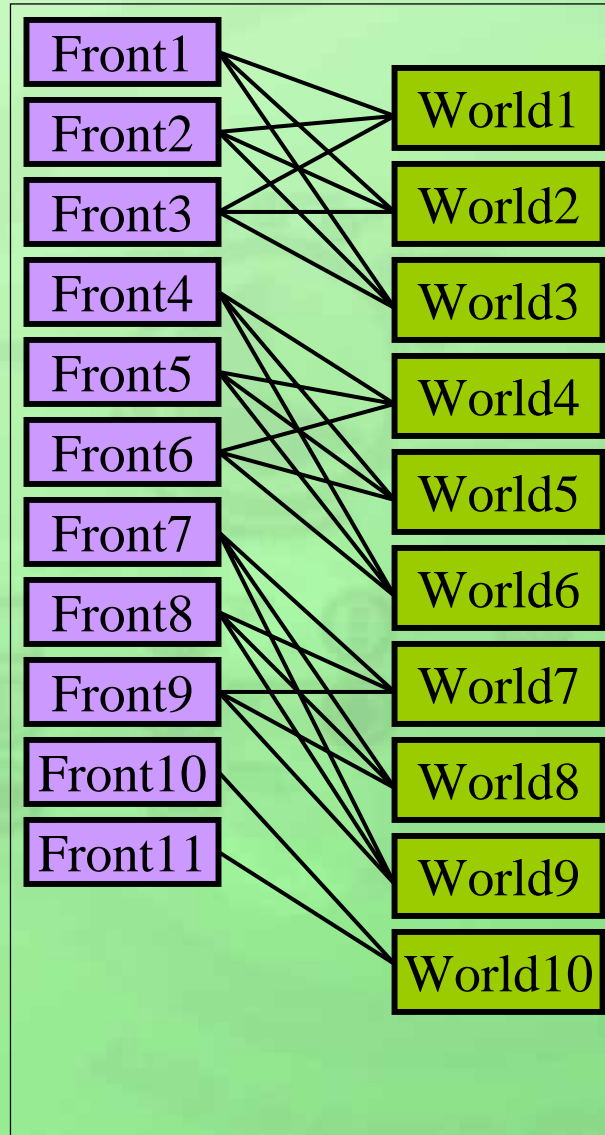
各FrontServerの負荷を上げて、使用する  
FrontServerの数を減らす

# Frontサーバ調整

2008/7/17



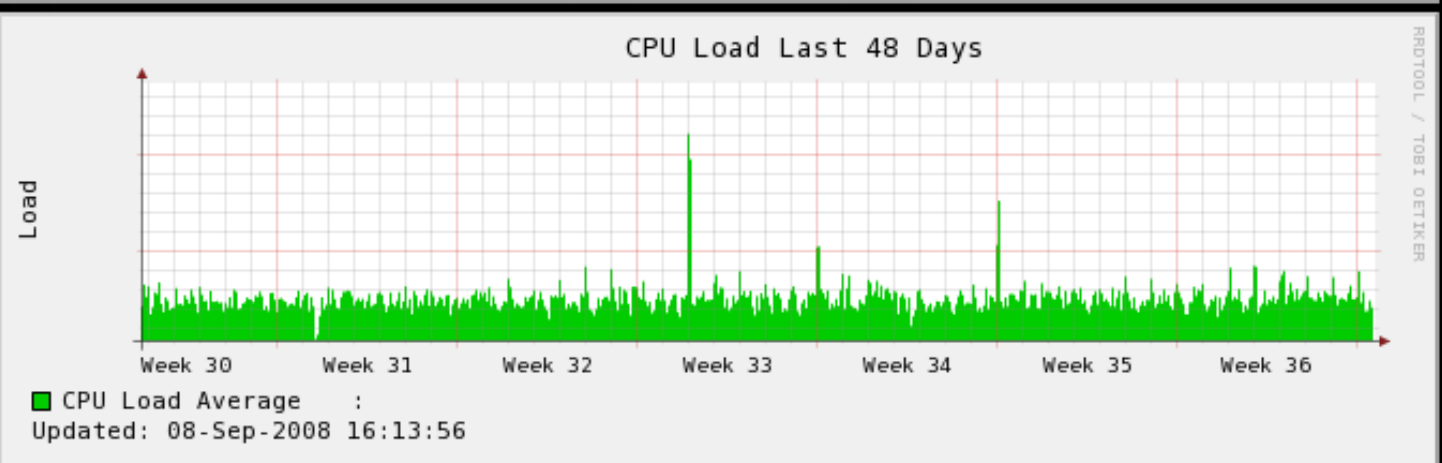
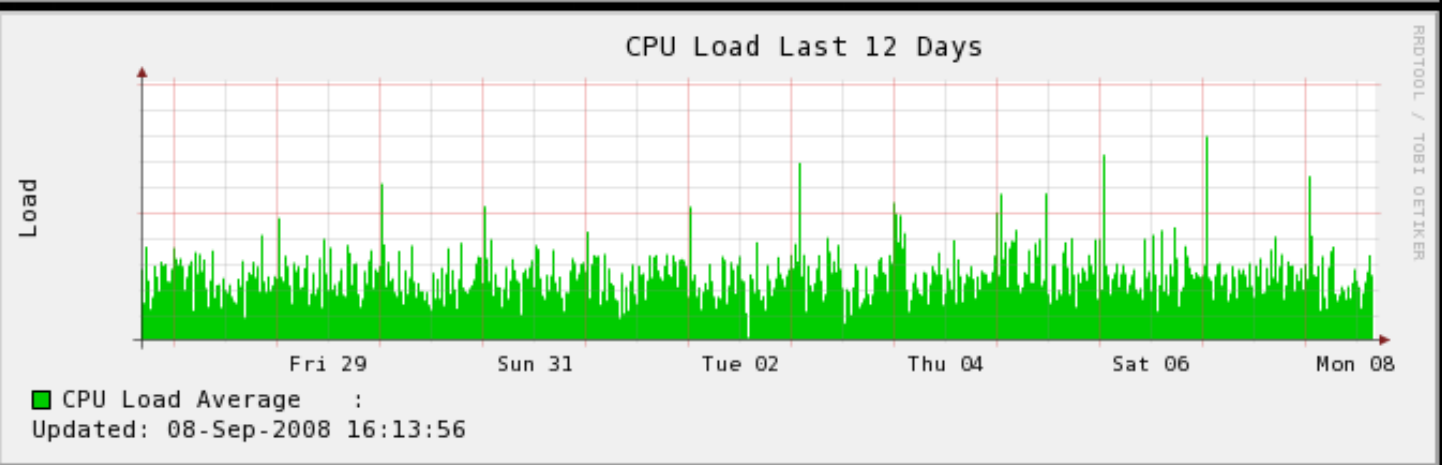
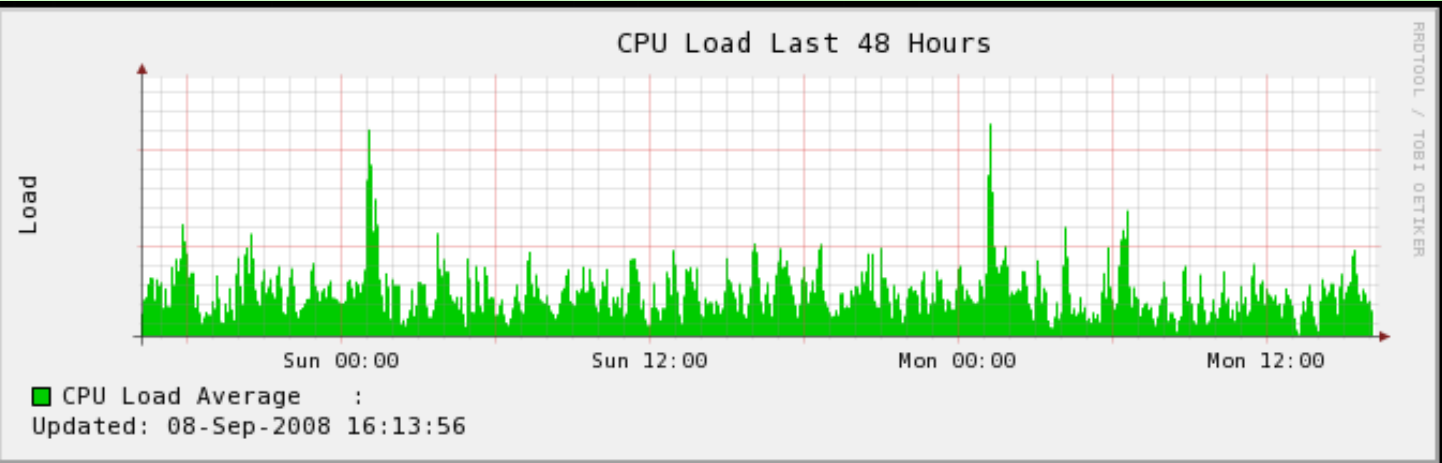
2008/8/5



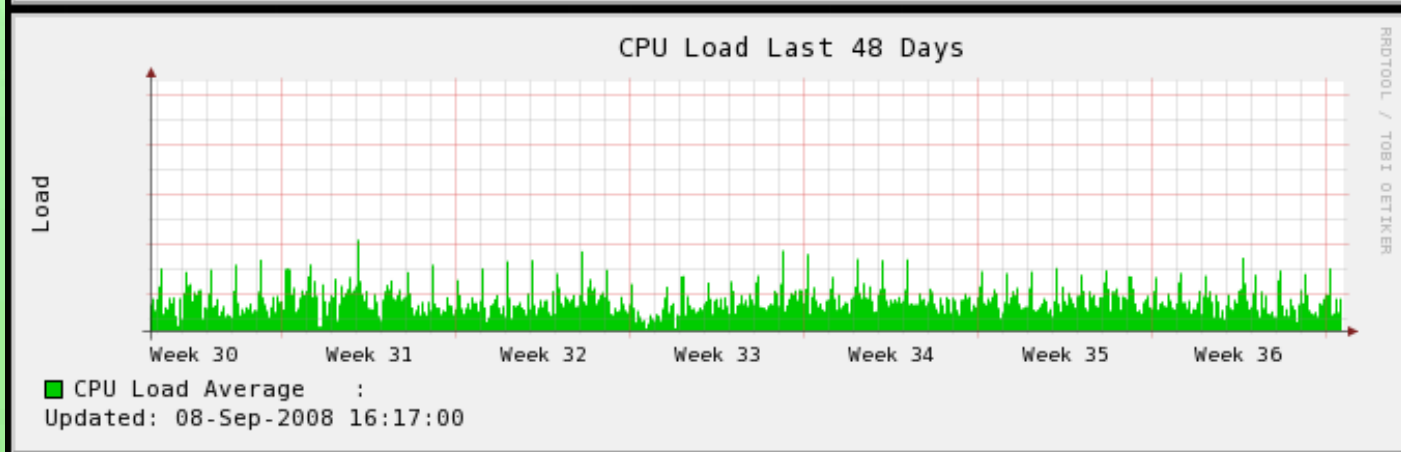
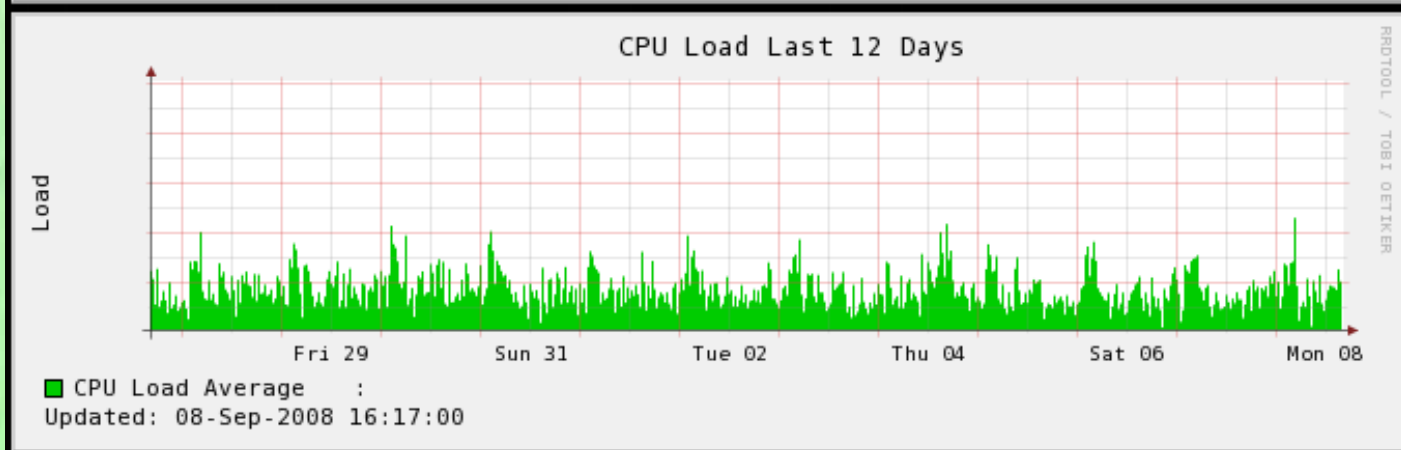
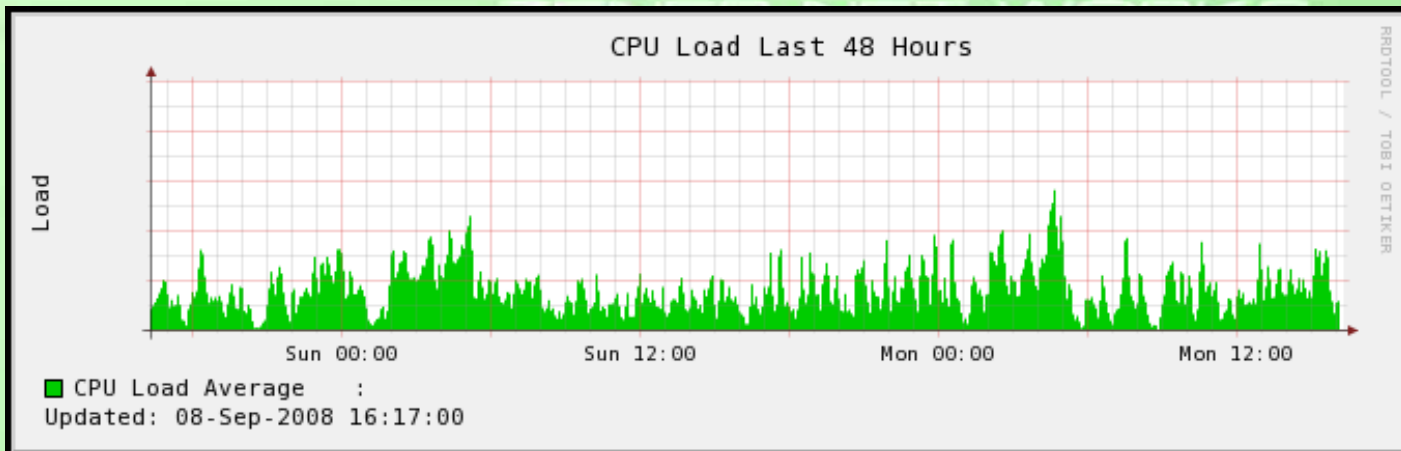
# 「CPUリソースの分散」まとめ

- 運営してみないとわからないので、変更しやすい構成にしておきましょう。
- サーバのリソースは有限。

# Front 負荷



# World 負荷





# 数値

## MMOやカジュアルゲームの経験値

- 100% : 登録ユーザ数
- 15 ~ 25% : 1日のユニークなアクセスユーザ数
- 5% : 同時接続数のMAX値

## 「プロ野球チームをつくろう！ONLINE」で得られた数値

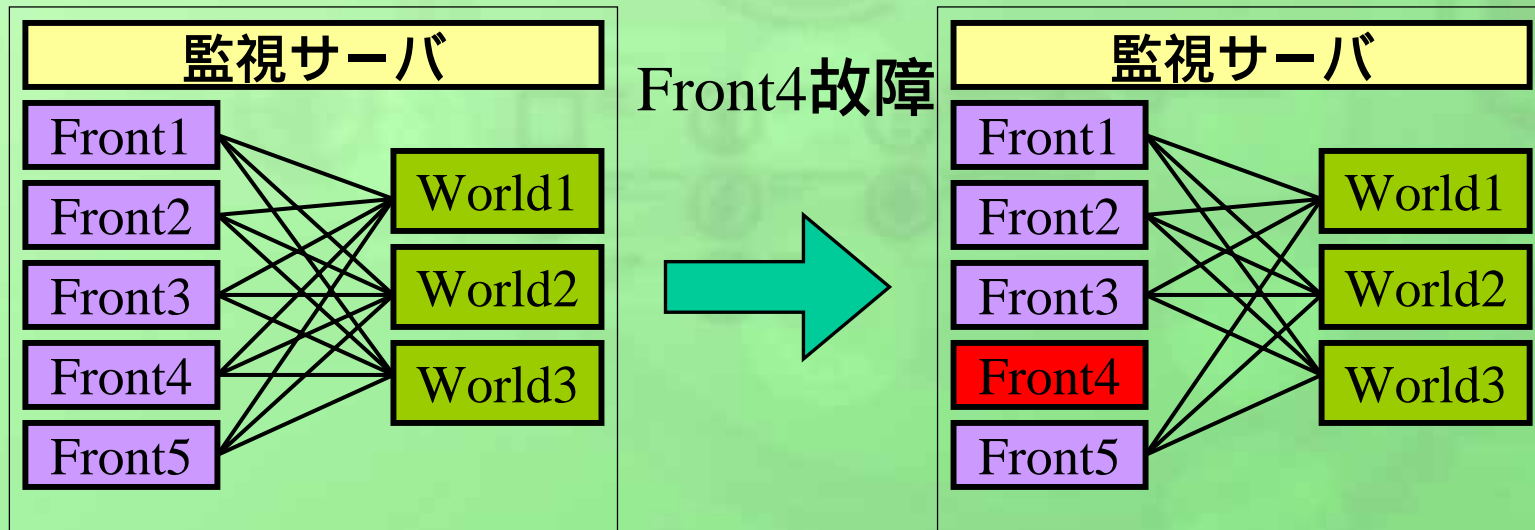
- 100% : 登録ユーザ数
- 60% : 1日のユニークなアクセスユーザ数
- 10% : 同時接続数のMAX値

## 第2章:サーバダウン対応

- 対応体制をどのように準備しておくのか  
「告知」「復帰」など..
- 24時間体制？(いつ問題が発生するのかわからない?)
- 勝手にサーバがやってくれるのが一番

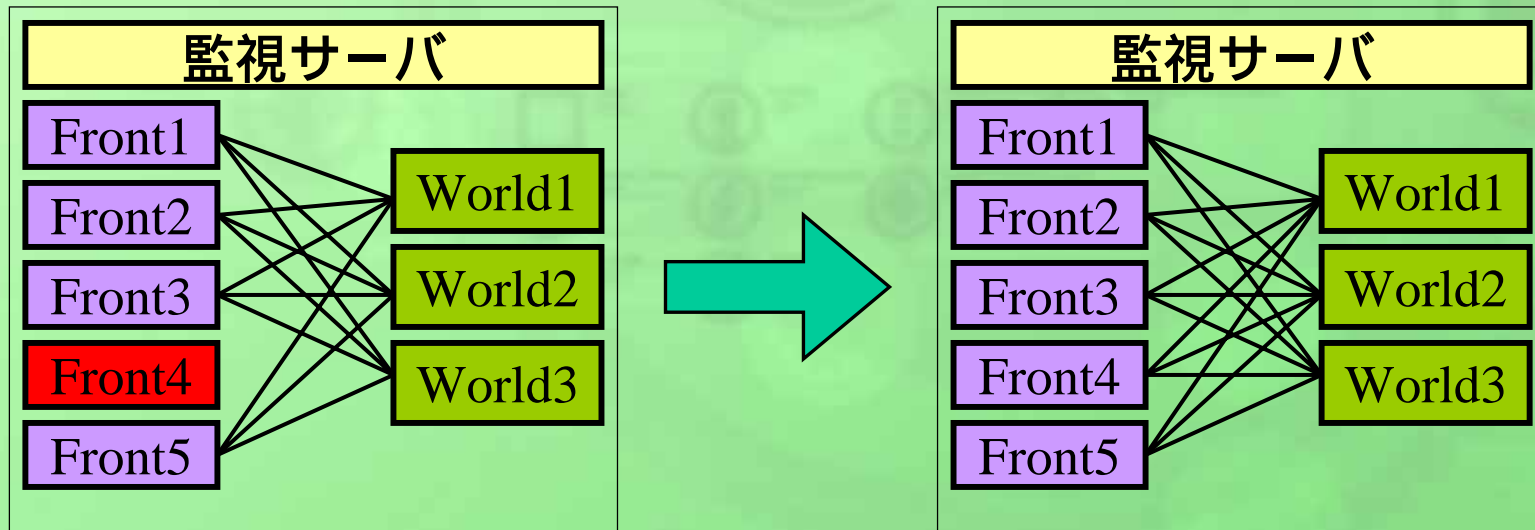
# FrontServerダウン

- Front4が故障、監視サーバがオートでFront4がないサーバ構成で再構築
- Front4にアクセスしていたユーザは通信切断、それ以外のユーザには影響が及ばない
- 監視サーバからメールにて運用チームに連絡



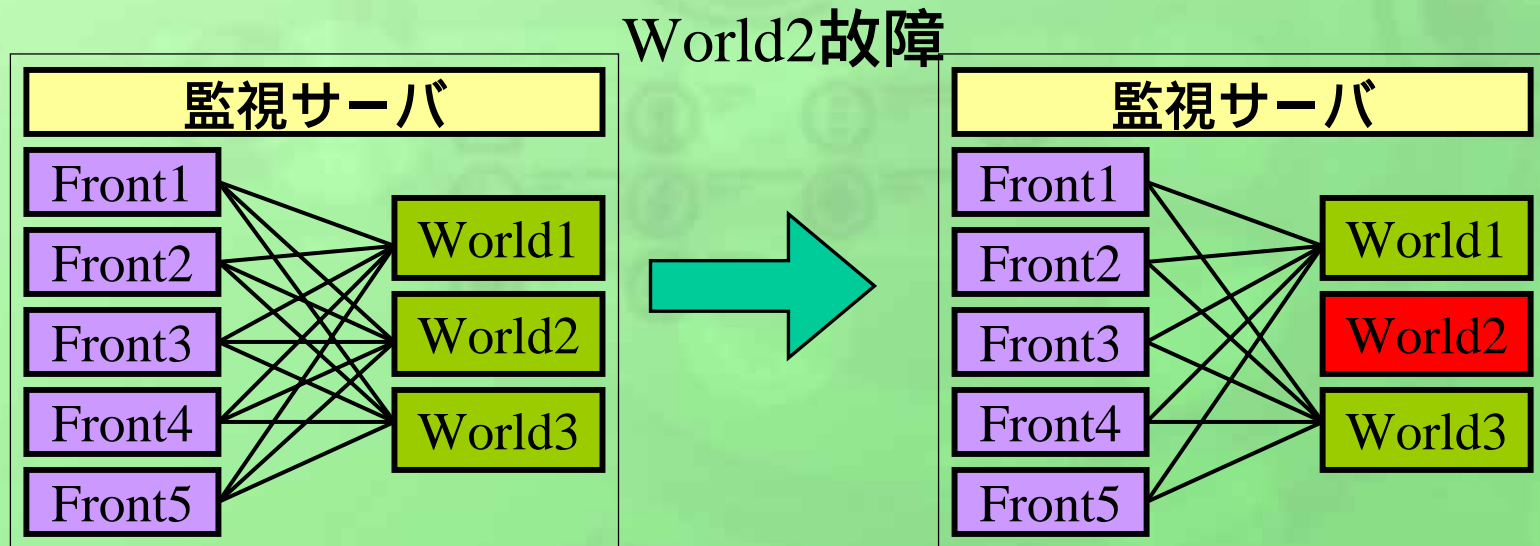
# FrontServer復帰

- サービス中でもFront4を起動すれば自動的にFront4を含んだ構成になります。



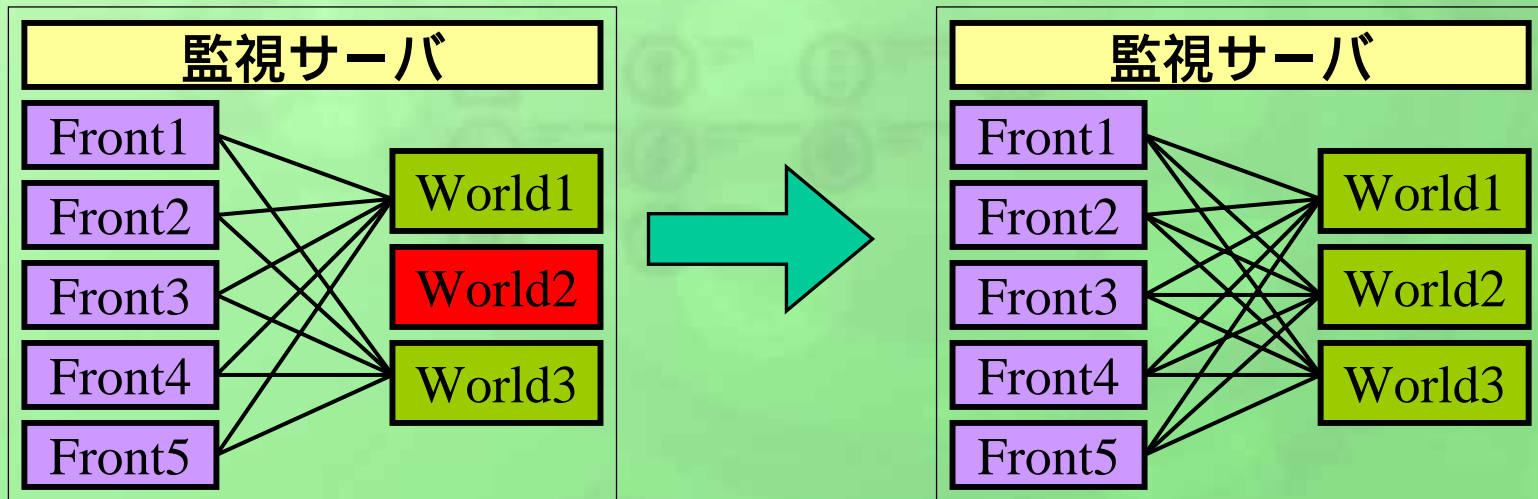
# WorldServerダウン

- World2が故障、監視サーバがオートでWorld2がないサーバ構成で再構築
- ユーザにはWorld2がアクセスできないよう自動で警告
- 監視サーバからメールにて運用チームに連絡



# WorldServer復帰

- サービス中でもWorld2を起動すれば自動的に再構成
- ほとんどの場合は安全を確認する為にメンテナンスを行い、起動実験をします。



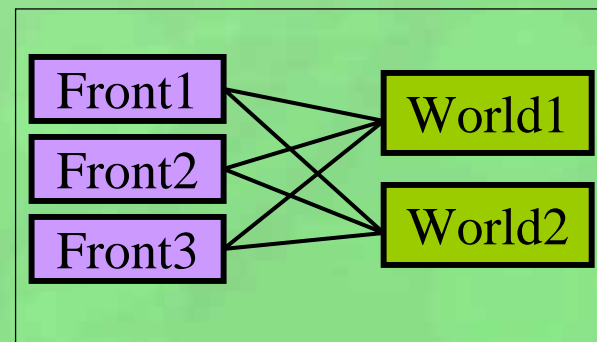
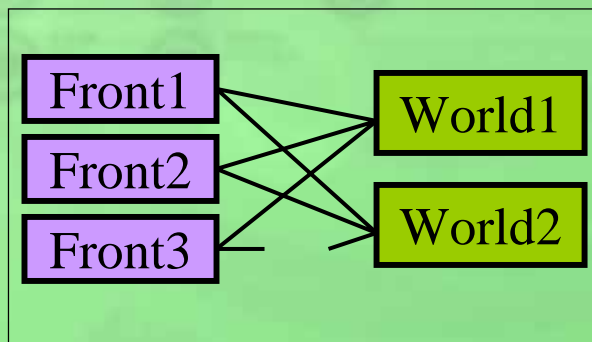
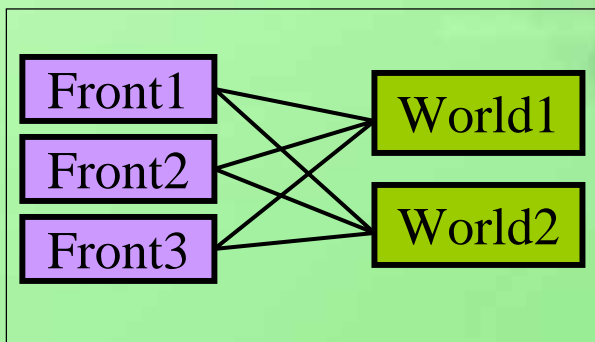
# サーバ間接続復帰

- FrontとWorldとの通信が切断されたとき勝手に再接続、何事もなかったようにServerは動作する。

Front3とWorld2が切断



勝手に再接続





# 「サーバダウン対応」まとめ

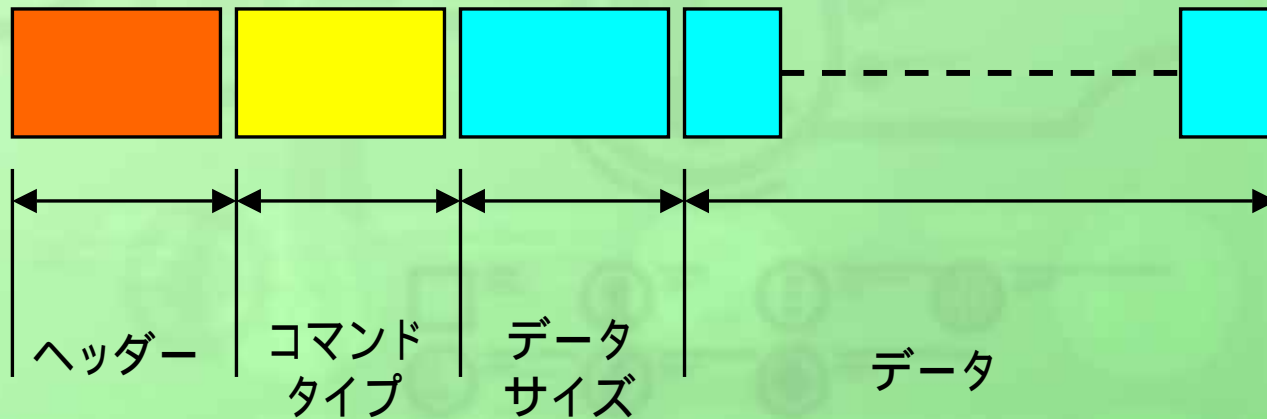
- 24時間監視し続けるのは、つらい。
- 自動で復帰してくれるのは便利である。
- 自動で復帰はするが、いざという時は手動が優先される必要がある。

# 第3章: パケット設計

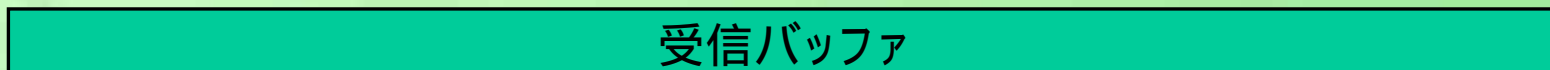
- 開発効率
- パケットのルールを作ろう
- サーバとクライアントのパケット

# パケット設計

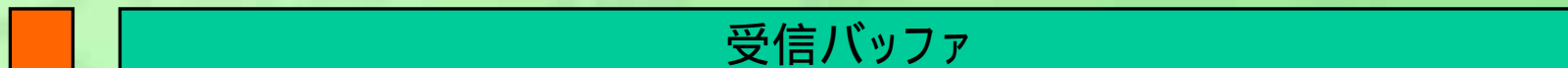
- パケットフォーマット



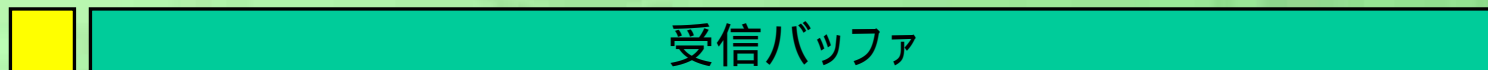
# パケット処理



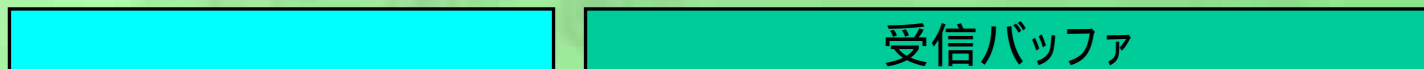
Step1:ヘッダー確認



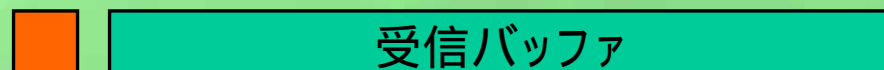
Step2:コマンドタイプ確認



Step3:データ受け取り



Step4:次のヘッダー確認



# 変更のあしかせ

- サーバとクライアントの開発が異なる。
- 規模が大きいと、なかなかライブラリを変えるタイミングが難しい。
- ちょっとしたテスト変更がしにくい。

# 実際の変更

ヘッダー  
部分

```
//          コマンド ヘッダー
struct HEADER{
    Sint16      head;
    Sint16      command_type;
    Sint16      command_size;
};
#define        HEADER_SIZE sizeof(HEADER)
```

受信側  
Client

```
//          サイクル情報通信。ワールド
struct COMMAND_TEAM_CYCLE : HEADER {
    Sint16      cycle;          //          サイクル
    Sint16      season;        //          シーズン
    Sint16      series;        //          シリーズ
    Sint16      worldNo;       //          ワールド番号
    COMMAND_TEAM_CYCLE(){
        head          = HEADER_STARTCODE;
        command_typ  = ZC_TEAM_CYCLE;
        command_size = sizeof( COMMAND_TEAM_CYCLE );
    }
};
```

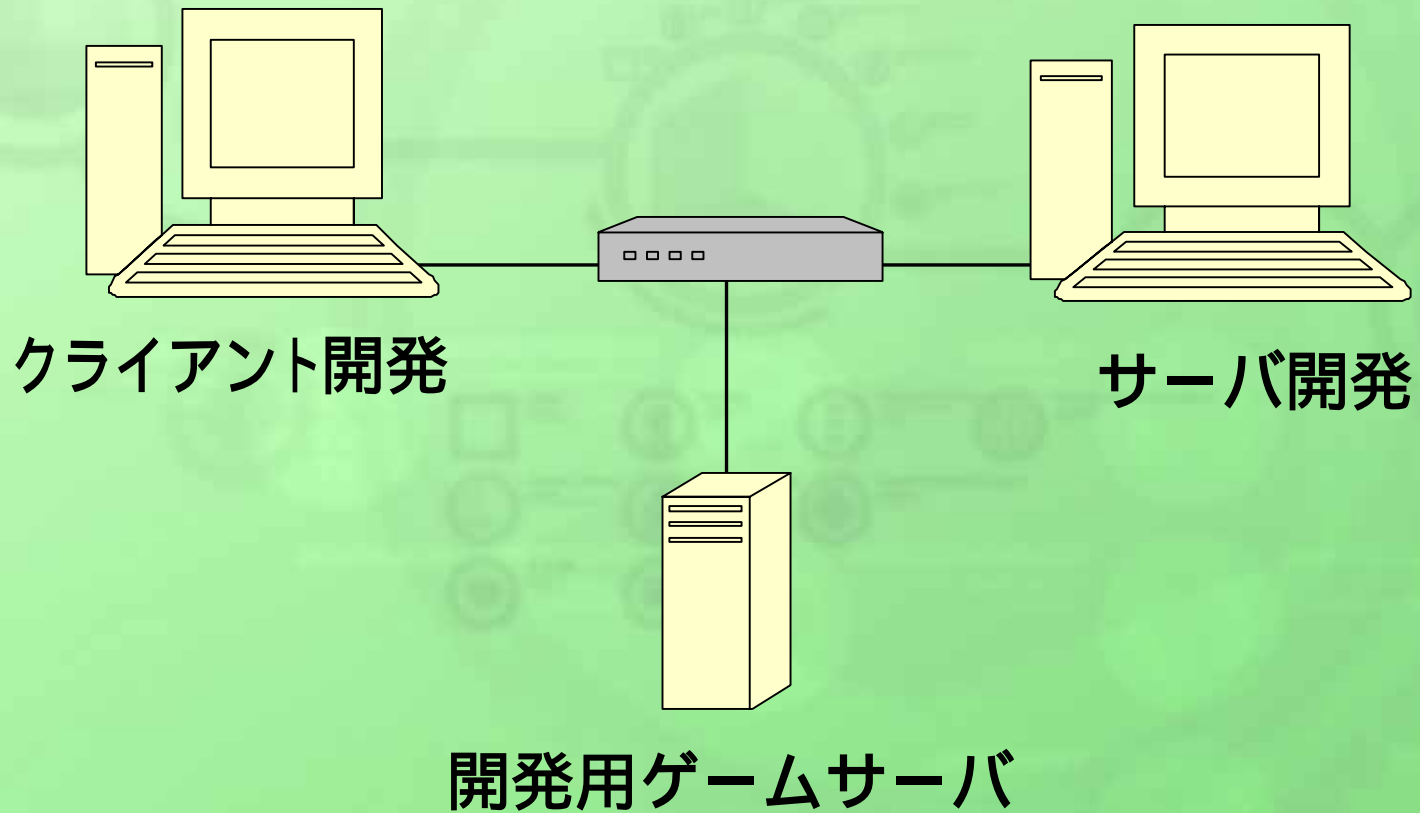
送信側  
Server

```
//          サイクル情報通信。ワールド
struct COMMAND_TEAM_CYCLE : HEADER {
    Sint16      cycle;          //          サイクル
    Sint16      season;        //          シーズン
    Sint16      series;        //          シリーズ
    Sint16      worldNo;       //          ワールド番号
    Sint16      userNo;        //          ユーザ番号
    COMMAND_TEAM_CYCLE(){
        head          = HEADER_STARTCODE;
        command_type  = ZC_TEAM_CYCLE;
        command_size  = sizeof( COMMAND_TEAM_CYCLE );
    }
};
```

データの追加



# 開発中

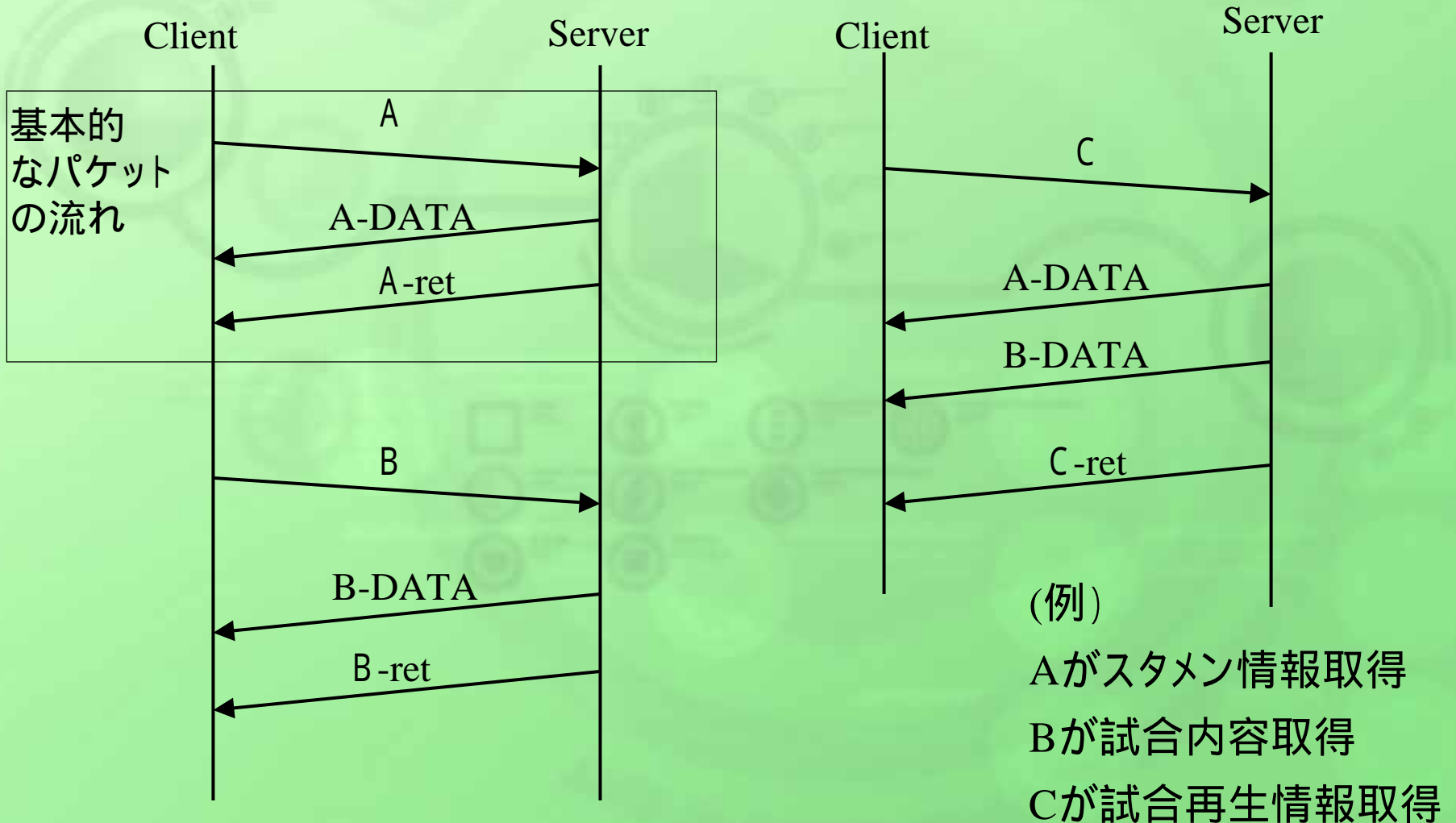




# パケットの再構築

- 仕様変更に対応する強いパケット設計
- パケットの送信、受信ルール

# ルール



# データの中身

- パケットのデータにどのようなデータを入れるのか？
- ハッキングされにくいデータ？

# (例) 選手データ選択



(NG)

「日高」を選択

(OK)

右から3番目 (相対的なデータの受け渡し)

# 「パケット設計」まとめ

- 規模が大きくなっても小回りできるように考えておく事は良かった。
- 開発中の変更を考慮した設計が必要。
- ハッキングされた事を考え、なるべく実データは送らない。

# 第4章：コミュニティのエンジン設計

- ドリームパスポート3
- ? ブラウザ
- PS2/PSP ブラウザ
- プロ野球チームをつくらう！ ONLINE

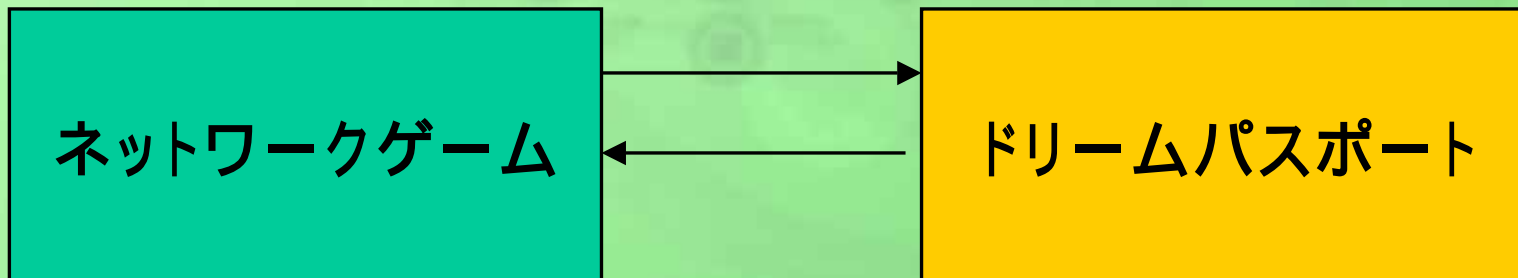
# ドリームパスポート3

- ゲーム中にそのままユーザ間の情報交換ができる媒体。



## ドリームパスポート3

- ネットワークゲームを終了させてからドリームパスポートに切り替える必要があった。
- ドリームキャストのメモリーが小さかったのでメモリーに乗りきらなかったなどの理由・・・





# ? ブラウザ

- ゲームの中に組み込む、ネットワークゲームとユーザ間の情報交換をスムーズにさせる。

ゲームアプリケーション

ブラウザ

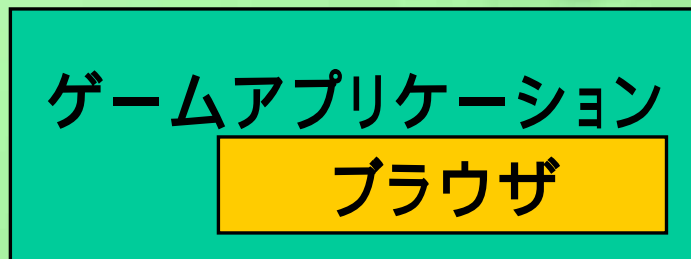
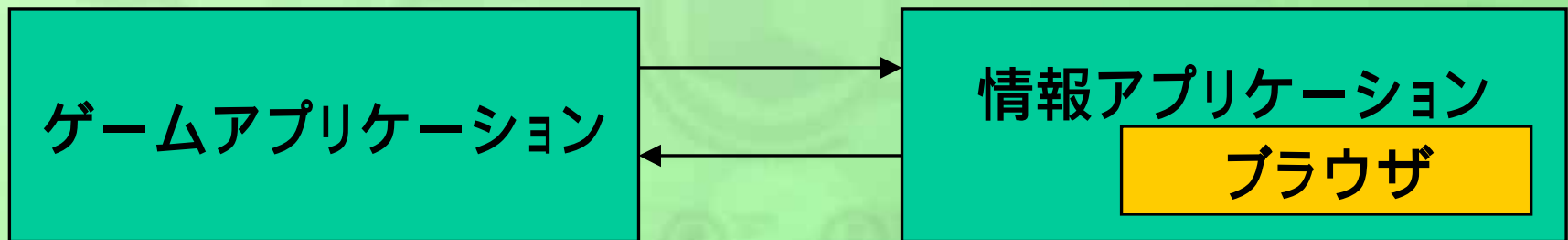
# PS2/PSPブラウザ

- SCEJの依頼でPSP用に  
外でネットワーク認証を  
する為にブラウザを開発



# PS2/PSPブラウザ

- ゲームに組み込むケースもあれば、ブラウザとして、切り替えて使う場合もあった。



(c) Sony Computer Entertainment Europe. All Rights Reserved.



(c) Sony Computer Entertainment Inc.

# プロ野球チームをつくらう！ ONLINE

- ゲームとのリンクを考えコミュニティに特化したプログラムにしてみる。

ゲームアプリケーション

コミュニティとしてのエンジン

コミュニティ 22364 pt

カテゴリ 参加コミュニティ

キーワード検索  検索

コミュニティカテゴリー一覧

<input type="checkbox"/> プロ野球	コミュニティ数: 18
<input type="checkbox"/> 選手	コミュニティ数: 9
<input type="checkbox"/> 野球つくONLINE	コミュニティ数: 9
<input type="checkbox"/> トレード	コミュニティ数: 39
<input type="checkbox"/> スキル合成	コミュニティ数: 3
<input type="checkbox"/> 地域	コミュニティ数: 22
<input type="checkbox"/> その他野球	コミュニティ数: 3
<input type="checkbox"/> その他	コミュニティ数: 10

閲覧 閉じる

# 「コミュニティーのエンジン設計」まとめ

- ネットワークゲームなので、ユーザとのコミュニティーの部分は大切。
- まだ、連動が足りない、今後検討する必要あり。

# 質問

# おつかれさまでした

「プロ野球チームをつくろう！ ONLINE2」

<http://www.yakyutsuku-online.com/>

「(株) ZENER NET WORKS」

<http://www.zener.co.jp/znw/index.html>