

PF 及び cERL における機器監視システムの開発

DEVELOPMENT OF EQUIPMENT MONITORING SYSTEM FOR PF AND CERL

亀田吉郎^{#, A)}, 帯名崇^{B)}

Yoshiro Kameta^{#, A)}, Takashi Obina^{B)}

^{A)} East Japan Institute of Technology Co., Ltd.

^{B)} High Energy Accelerator Research Organization (KEK)

Abstract

A reliable equipment monitoring system is a key to realize smooth and stable accelerator operation. In addition to monitoring various kinds of hardware faults, it is also a necessary item to check whether the software is working correctly. Therefore, we developed a monitoring system such as host monitoring, IOC monitoring, difference monitoring and process monitoring. Besides, to improve operator's convenience, communication with the existing EPICS CSS Alarm and the co-operation with the e-mail notification system are essential. In this presentation, we will report details of the system, operational situation, and future issues.

1. はじめに

加速器を安全かつ円滑に制御するには、様々な機器を遠隔監視して異常発生時に通知することが重要である。加速器を構成する多種多様なハードウェアを監視することの他にもソフトウェアが正常に動作しているかどうかを確認することも必要な項目である。計算機のハードウェア情報に関しては一般的な監視ソフトウェアで SNMP や agent による情報の取得は比較的容易である。しかし、ソフトウェア面では“入出力コントローラ (Input/Output Controller) が正常に動作しているか”や“設定値とモニタ値の差分を監視する”といった事象を監視するには判別する条件が対象によって異なるため個別に対応することが必要となり、一般のソフトウェアでは困難である。

そこで本件ではこれらの問題を解決するために開発目標を以下に定めた。

- EPICS[1]に対応すること。
- ホスト監視では ping による応答で対象機器が動作しているか監視を行う。
- IOC 監視では各 IOC にタイムスタンプを設け、常時更新しているか監視を行う。
- 差分監視では 2 つのレコードを監視し差が出ないか監視を行う。
- プロセス監視では指定されたサーバでプロセスが動作しているか監視を行う。
- これらの監視機能は独立して動作する。また、容易に変更や追加が可能であること。
- 利便性を高めるため既存の CSS Alarm[2, 3]と連携し異常時には通知を行う。
- メール通知機能と連携し異常時には指定された宛先にメールで通知を行う。
- アラームを発報する条件を個別に定義可能とする。
- 監視状態を web 経由で確認可能であること。

これらを踏まえたシステムの概要を Fig. 1 に示す。

本発表ではシステムに関する詳細及び運用状況、今後の課題について報告する。

[#] hig-kame@post.kek.jp

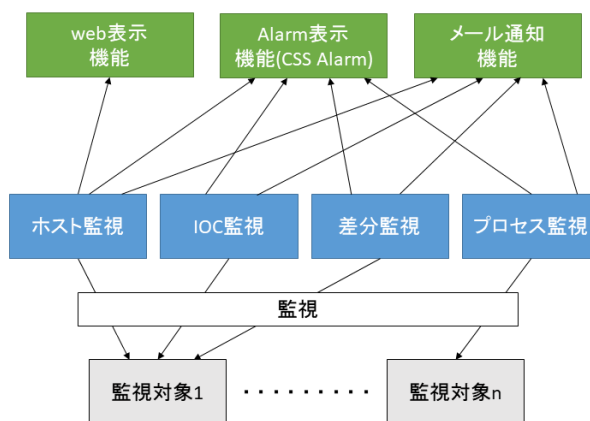


Figure 1: System overview.

2. 監視システム

2.1 ホスト監視

ホスト監視は定義された複数機器の IP アドレスに対して ping を送信し、応答を解析してネットワークの疎通を定期的に確認する機能である。これは ping によるコマンドの送信と応答解析を行う python のホスト監視スクリプト、GUI 表示や CSS Alarm と連携するためのホスト監視 IOC とに分かれている。スクリプトと IOC は Python CA[4] でデータの受け渡しを行う構成になっている。また、監視スクリプトでは複数のホストに ping コマンドを同時に送信する fping を使用している。これは数百ものホストに対してシーケンシャルに送信しては時間がかかりすぎてしまい短時間で定期的に行うことが不可能になってしまうからである。これらを踏まえてホスト監視の構成図を Fig. 2 に示す。

定義ファイルには Alarm で発報する条件を指定することができる。例えば加速器がメンテナンス状態なのか運転状態なのかを判断し運転状態の時にのみ発報するという指定も可能になる。こういったことにより不都合な発報を避けることができ、利便性を上げることができる。

ホストの状態を一覧で確認できる GUI について Fig. 3 に示す。

GUI ではホスト名・IP アドレス・Status・Alarm を一覧で確認することが可能な仕様とした。また、ホスト数が多いためホスト名・Status・Alarm によってフィルタリングする機能も実装し利便性を高めている。

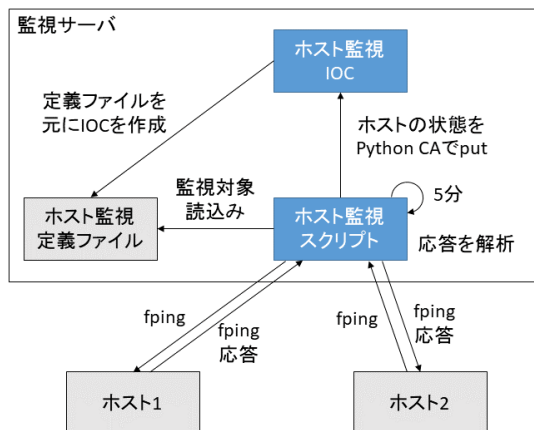


Figure 2: Configuration diagram of host monitoring.

No	HostName	IP Address	Status	Alarm
001	erlechw1.cerl.kek.jp	172.28.64.1	RUN	NO ALARM
002	zabbix.cerl.kek.jp	172.28.64.10	RUN	NO ALARM
003	erlwl-ap1.cerl.kek.jp	172.28.64.90	RUN	NO ALARM
004	erlwl-ap2.cerl.kek.jp	172.28.64.91	RUN	NO ALARM
005	erlwl-ap3.cerl.kek.jp	172.28.64.92	STOP	MINOR
006	erlwl-br1.cerl.kek.jp	172.28.64.93	STOP	MINOR
007	erlwl-br2.cerl.kek.jp	172.28.64.94	STOP	MINOR
008	erlwl-br3.cerl.kek.jp	172.28.64.95	STOP	MINOR
009	erlwl-br4.cerl.kek.jp	172.28.64.96	STOP	MINOR
010	erlwl-br5.cerl.kek.jp	172.28.64.97	STOP	MINOR
011	erlwl-br6.cerl.kek.jp	172.28.64.98	STOP	MINOR
012	erlwl-ap4.cerl.kek.jp	172.28.64.99	RUN	NO ALARM
013	erlserv2.cerl.kek.jp	172.28.64.100	RUN	NO ALARM
014	erlnetadm.cerl.kek.jp	172.28.64.101	STOP	MINOR
015	cryo-cerl.cerl.kek.jp	172.28.64.102	RUN	NO ALARM
016	cryo-cerl2.cerl.kek.jp	172.28.64.103	RUN	NO ALARM
017	cryo-cerl-lq.cerl.kek.jp	172.28.64.104	RUN	NO ALARM
018	cryo-cerl-2k.cerl.kek.jp	172.28.64.105	RUN	NO ALARM
019	cryo-cerl-logger.cerl.kek.jp	172.28.64.106	RUN	NO ALARM
020	cryo-cerl-3.cerl.kek.jp	172.28.64.107	STOP	MINOR
021	cryo-cerl-4.cerl.kek.jp	172.28.64.108	RUN	NO ALARM
022	cryo-cerl-5.cerl.kek.jp	172.28.64.109	STOP	MINOR
023	cryo-cerl-6.cerl.kek.jp	172.28.64.110	STOP	MINOR
024	cryo-cerl-7.cerl.kek.jp	172.28.64.111	RUN	NO ALARM
025	cryo-cerl-8.cerl.kek.jp	172.28.64.112	RUN	NO ALARM
026	cryo-cerl-9.cerl.kek.jp	172.28.64.113	STOP	MINOR
027	erlserv2a.cerl.kek.jp	172.28.64.114	STOP	MINOR
028	erlnas2.cerl.kek.jp	172.28.64.115	RUN	NO ALARM
029	erlserv2b.cerl.kek.jp	172.28.64.116	STOP	MINOR
030	erlserv2c.cerl.kek.jp	172.28.64.117	STOP	MINOR
031	magnote1.cerl.kek.jp	172.28.64.118	STOP	MINOR
032	magnote2.cerl.kek.jp	172.28.64.119	STOP	MINOR
033	pfrproc6.cerl.kek.jp	172.28.64.120	RUN	NO ALARM
034	cerl-inj-1.cerl.kek.jp	172.28.64.121	STOP	MINOR
035	cerl-inj-2.cerl.kek.jp	172.28.64.122	STOP	MINOR
036	cerl-inj-3.cerl.kek.jp	172.28.64.123	STOP	MINOR

Figure 3: Control panel of host monitoring.

2.2 IOC 監視

IOC 監視は対象の IOC 毎にタイムスタンプレコードを実装し、このタイムスタンプレコードが更新されているか否かで常時監視する機能である。但し、この IOC 監視は IOC と制御機器間で正常に通信が行われているかどうかはまでは考慮していない。あくまで IOC が起動しているか否かの監視に留まる。IOC 監視の構成図を Fig. 4 に示す。

ホスト監視同様に定義ファイルに Alarm で発報する条

件を指定することが可能としている。

GUI では同様に Status・Alarm でフィルタリング可能な他、Group や Function でもフィルタリング可能な仕様とした。cERL において、現在監視している IOC は 200 以上となるため目的の情報を常時表示したい場合などには有効な機能になっている。IOC の状態を一覧で確認できる GUI について Fig. 5 に示す。

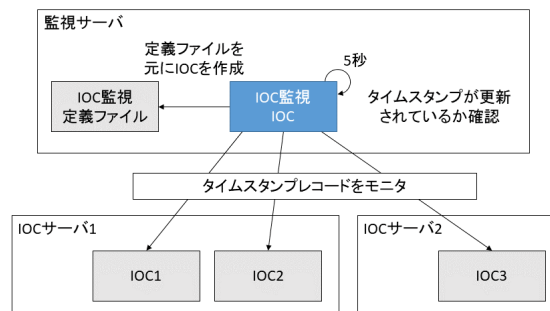


Figure 4: Configuration diagram of IOC monitoring.

No	Ch	Group	Function	Timestamp	Status	Alarm
001	CH001	GUN	550kV Electron-Gun Power Supply	2018/07/24 09:57:10	RUN	NO ALARM
002	CH002	GUN	Inj Vac	2018/07/24 09:57:09	RUN	NO ALARM
003	CH003	GUN	ALOKA MAR783	2018/07/24 09:57:10	RUN	NO ALARM
004	CH004	GUN	GunPS Ext	2018/07/24 09:57:10	RUN	NO ALARM
005	CH005	LASER	Gun Laser Display	2018/07/24 09:57:10	RUN	NO ALARM
006	CH006	LASER	LaserRoomLogger	2018/07/24 09:57:10	RUN	NO ALARM
007	CH007	LASER	Oscillator 1.3GHz	2018/07/24 09:57:10	RUN	NO ALARM
008	CH008	LASER	Oscillator 162.0MHz	2018/07/24 09:57:10	STOP	MINOR
009	CH009	LASER	Surgery DS162	2018/07/24 09:57:10	RUN	NO ALARM
010	CH010	LASER	FilterWheel	2018/07/24 09:57:10	RUN	NO ALARM
011	CH011	LASER	LaserCtrl	2018/07/24 09:57:10	RUN	NO ALARM
012	CH012	MAG	Q scan	2018/07/23 16:52:11	STOP	MINOR
013	CH013	MAG	Mag Degauss Control	2018/07/23 16:52:12	STOP	MINOR
014	CH014	MAG	Mag Standardization Control	2018/07/23 16:52:12	STOP	MINOR
015	CH015	MAG	Mag Sequence Set	2018/07/23 16:52:11	STOP	MINOR
016	CH016	MAG	BendEnergy	2018/07/23 16:52:12	STOP	MINOR
017	CH017	MAG	Kool	2018/07/23 16:52:12	STOP	MINOR
018	CH018	MAG	LeakSump	2018/07/23 16:52:12	STOP	MINOR
019	CH019	MAG	MagModSeq	2018/07/23 16:52:12	STOP	MINOR
020	CH020	MAG	SolenoidScan	2018/07/23 16:52:12	STOP	MINOR
021	CH021	MAG	Rastering	2018/07/23 16:52:12	STOP	MINOR
022	CH022	MAG	CAENeIs LiAM005 No.1	2018/07/23 16:52:11	STOP	MINOR
023	CH023	MAG	CAENeIs LiAM005 No.2	2018/07/23 16:52:12	STOP	MINOR
024	CH024	MAG	CAENeIs LiAM005 No.3	2018/07/23 16:52:11	STOP	MINOR
025	CH025	MAG	CAENeIs LiAM005 No.4	2018/07/23 16:52:12	STOP	MINOR
026	CH026	MAG	CAENeIs LiAM005 No.5	2018/07/23 16:52:12	STOP	MINOR
027	CH027	MAG	CAENeIs LiAM005 No.6	2018/07/23 16:52:11	STOP	MINOR
028	CH028	MAG	CAENeIs LiAM005 No.7	2018/07/23 16:52:12	STOP	MINOR
029	CH029	MAG	CAENeIs LiAM005 No.8	2018/07/23 16:52:12	STOP	MINOR
030	CH030	MAG	CAENeIs LiAM005 No.9	2018/07/23 16:52:11	STOP	MINOR
031	CH031	MAG	CAENeIs LiAM005 No.10	2018/07/23 16:52:11	STOP	MINOR
032	CH032	MAG	CAENeIs LiAM005 No.11	2018/07/23 16:52:11	STOP	MINOR
033	CH033	MAG	CAENeIs LiAM005 Seq	2018/07/23 16:52:11	STOP	MINOR
034	CH034	MAG	CAENeIs ST3534 SW	2018/07/23 16:52:12	STOP	MINOR
035	CH035	MAG	CAENeIs ST3534 Z0	2018/07/23 16:52:12	STOP	MINOR
036	CH036	MAG	Injection_PS_PLF.No.1(cerlmonkhor?)	2018/07/23 16:52:12	STOP	MINOR

Figure 5: Control panel of IOC monitoring.

2.3 差分監視

差分監視は 2 つの数値レコードの差を常時監視する機能である。主に電磁石電源の設定値とモニタ値の差を監視しているが、電磁石の消磁や標準化において差が大きくなり Alarm が発報してしまうことがあった。この為、消磁や標準化中は監視しないといったことや一定時間差があった場合に発報するという動作を実装している。また、電磁石電源によってモニタできる値の誤差が異なるため許容する範囲を個別に指定できる仕様としている。差分監視の構成図を Fig. 6 に示す。

GUI でも同様に 2 つの監視レコード・差分・Alarm でフィルタリング機能を設けている。同様に監視対象が 200 を超えているため、この機能によって発報した対象が即座に判別できるようになっている。差分の状態を一覧で確認できる GUI について Fig. 7 に示す。

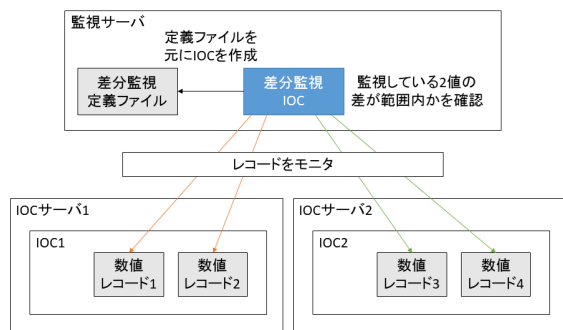


Figure 6: Configuration diagram of differential monitoring.

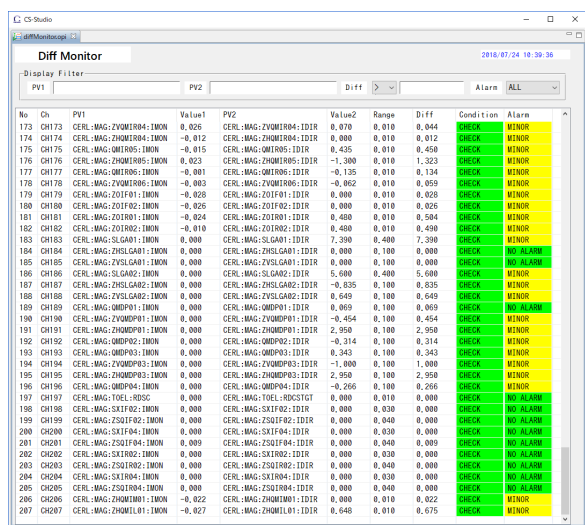


Figure 7: Control panel of differential monitoring.

2.4 プロセス監視

プロセス監視は指定したプロセスが指定したホストで動作しているか監視する機能である。ssh を使用してリモートで ps コマンドを実行し、その応答を見てプロセスが動作しているか否かを監視している。リモートで行うことによってメンテナンスが容易になるためこの方法をとっている。監視には 2 つの検出モードを用意した。一つは単純に ps コマンドで検索する文字列を指定するモードで、もう一つは ps コマンドの結果が複数あった場合に procsfs にある情報をチェックすることによって重複起動かどうか判断するモードである。この重複起動をチェックするには監視対象となるプロセスの起動方法に関わるため注意が必要となる。またホスト監視と同様、IOC とスクリプトに分かれた構成としている。役割も同様に主な監視はスクリプトで行い IOC は GUI・Alarm と連携している。リモートについてだがスクリプトで動作させるために公開鍵認証でパスワードの入力を省略する方法をとっている。プロセス監視の構成図を Fig. 8 に示す。

GUI でも他の監視と同様にフィルタリング機能を設けている。プロセスの状態を一覧で確認できる GUI を Fig. 9 に示す。

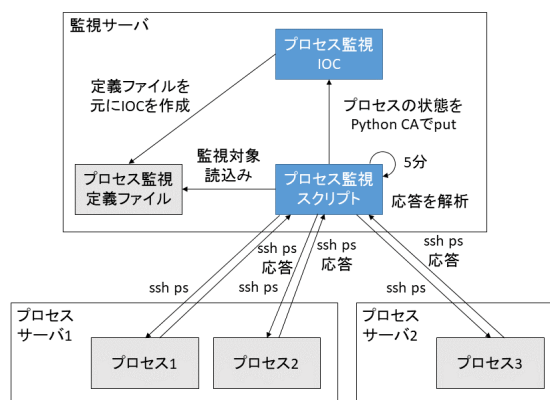


Figure 8: Configuration diagram of process monitoring.

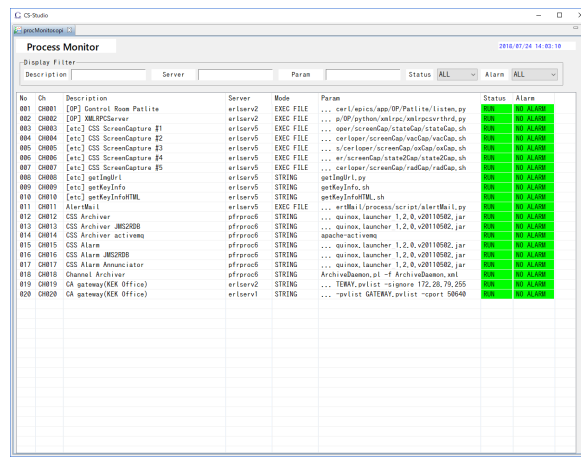


Figure 9: Control panel of process monitoring.

3. システム連携

監視システムは異常を的確かつ速やかに検出し通知する必要がある。そのために cERL や PF で既に運用している CSS Alarm と連携するようにした。CSS Alarm は多数の異常通知を階層構造で管理でき、発報時間や重症度などの管理も可能である。既存の CSS Alarm は IOC との間に中間 IOC を介して連携している。中間 IOC は対象のレコードをモニタし CSS Alarm 用に上限・下限値の判定や重症度の付加を行っているが、今回開発したホスト監視・IOC 監視・差分監視・プロセス監視では内部でこの機能を有しており、中間 IOC を介さず直接 CSS Alarm と連携することによってメンテナンス性を容易にした。CSS Alarm の表示画面を Fig. 10 に示す。

異常発生時にはメールによる通知システムと連携をとるようにしている。これは python で作成したスクリプトで、レコードをモニタし指定した条件になった場合に smtp サーバを介しメールを送信するシステムである。例えばあるホストがダウンした時にメールで通知したい場合、ホスト監視のアラームレコードをメール通知システムに登録することによって異常発生メールとして運用することが可能である。現在は主にサーバがダウンした時に異常通知メールとして送られるように設定し運用している。

また、ホスト監視のみではあるが web 経由でホストの状態が確認できるようになっている。これは web で表示する

ことを当初の仕様としたためであり、必要であれば他の監視機能も web で確認できるようにしていく。

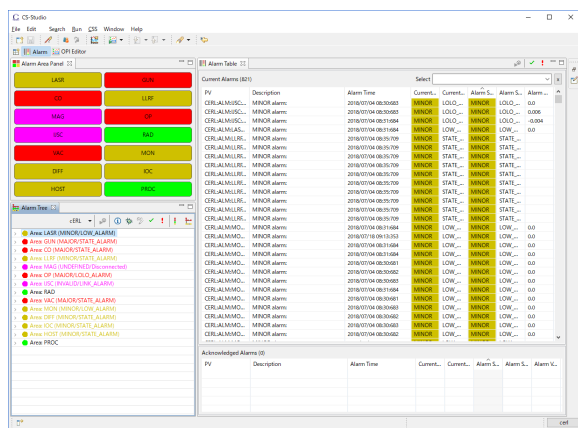


Figure 10: CSS alarm panel.

4. 運用状況と効果

4.1 cERL

cERL での運用状況は概ね良好であった。例えばホスト監視では測定機器のハングアップなどで ping に反応しなくなるといったトラブルが発生した際に、これを検出し通知されたため速やかに対処することができた。

IOC 監視では IOC の起動漏れといった人的ミスも CSS Alarm で確認できるので加速器の立ち上げ時にも大きな効果があった。

差分監視については主に電磁石電源の監視を行っている。台数も多く加速器の調整時には頻繁に変更されるので電源の故障などによる設定のずれにもいち早く対処することができた。

プロセス監視についてはプロセスが停止したと誤検出する事例があった。調査したところ対象サーバのメモリがキャッシュに使用され圧迫しており、swap も 100% に近い状況であった。ssh でログインしても反応が遅いため、リモートで実行しようとしているプロセス監視がタイムアウトなどにより意図しない応答で処理が継続できなかったものと思われる。現在はサーバのメモリ使用量を改善し様子をみている状況である。

4.2 PF

PF での運用状況だがホスト監視において誤検出が 2 回ほど発生した。一つは仮想マシンの監視で発生しており、ログなどを見ても異常が見当たらず原因不明である。発生頻度は運用を開始してから 1 度だけなので様子を見ている状況である。もう一つは同時刻に複数のスイッチで誤検出が発生した。当初は運転期間外なのでメンテナンスため電源 OFF したと思われたが、SNMP による記録を調べたところダウンはしていなかったデータが見られた。この件については今後も継続して調査をしていく。

差分監視においては、挿入光源の補正電源において検出通知があった。ハードウェアの故障かと思われたが、機器の電源 OFF/ON で復旧したため様子を見ている状態である。この件についても早期に対応出来たため運転への影響は少ない状況であった。

IOC 監視については問題なく運用された。プロセス監

視については、これから構築・運用していく予定である。

5. まとめと今後の課題

ホスト監視・IOC 監視・差分監視・プロセス監視を cERL に実装し運用を開始した。一部の機能ではあるが PF でも実装し運用を開始した。誤検出してしまう問題もいくつかあるので今後対策をしていかなければならない。また、PF では一部の機能のみで全て動作していないため今後対応していく。

今後の課題だが監視機能の充実を考えている。例えば、IOC と機器間の通信が正常に行われているかどうかを監視する機能である。これについては実現方法の検討も含めて設計・実装をしていかなければならない。他には、プロセスが起動していること自体はプロセス監視で実現できているが、サービスとして動作しているかどうかの監視も行っていきたい。例えば、データベースへのアクセスが正常に行えるかどうかや、web ページへ正常にアクセスできるかなどが挙げられる。これについても実現方法の検討から行っていく必要がある。

参考文献

- [1] EPCIS; <http://www.aps.anl.gov/epcis/>
- [2] CSS (Control System Studio); <http://controlsystemstudio.org/>
- [3] T. Nakamura *et al.*, “SuperKEKB の CSS アラームシステム運用状況”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Japan, Aug. 8-10, 2016.
- [4] Python CA; <http://www-acc.kek.jp/kekb/control/Activity/Python/CA-Python.html/>