

# NTPとうるう秒

---

JANOG42

NTPサーバーの構築とうるう秒対応

# 自己紹介

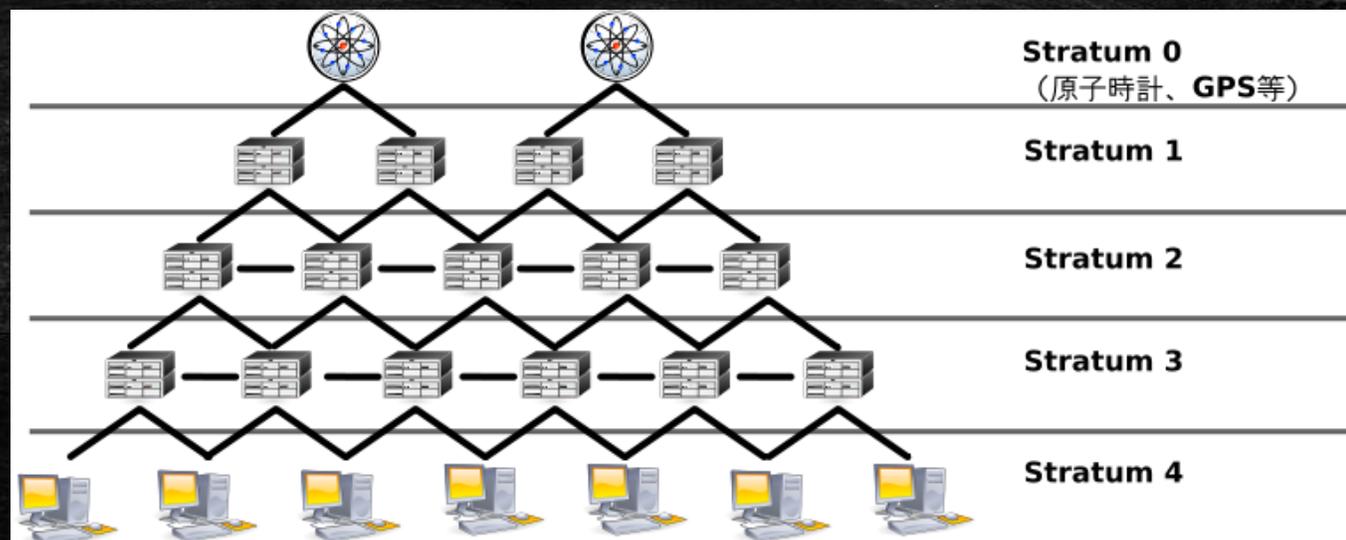
---

- 名前：九龍真乙 (PN)
- 所属：自宅ラック勉強会
- お仕事：Sierでのテクニカルサポート (AWS,Zabbix,ネットワーク,サーバーインフラ)
- 好きなプロトコル:NTP,SNMP
- Twitter:@qryuu
- Facebook
  - <https://www.facebook.com/qryuu>
- github
  - <https://github.com/qryuu>
- cookpad
  - <https://cookpad.com/kitchen/4142562>



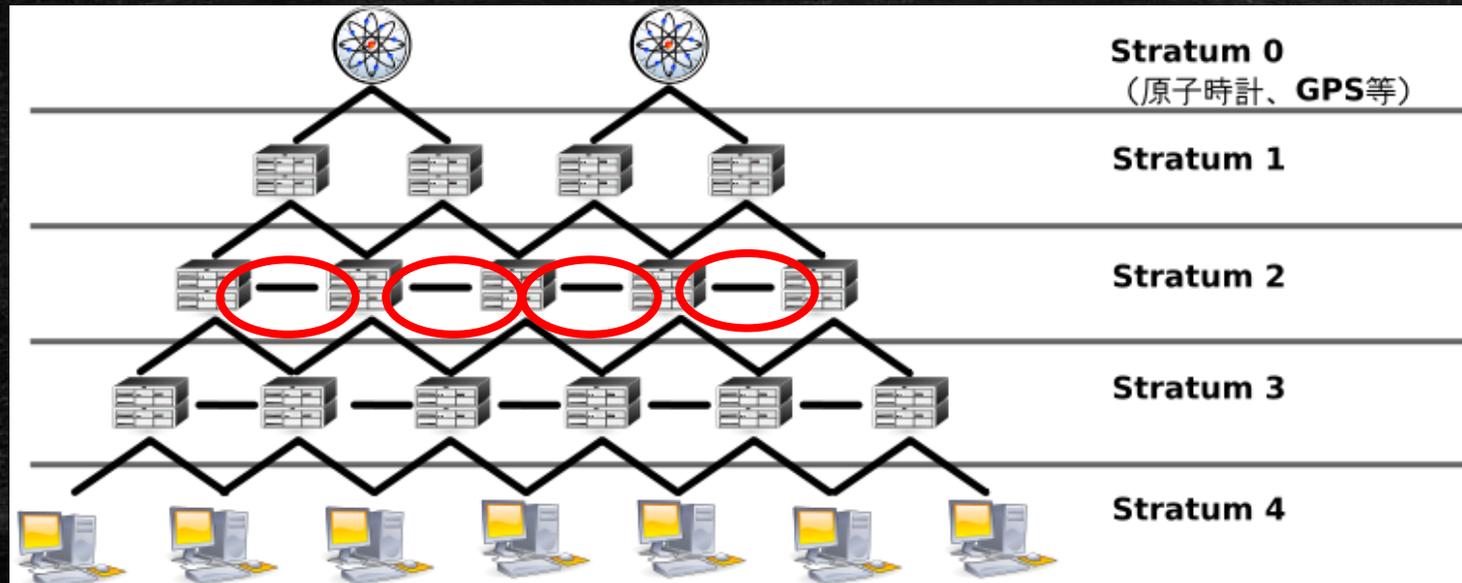
# NTPとは

- システム間の時刻同期に利用
- Stratumという階層構造
- 世界的にはNTP Pool Projectが有名ですが
- 日本ではNICTによる変態的(褒め言葉)な性能のntp.nict.jpが利用可能



# NTPの設定

- NTPサーバーとNTPクライアント
  - NTP Peerってご存知ですか？
- ntpdには統計誤差を補正するためにドリフトファイルが存在します。
  - 時刻校正をするに、同一Stratumのサーバー同士でpeerを設定します。



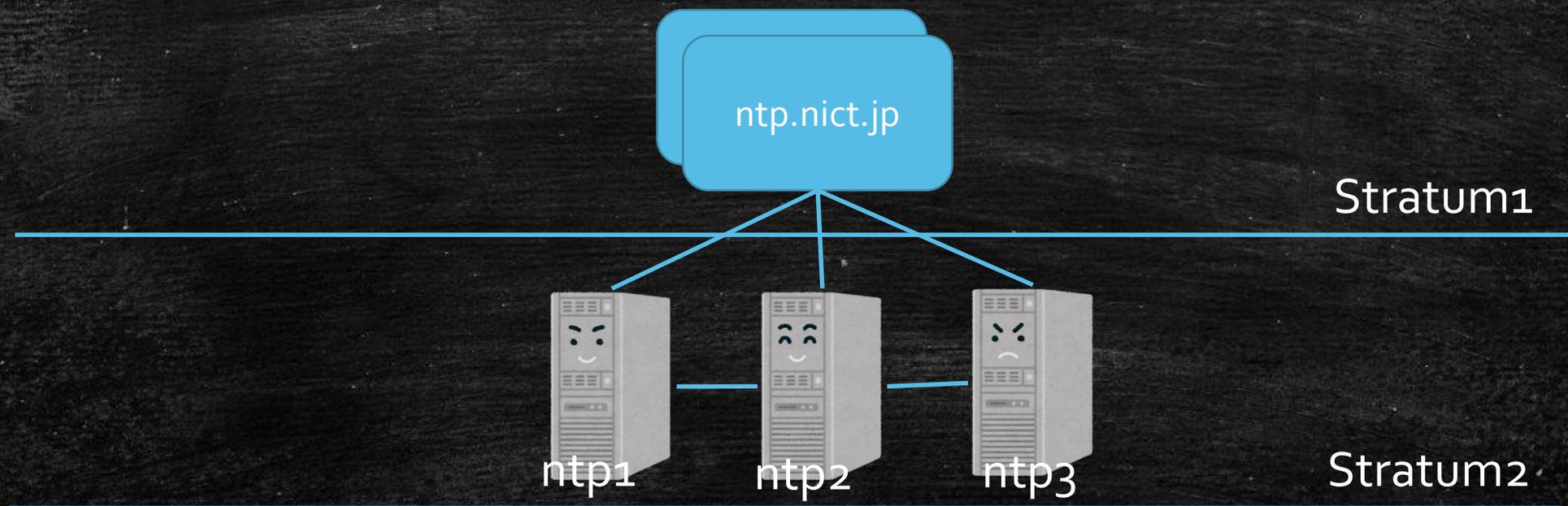
# 「家は3回建てろ」、「NTPは3台建てろ」

---

- 2台では統計誤差を判断出来ないので、3台のNTPサーバでpeerを組むことにより統計誤差の補正ができます。
- 三賢者

# 個人的推奨構成

- ※中小規模向きです（大規模は自前で原子時計とかGPS同期とかのStratum0を持っているはず）



NTP Peer設定とDNSラウンドロビン

## うるう秒とは

- 原子時計により決定される国際原子時TAIをベースとする協定世界時UTCに対して、天文時(世界時)UT1との差が0.9秒以内に収まるように挿入される、±1秒の事を言います。



# UT1とは

---

- UT1とは
- 地球の自転により計測された1日の長さから求められた $1/86400$ 日 = 1秒とする時刻系

# 地球の自転速度とUT1

---

- 地球の自転速度は、大気動向や海流、潮汐力、惑星運動の相互作用などにより、一定ではありません。  
(巨大地震などでも自転速度は変わります。)
- この変化する自転速度から定められるのが時刻がUT1です。
- UT1は「正午に太陽が真南に昇る」など生活と時刻意識を合わせる上で重要です。

# 時計の歴史

---

- 時計 = 時を計る機器は当初は宗教儀式のために、現在は物理測定のために進化してきました。

# 時計の歴史

---

- 宗教的な儀式のために、古来より時計は作られてきました。
  - 日時計



# 時計の歴史

---

- 宗教的な儀式のために、古来より時計は作られてきました。
  - 日時計
  - 火時計



# 時計の歴史

- 宗教的な儀式のために、古来より時計は作られてきました。
  - 日時計
  - 火時計
  - 水時計



# 時計の歴史

---

- 振り子時計

- より正確な時計を求めて「振り子の等時性」により1秒を正確に刻むようになりました。
- 機械式時計では「脱進機」により重力方向に依存せずに1秒を刻むようになりました。



# 時計の歴史

---

- クォーツ時計

- 水晶の結晶に交流電圧をかけると一定の周波数で振動する。(水晶振動子)
- この一定の振動をカウントすることで1秒を決定する
- 通常は $32,768\text{Hz} = 32,768$ 回振動で1秒



# 原子時計

- 時間を正確に刻むために、セシウム原子の基底状態のエネルギー状態から1秒を決定 これが1秒の定義にもなっている。
  - 9,192,631,770回振動したら1秒



# 協定世界時UTCとは

---

- 地球の自転により変動する天文時UT1対して、国際原子時TAIはセシウム原子時計を含む50カ国以上、300個以上の原子時計により維持されているため、変動しません。
- このため、天文時UT1とTAIではズレが生じるため、TAIとUT1のズレをうるう秒を用いて補正されたUTCが協定世界時として利用されています。

# 時間の取り扱いに関わる団体

---

- 国際天文学連合 (IAU) ※天文学術系
- 国際地球回転・基準系事業 (IERS) ※天文学術系
- 国際無線通信諮問委員会 (CCIR、現ITU-R) ※IT技術系はここだけ
- など

# うるう秒の挿入ルール 今回の議題1はここ！！

---

- うるう秒の挿入は
- 第一優先
- 6月末日もしくは12月末日(UTC)
  - アメリカだとクリスマスホリデー・ニューイヤーカウントダウン (UTC-8)
  - 日本だとお正月元旦(UTC+9)
- 第二優先
- 3月末日もしくは9月末日(UTC)となっています
  - アメリカだと大学入学前日 (UTC-8)
  - 日本だと入社式(UTC+9)

## 挿入の周期

---

- 過去26回のうるう秒挿入が実施されており、直近の実施では
- 2008年1月1日
- 2012年7月1日
- 2015年7月1日
- 2017年1月1日 <New
- となります。

# うるう秒挿入とITシステム

---

- うるう秒では1秒挿入されるため、システムによっては通常存在しない23:59:60(JSTでは08:59:60)が存在する場合があるためプログラムが想定されていない秒数として正常に処理出来ない場合があります。
- 2015年7月のうるう秒ではLinuxカーネルのバグでシステムハングやサーバ高負荷が発生

# OSとうるう秒

---

- Windows

- Windowsはうるう秒を考慮して動作しません。このため23:59:60秒は表示されず、通常通り0:00:00秒として動作します。  
NTP同期の際にズレがあれば、その時点でSTEP方式で同期します。

- RHEL

- ntpdが動作していない場合は、うるう秒に関知しませんので、UTCと1秒ズレた状態で動作し続けます
- ntpd動作している場合、うるう秒を処理します。  
カーネルバージョンによってはバグが報告されていますので、ベンダーサイトを  
確認してください。

# うるう秒とDBMS

---

- MySQL

- 59:59を複数回繰り返すことで59:60の発生を抑止します。
- 時刻をキーとしているシステムでは値の重複に注意が必要です。

- PostgreSQL

- 動作するOS上でうるう秒(59:60)が実装されている場合
- extract関数のsecondフィールドに60秒が現れる可能性があります。
- DBを利用するアプリケーションで"60"秒を正常に処理出来るか確認が必要です。

# うるう秒のLeap Smearing (分散挿入)

---

- 1秒を数時間から1日掛けて分散して挿入する方法です。
- [RFC5905](#): NTPv4で定義され、[利用されています](#)。
  - 実装としてはntpdバージョン4.2.8.p3および4.3.47でコンパイルオプションとして追加されました。
- ただし、どれくらいの期間に分けて挿入するかは規定されていません。
  - 実績は2時間から24時間の間で報告されています。

# うるう秒のLeap Smearing (分散挿入)

---

- メジャーな実施例
  - Amazon AWS: マネージドサービスおよびTime sync Serviceで24時間の分散挿入を実施しています。
  - [Google Public NTP](#): [24時間の分散挿入を実施しています。](#)
  - セイコー タイムスサーバー: [アジャストモード](#)として、2時間掛けて分散挿入します。
- 実装やメーカーによってLeap Smearingの粒度が異なります。
  - そもそも、NTPベストプラクティスでは、SmearingサーバーとNon Smearingサーバーは同期しないため、パブリックフェースのNTPサーバーでは使用してはならないとされています。( [Google Public NTP](#) へっ)

# うるう秒のLeap Smearing (分散挿入)

## 今回の議題2はここ！！

---

- Leap Smearingとどう向き合うべきか
  - RFCで間隔までふくめて標準化？
  - あくまでも障害回避のためのローカル環境用実装？

# NTPとエンジニアの関係

---

- 2036年問題
  - NTPプロトコルのビットオーバーフロー
  - 2038年問題のUNIX TIMEオーバーフローよりも2年先に来ます。
- NTPを拡張するか  
(SNTPv4:RFC 4330では最上位ビットで期間遷移して対処)
- PTPを拡張するか  
(Precision Time ProtocolはあくまでもLANを想定しており、インターネット同期用ではない)
- そもそも時刻同期不要とする
- チップ型原子時計を登載

## 時間は面白い

---

- 次回はぜひNICTさんやセイコーさんの話が聞きたい。