



**10年後の企業情報システムの
インフラはどうなるのか？
その時インフラエンジニアは？**

マジセミ代表 寺田雄一

質問

- インフラエンジニア/運用エンジニアの方
- アプリケーションエンジニアの方
- それ以外の方

■ 中学・高校時代



(出所) <http://www.sharpmz.org/mz-700/first700.htm>

(出所) http://www.cqpub.co.jp/column/books/2001a/34311tlcs900/img/list1_1.gif

(例1) 転送命令																																	
LD A, B	;Bレジスタの内容をAレジスタへコピー																																
LD A, 34H	;イミディエイト値34HをAレジスタへ																																
LD HL, 3456H	;イミディエイト値3456HをHLレジスタへ																																
LD (HL), 78H	;HLレジスタの内容でポイントするアドレスへ78Hを書き込み																																
EX DE, HL	;Z80の16ビット転送命令はあまり充実されていないため、交換命令EXまたはスタック経由で行う DE←HL																																
PUSH HL																																	
POP DE																																	
EQU \$;Z80には LEA命令がないので、実行番地を取得するにはアセンブラ疑似命令 EQUを使い、特殊定数\$を読む																																
(例2) ブロック処理命令																																	
LD HL, 1000H	;ブロック転送の転送元先頭アドレス																																
LD DE, 2000H	;ブロック転送の転送先頭アドレス																																
LD BC, 20	;ブロック転送数 20バイト																																
LDIR	;ブロック転送命令																																
(例3) 演算命令																																	
ADD A, 10H	;Aレジスタの内容に10Hを加算し、答えをAレジスタへ																																
ADD HL, BC	;HLレジスタ、BCレジスタの内容を加算し、答えをHLレジスタへ																																
SBC HL, DE	;HLレジスタの内容からDEレジスタの内容とキャリ・フラグの状態を引き、答をHLレジスタへ																																
CP B	;AレジスタとBレジスタの内容比較を行い、結果をフラグ・レジスタへ反影する 比較はA-Bを行いますレジスタの内容は変化しません																																
AND A, 10101111B	;Aレジスタの内容とビット・パターンとのANDをとり、答えはAレジスタへ																																
(例4) ビット操作命令																																	
BIT 3, A	<table border="0"> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> <tr><td colspan="8">Aレジスタ</td></tr> <tr><td colspan="8">↑</td></tr> <tr><td colspan="8">このビット状態を調べ結果は フラグへ反影する</td></tr> </table>	□	□	□	□	□	□	□	□	Aレジスタ								↑								このビット状態を調べ結果は フラグへ反影する							
□	□	□	□	□	□	□	□																										
Aレジスタ																																	
↑																																	
このビット状態を調べ結果は フラグへ反影する																																	
SET 1, (HL)	<table border="0"> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> <tr><td colspan="8">HLの内容でポイントするメモリ</td></tr> <tr><td colspan="8">↑</td></tr> <tr><td colspan="8">このビットをONする '1'にする</td></tr> </table>	□	□	□	□	□	□	□	□	HLの内容でポイントするメモリ								↑								このビットをONする '1'にする							
□	□	□	□	□	□	□	□																										
HLの内容でポイントするメモリ																																	
↑																																	
このビットをONする '1'にする																																	
RES 4, A	<table border="0"> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> <tr><td colspan="8">Aレジスタ</td></tr> <tr><td colspan="8">↑</td></tr> <tr><td colspan="8">このビットをOFFする '0'にする</td></tr> </table>	□	□	□	□	□	□	□	□	Aレジスタ								↑								このビットをOFFする '0'にする							
□	□	□	□	□	□	□	□																										
Aレジスタ																																	
↑																																	
このビットをOFFする '0'にする																																	

(例5) シフト、ローテート命令																																	
SLA A	<table border="0"> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> <tr><td colspan="8">← 0 Aレジスタの各ビットをシフトアップし、ビット'7'の値をキャリ・フラグへ、ビット0には'0'を入れる</td></tr> <tr><td colspan="8">↑</td></tr> <tr><td colspan="8">このビット状態を調べ結果は フラグへ反影する</td></tr> </table>	□	□	□	□	□	□	□	□	← 0 Aレジスタの各ビットをシフトアップし、ビット'7'の値をキャリ・フラグへ、ビット0には'0'を入れる								↑								このビット状態を調べ結果は フラグへ反影する							
□	□	□	□	□	□	□	□																										
← 0 Aレジスタの各ビットをシフトアップし、ビット'7'の値をキャリ・フラグへ、ビット0には'0'を入れる																																	
↑																																	
このビット状態を調べ結果は フラグへ反影する																																	
RLC A	<table border="0"> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> <tr><td colspan="8">← Aレジスタの各ビットをシフトアップ方向に回転させ、ビット'7'の値をキャリ・フラグへ入れる</td></tr> <tr><td colspan="8">↑</td></tr> <tr><td colspan="8">このビット状態を調べ結果は フラグへ反影する</td></tr> </table>	□	□	□	□	□	□	□	□	← Aレジスタの各ビットをシフトアップ方向に回転させ、ビット'7'の値をキャリ・フラグへ入れる								↑								このビット状態を調べ結果は フラグへ反影する							
□	□	□	□	□	□	□	□																										
← Aレジスタの各ビットをシフトアップ方向に回転させ、ビット'7'の値をキャリ・フラグへ入れる																																	
↑																																	
このビット状態を調べ結果は フラグへ反影する																																	
(例6) 分岐命令																																	
JP C, 1000H	;キャリ・フラグがあれば1000H番地へジャンプ、なければ次の命令を実行																																
RET Z	;ゼロ・フラグがあればサブ・ルーチン終了、なければ次の命令を実行																																
CALL 2000H	;2000Hのサブ・ルーチンへジャンプ、スタックへ戻り番地をメモ																																
	;JP,CALL命令のジャンプ先は通常ラベルで記述する																																
(例7) CPU制御命令																																	
EI	;この命令以後、割り込みが許可される																																
DI	;この命令以後、割り込みが禁止される																																
NOP	;なにもしない。Z80の場合1μsec程度時間待ちになる																																
HALT	;割り込み、またはRESETを受け付けるまでCPU動作が停止する																																





プロのダンサーになるか？
IT業界に就職するか？

もともとはインフラエンジニア

詳報 ● 大規模金融システムでLinux採用が相次



イー・トレード証券は7月, UFJグループ

高橋信頼

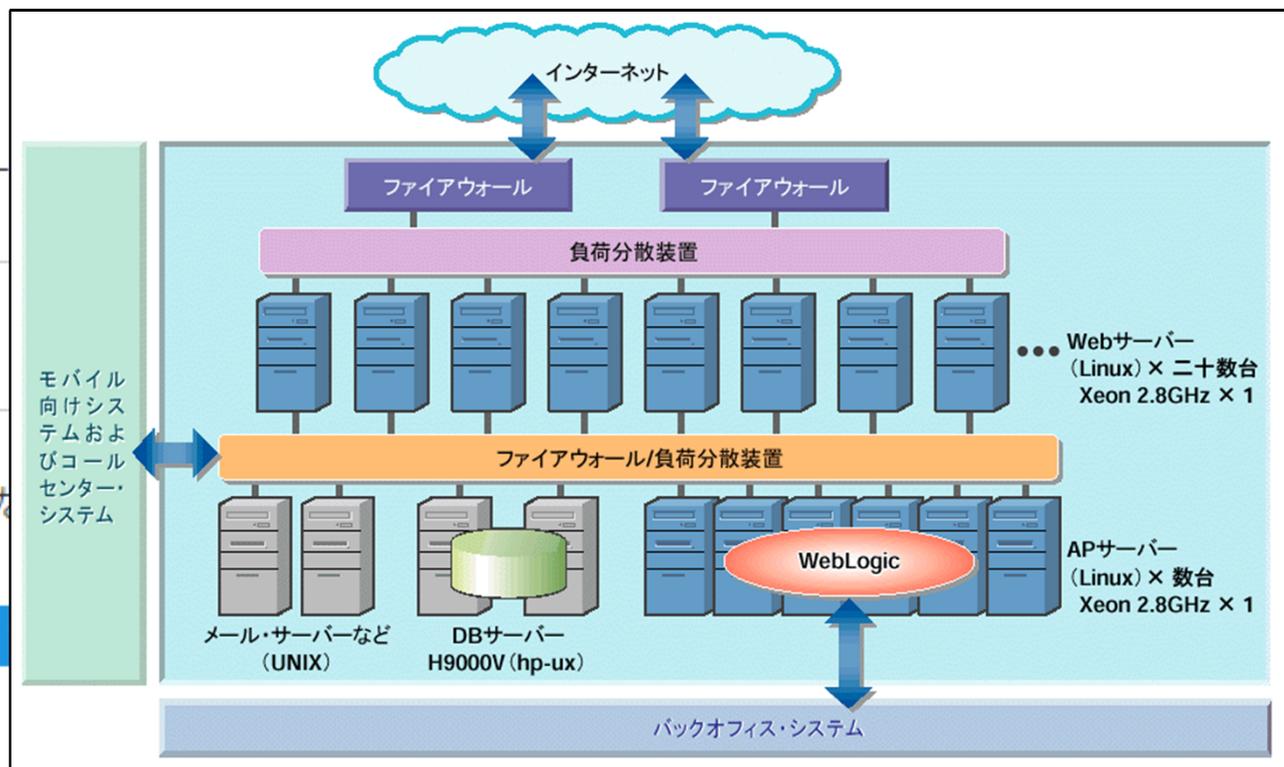
出典: 2003年9月号18ページ

(記事は執筆時の情報に基づいており、現在では異が)

シェア 0

B!ブックマーク 0

Pocket



要約 イー・トレード証券は7月から稼働したインターネット証券取引システムで, WebサーバーおよびAPサーバーとして三十数台のLinuxサーバーを使用している。UFJ銀行をはじめとするUFJグループは9月から順次稼働するグループ企業向け共通システムにLinuxを採用した。ともに価格性能比の向上を狙う。

2006年、OpenStandia

- オープンソース事業、スタート。
- 本当は、3年で分社化、もしくは退社して独立という予定だったが・・・
- 約7年かけて、二桁億の事業に育て、2013年に退社。
- 事業をやる中で、マーケティングの重要性と、IT業界における意識の低さに気づく。



(出所)<http://news.mynavi.jp/news/2008/05/28/020/>

2014年、マジセミ 立上げ

- IT業界を変えるには、
中小IT企業がもっと活躍する必要がある。
- 中小IT企業の知名度を挙げて、
ユーザ企業と直接取引ができる
ようにしたい。
- そのためのマーケティング
支援サービスを開始。



代表取締役所長 寺田雄一のプロフィール



寺田雄一
(てらだゆういち
/yuichi terada)
1969年8月21日生
まれ 48才



- 2003年、野村総合研究所に大手で日本初のOSS専門組織オープンソース・ソリューションセンターを設立。
- 2006年、オープンソース・ワストップサービス「OpenStandia（オープンスタンディア）」事業を創業。
- 2013年まで野村総合研究所において、オープンソース事業の責任者を務める。
- オープンソースビジネス推進協議会（OBCI）、OpenAMコンソーシアムを設立（発起人）。同会の理事、会長や、NPO法人日本ADempiereの会の理事などを歴任。
- 情報サービス産業白書2012年版～2015年版「オープンソースソフトウェアの動向」執筆を担当。
- 「エンジニアの楽園」を目指し「開発合宿（ハッカソン）」などをプロデュースする活動も行う。
- 2013年、IT業界の構造改革を実現するため、野村総合研究所を退社し、株式会社オープンソース活用研究所を設立。
- 著書に、オープンソースビジネスを解説した「オープンソースがよ〜くわかる本(秀和システム)」

世界にコンピュータは「5つ」あれば足りる

Sun Microsystems の CTO グレッグ・パパドポラス

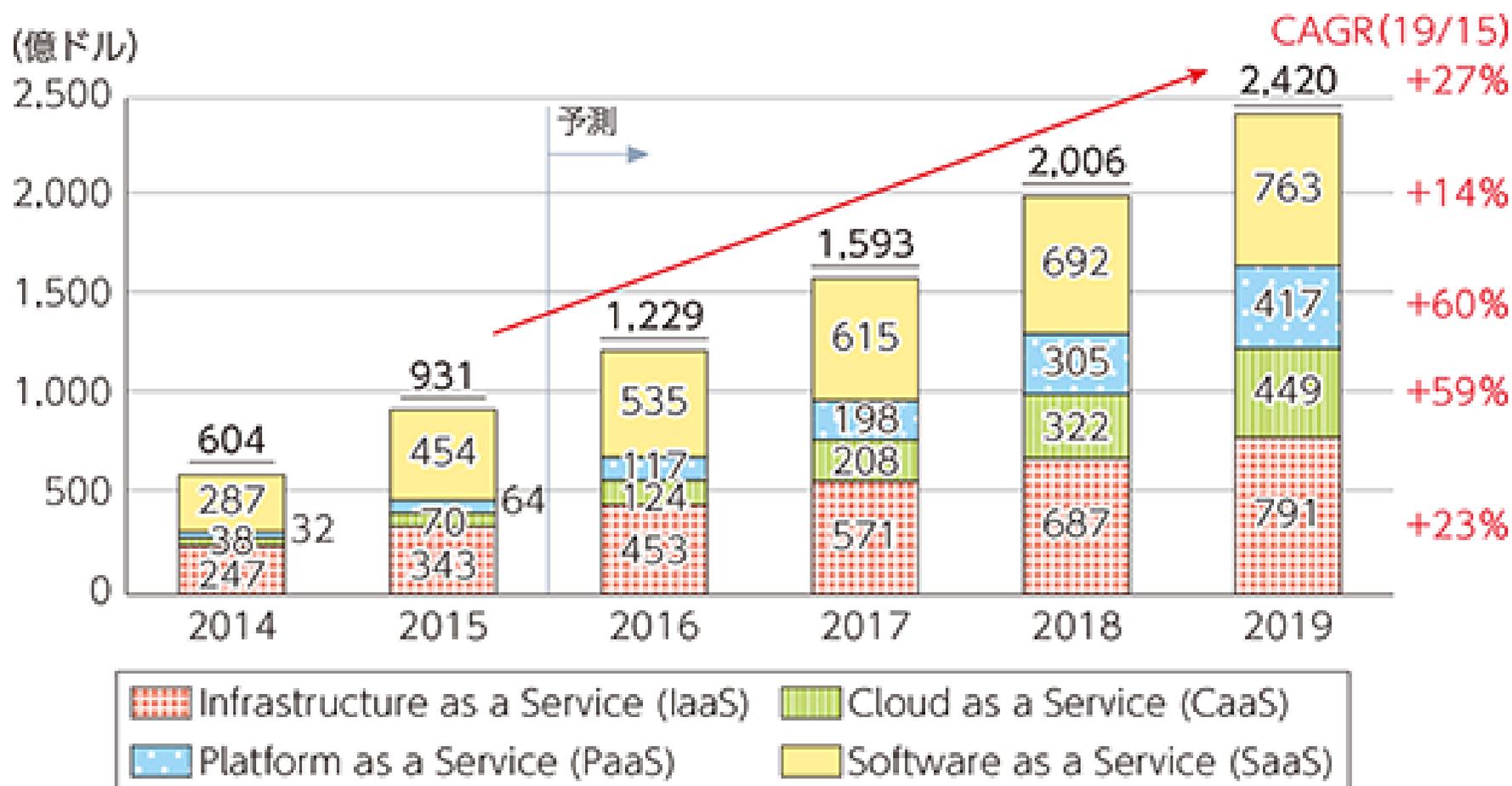
Amazon、Google、IBM、Microsoft (Azure) 、 …

現時点で、Amazon、Google、IBM、Microsoft
(Azure) でクラウドのシェア50%以上

2019年までに

世界のコンピュータートラフィックのうち83%がクラウド上を流通し、

処理量の約80%がクラウド上で処理される



AWSサービス開始

2007年

三菱UFJフィナンシャル
グループがAWSの活用を
開始

2017年

2027年

現在

2020年
金融系、省庁・自治体以外
は、クラウドが主流に

2023年
金融系でも
クラウドが主流に

サーバーレス
アーキテク
チャーが主流に

政府・自治体でも
クラウドが主流に、
オンプレが消滅

インフラエンジニアの価値

- 難しいインフラの設定ができる
 - 例) Oracleの設定、チューニングができる
- 制約が多い中で、最適な構成を設計できる
 - 例) C/S型システムにおいて、ネットワーク帯域が細い中で、SQLを流さないようにする
 - 例) サーバーの増設が難しい中で、ピーク時のトラフィックを予想し、サーバーをサイジングする
- 技術変化の激しい中、「勝ちパターン」を見つけ、標準化する
 - 例) Web3階層

ITインフラが自動化される中で、
インフラエンジニアの業務も
「自動化」

それが「進歩」

■テレコミュニケーション

- モールス電信機・・・言葉に変換する人がいた
- 昔の電話・・・交換手がいた
- 今の電話・・・

■システム開発

- 昔のシステム開発
 - ・ディスプレイにドット単位の表示制御、テープデバイスへの書き込み制御、アセンブラでレジスタを使って、、、
- 今のシステム開発

だから、
インフラエンジニアは
キャリアチェンジが必要

私のキャリアチェンジ

■ 2000～2005年ころ

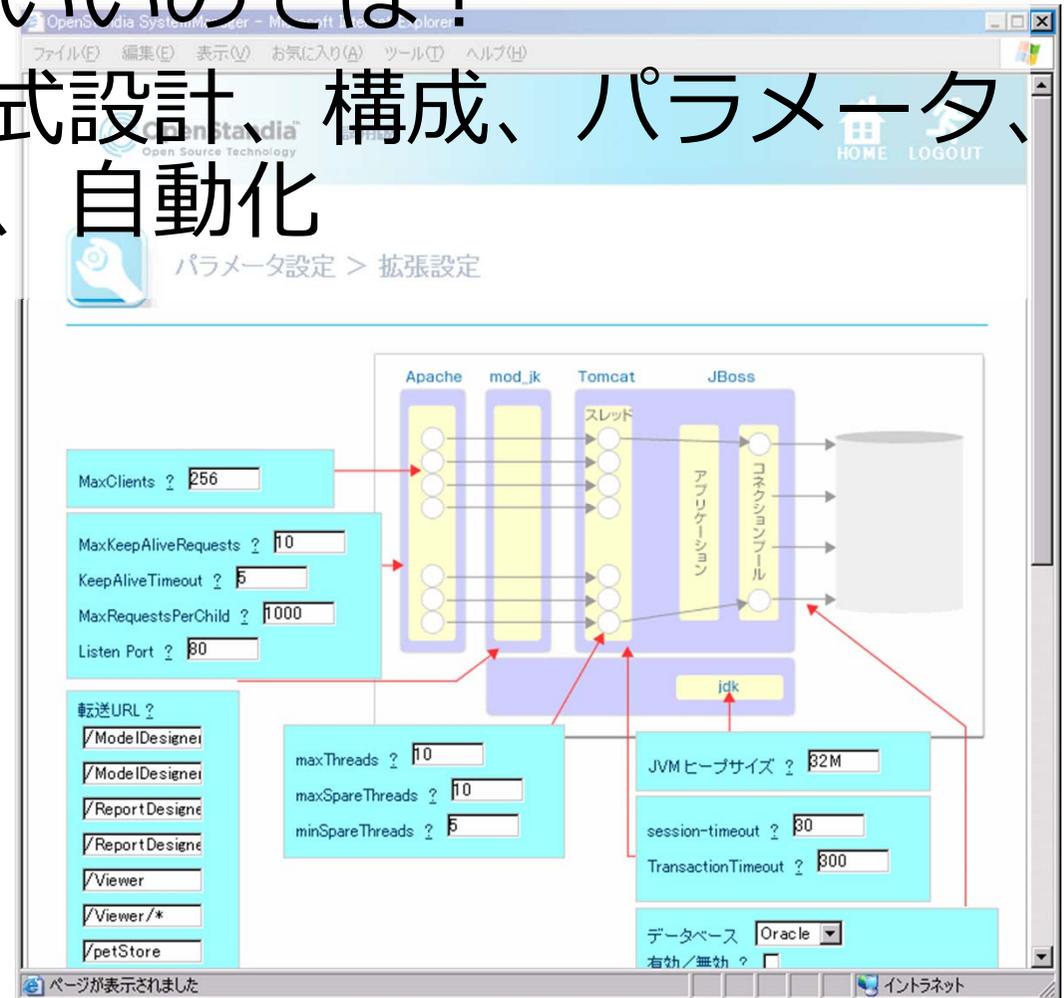
- 全部Webシステムでいいのでは？

- どのシステムでも方式設計、構成、パラメータ、変わらない→標準化、自動化

■ インフラエンジニアは不要になる、と確信

■ 2005年

- 起業（社内ベンチャー）



インフラエンジニアからの キャリアチェンジの一例

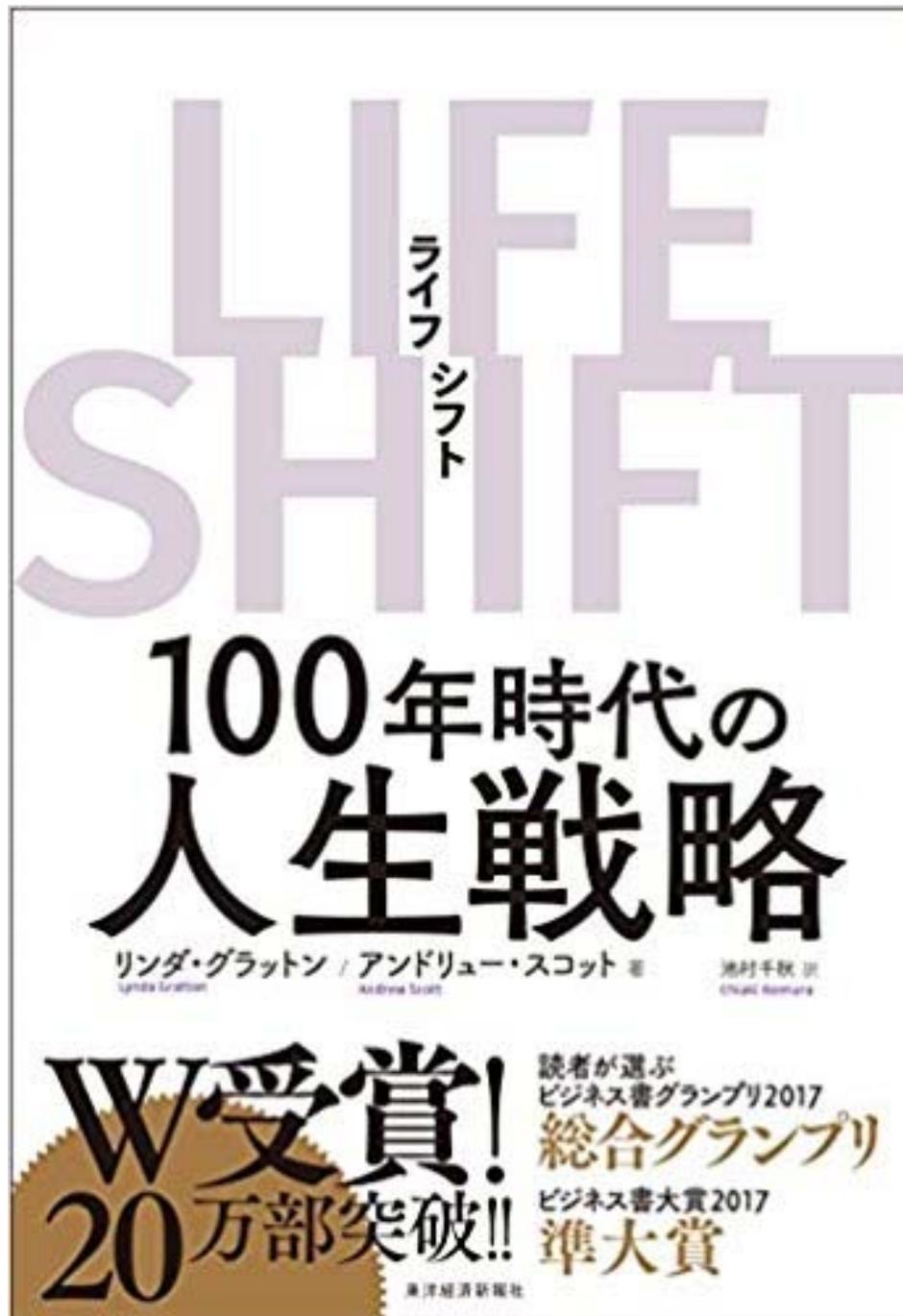
- アプリケーションエンジニア
- セキュリティエンジニア
- ITアーキテクト
 - クラウド、APIエコノミー、ビッグデータ、AI、IoT、...
- データアナリスト
 - システムのログを**全体感**をもって分析
 - マーケティングに活かす
 - ・ 「ここをこう変えれば、パフォーマンスが5%改善して、売上が10%増加します！」

インフラエンジニア
の強みは“全体感”

キャリアチェンジは
いつするべきか？

準備は、「今」

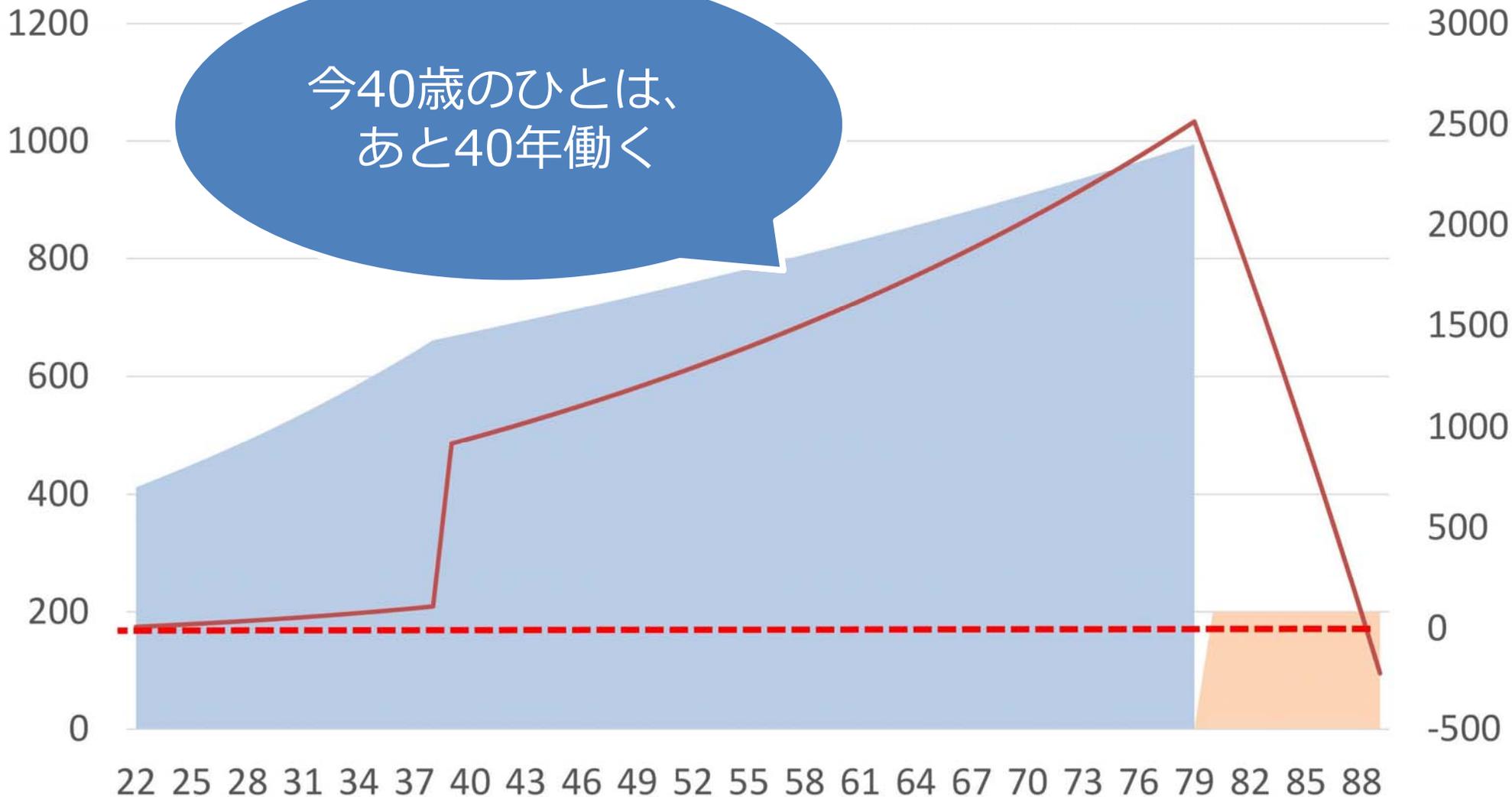
しかも、あなたは今後、
複数回のキャリアチェンジが必要



80歳リタイアのモデル

所得・年金

貯蓄



公的年金 所得 貯蓄

必要なこと

自らを変化させる力、スキル

それを身につけるには？

変化に身を置くこと

例えば、
異動、副業、転職、起業、
外部のコミュニティ、
...

